

Predavanje 8

Projektovanje informacionog sistema

Prelaz sa analize na projektovanje

Cilj analize je bio da se napravi logički model koji će opisati kako funkcioniše postojeći sistem. Ovo se zasniva na pretpostavci da postojeći sistem predstavlja dobar primer onog što će se zahtevati i od novog sistema (ovo nije isto sa time kako će novi sistem to da ostvari). Postojeći sistem i njegov model sigurno imaju određena ograničenja.

1. Može se desiti da postoje novi zahtevi, koje postojeći sistem ne ispunjava – ovo treba uzeti u obzir.
2. Može se desiti da se neefikasnost postojećeg sistema manifestuje i u logičkom modelu. Model bi trebalo da otkrije to što u postojećem sistemu nije dobro i da to popravi.
3. Ponekad se dešava da se fizički aspekti uvuku u logičku analizu. To treba izbaciti.

Novi zahtevi

U trenutku kada informacioni sistem treba da pretrpi velike promene, kao što je slučaj kada se manuelni sistem zamenjuje računarskim ili kada se stari računarski zamenjuje novim, treba napraviti reviziju zahteva koji se pred informacioni sistem postavljaju. Ta revizija može dovesti i do novih zahteva. Potreba za zadovoljenjem novih zahteva se ustanovljava intervjuisanjem korisnika i menadžera. Po svojoj definiciji, logički model postojećeg sistema ne može da sadrži nove zahteve, pa ga je zato potrebno proširiti.

Proširenja će vrlo verovatno voditi ka novom procesu, koji onda treba dodati u dijagrame tokova podataka na najvišem nivou. Dalje se vrši dekompozicija ubačenih procesa na pojedine potprocese, koji se prikazuju na dijagramima tokova podataka nižeg nivoa.

Novi zahtevi se češće odnose na pribavljanje novih vrsta informacija za menadžere iz postojećih skladišta podataka, nego na promenu postojeće obrade tih podataka. Takvi zahtevi se prikazuju na dijagramu toka podataka, tako što se ubacuju novi procesi koji kao ulaz prihvataju podatke iz skladišta podataka i kao izlaz daju relevantne podatke. U takvim slučajevima treba proveriti i li su putanje kroz model u skladu sa postojećim entitetima.

Neefikasnost postojećeg sistema

Kreiranje dijagrama tokova podataka se obično radi na osnovu dekompozicije odozgo na dole glavnih procesa. Ono što se na dijagramima nižeg nivoa prikazuje nije uvek samo posledica logičke analize glavnih procesa. Struktura ovih dijagrama je obično određena delimično time šta radi postojeći sistem, a delimično time šta treba da se uradi da se ispuni funkcija. Ako je ono što postojeći sistem radi neefikasno, to može imati neželjene posledice po logički model. U ovoj fazi treba izvršiti prilagođavanje modela.

Pogledajmo na primer neko preduzeće koje prima narudžbe od kupaca, koje treba obraditi. Kopija narudžbenice se privremeno čuva na jednom mestu. Pravi se druga kopija, koja se dalje šalje na proveru da li na računu ima dovoljno novca. Ako je posle provere stanja računa sve u redu, onda se uskladištena kopija koristi za slanje potvrde kupcu da je njegova narudžba prihvaćena. Ovo privremeno skladištenje kopije narudžbenice je posledica načina rada postojećeg sistema. Tada je bilo potrebno da se sačuva kopija u odeljenju prodaje, sve dok drugo odeljenje ne izvrši proveru računa. Primenom

računarskog sistema to više nije potrebno. Proces provere stanja na računaru može da automatski pokrene i proces slanja potvrde da je narudžba prihvaćena.

U ovom trenutku nije obavezno rešiti sve probleme postojećeg sistema. Svrha ovog primera je da se pokaže da logički model nastao kao rezultat analize postojećeg sistema nije nepovrediv, odnosno da se može menjati i prilagođavati. Promene se prikazuju kao da su nastale tokom analize, iako su zapravo nastale u fazi prelaza sa analize na projektovanje.

Tretman fizičkih aspekata

Kao što je već pomenuto, može se desiti da se fizički aspekti problema uvuku u logički model. Pogledajmo na primer, isto preduzeće koje prati narudžbe kupaca. U postojećem sistemu se podaci čuvaju u datotekama, i u njemu su narudžbenica, ugovor i otpremnica nerazdvojni trio. Razlog što su oni tretirani zajedno je što se sve čuvalo u datotekama. Međutim, odvajanjem ovih pojmova neće doći do gubitka veze između njih. Ta veza je sada izražena preko atributa, pošto svaki član tria sadrži atributi narudžba. Pored toga, svaka narudžba je preko veza u relacionom modelu povezana sa ugovorom i otpremnicama.

Predlaganje alternativnih rešenja

Do ovog trenutka analitičari su se bavili pitanjima o tome šta treba uraditi da bi se zadovoljili zahtevi. Sada se pažnja usmerava na to, kako da se to uradi. Ovo je proces koji se odvija u dva koraka. Analitičar na početku u glavnim crtama, treba da predloži dve ili tri alternative. Menadžeri će izabrati jednu od predloženih alternativa. Tek posle ovog se radi detaljan projekat izabranog sistema. Možda ste primetili da su u našoj priči faze analize i projektovanja jasno odvojene. Razlog je u tome što je ceo proces lakše objasniti ako se koristi takav pristup. U praksi se takvi slučajevi retko dešavaju, odnosno ove faze se najčešće uzajamno preklapaju.

Zašto bi analitičari predlagali alternativne sisteme, ako je već urađena studija izvodljivosti, koja sadrži neku okvirnu cenu? Razlog je taj, što analitičari sada imaju mnogo jasniju sliku sistema nego ranije. Možda će se sada pojaviti nove mogućnosti. Ako imate definisan logički model, mnogo je lakše skicirati i menadžmentu predstaviti alternative.

U pripremi projekta analitičar će morati da napravi neke izbore. Biće odluka o tome koje su granice kompjuterizacije, odnosno, koji procesi će se raditi pomoću računara, a koji će ostati manuelni. Biće tu i odluka o tome tipu sistema koji će se koristiti za procese koji se automatizuju. Iako je ovo trenutak kada odluke postaju eksplicitne, verovatno je da su izvesna razmatranja o tipu sistema već rađena.

U ovom trenutku treba definisati sledeće:

- Da li će se koristiti centralizovan ili distribuiran sistem
- Da li će se podaci čuvati u datotekama ili u bazi podataka.
- Da li će se koristiti batch ili onlajn sistemi
- Kakvi su metodi unosa podataka
- Da li će se koristiti gotovi programi ili programi pravljeni po narudžbi
- Kakav će se hardver koristiti

Centralizovan ili distribuiran sistem

Distribuiran sistemi su oni sistemi kod kojih:

- Postoje dva ili više geografski odvojena računara
- Su ti računari povezani telekomunikacionim linkom
- Mreža računara funkcioniše kao jedna organizacija

Centralizovani sistem je onaj kod kojeg jedan računar servisira sve potrebe preduzeća. U ovakvim slučajevima mogu da postoje udaljeni terminali, koji su telekomunikacionim linkovima povezani sa tim serverom.

Velika preduzeća koja su fizički raspoređena na različitim lokacijama mogu odlučiti da postave distribuirani sistem, pošto se veći deo obrade podataka i kreiranja informacija dešava na licu mesta. U tom slučaju je moguće koristiti lokalna skladišta podataka i lokalnu obradu. Potreba za distribuiranim sistemom (kao razlika u odnosu na više odvojenih samostalnih računara), dolaze iz potrebe da podaci i rezultati obrade tih podataka, koji nastaju na jednom mestu, treba da budu dostupni i na drugim mestima.

Preduzeće koje se fizički nalazi na jednoj lokaciji se takođe može odlučiti da koristi skup mikro ili mini računara, koji će se povezati preko lokalne mreže. Ovo je moguće uraditi ako nije potrebna velika snaga za obradu podataka. Prednost ovakvog načina rada je u tome da svaki čvor na mreži može biti korišćen za lokalne potrebe (na primer unakrsne tabele), a takođe se u trenutku kada je to potrebno u mrežu mogu dodavati novi čvorovi.

Prednosti distribuiranog načina rada su:

- Cena telekomunikacionog linka između različitih sajtova je smanjena, pošto se veći deo posla obavlja na lokalnim računarima.
- Veća je fleksibilnost, jer se u slučaju potrebe u mrežu mogu ubaciti novi računari.
- Preduzeće nije vezano samo za jedan računar, koji može da otkáže.

Nedostaci distribuiranog sistema su:

- Za podatke koji se često koriste se često dešava da postoji više kopija na različitim mestima. Promena ovih podataka može dovesti do nekonzistentnosti baze.
- Ako se računari nalaze na različitim mestima, može se desiti da se koristi različita oprema, softver i skladišta podataka.
- Teže je obavljati kontrolu

Sistemi sa datotekama podataka ili sistemi sa bazama podataka

Skladište podataka koje će programe snabdevati podacima može se projektovati ili kao niz nezavisnih datoteka ili kao integrisana baza podataka. Bazu podataka treba izabrati u sledećim slučajevima:

- Ako se isti podaci koriste u različitim aplikacijama.
- Ako je podataka u jednom delu preduzeća povezan, ili integrisan sa podacima u drugim delovima.
- Ako je struktura podataka stabilna.
- Ako je verovatno da će se aplikacije, koje podatke koriste, tokom vremena menjati.
- Ako je postoje zahtevi za fleksibilnim izveštajima.

Sa druge strane sledeće karakteristike upućuju na to da treba koristiti sistem datoteka:

- Različite aplikacije koriste vrlo malo zajedničkih podataka.
- Potrebna je brza obrada transakcija
- Nisu potrebni fleksibilni i složeni izveštaji.

Ako iz prethodnog zaključite da treba koristiti sistem baze podataka, treba imati na umu da ovi sistemi imaju i prednosti i nedostatke.

Prednosti sistema sa bazama podataka su:

- Redundantnost podataka je minimalna, čime se baza jednostavno zadržava u konzistentnom stanju.
- Produktivnost programera aplikacija se povećava.
- Kontrola nad podacima je centralizovana.
- Različite aplikacije mogu da lako dele iste podatke.
- Fizički detalji o skladištenju podataka su skriveni od korisnika. Korisnici vide samo poglede na te podatke, koji su njima prilagođeni.

Nedostaci sistema sa bazama podataka su:

- Potrebno je unapred napraviti integrisani inicijalni projekat (nije lako uraditi), umesto da se radi malo po malo u trenutku kada se naiše na problem.
- Obrada je obično sporija nego kod sistema sa datotekama.
- Potrebno je kupiti skupi softver za upravljanje bazama podataka.
- Potrebno je dosta mesta na disku.

Batch ili onlajn sistemi

Jedan od izbora koji analitičar treba da napravi je da li će koristiti batch ili onlajn sisteme. Neki delovi sistema se mogu napraviti tako da funkcionišu u jednom, a drugi u drugom režimu rada.

Kod batch režima rada, transakcije koje se unose u sistem se privremeno čuvaju, pre nego što se obrade. Obrada se vrši u trenutku kada se sve unese. Tada se obrađuje ceo skup transakcija. Prednosti ovakvog načina rada su:

- Kontrola. Pošto se podaci ne obrađuju odmah nakon unošenja, postoji još jedna šansa za otkrivanje i otklanjanje grešaka. Moguće su različite kontrole:
 - Kontrola ukupne sume: Može se na primer, ručno izračunati ukupna suma, koja se poredi sa onom koju daje računar.
 - Vizuelna provera: Može se odštampati lista svih transakcija, nakon čega se to poredi sa izvornim dokumentima.
 - Broj transakcija: Broj transakcija koji daje računar se može uporediti sa ukupnim brojem transakcija koji je dobijen ručno.
- Efikasnija obrada. Podaci se mogu obraditi u trenucima kada su računari manje opterećeni.

Batch sistemi se koriste kod obrade velikih količina transakcija koje se ponavljaju i to tamo gde su kontrola i efikasna upotreba računara bitni. Tipičan primer su glavna knjiga, platni spiskovi i sl. Nedostatak ovakvog načina rada je u tome da može proteći izvesno vreme pre nego što transakcija promeni datoteku ili bazu. Ovo je nedostatak samo u slučajevima kada su potrebni ažurni izveštaji.

Onlajn obrada je obrada kod koje se transakcija obrađuje odmah nakon unosa. Ovde nije moguća grupna provera podataka, kao u prethodnom slučaju, pošto se transakcije obrađuju jedna po jedna. Zahtev za neposrednom obradom transakcija vodi ka tome da računar mora biti na raspolaganju sve vreme.

Načini unosa podataka

U ovoj fazi analitičar treba da definiše načine unosa podataka. Prilikom odlučivanja o tome treba u obzir uneti sledeće:

- Količina transakcija koje se unose
- Tražena brzina unosa
- Inicijalna cena i cena upotrebe izabranog metoda
- Traženi stepen tačnosti kod unosa
- Specifične karakteristike aplikacije

Metod unosa podataka je blisko povezan sa odlukom o tome da li se obrada transakcija radi po principu batch ili onlajn obrade. Metodi unosa kod kojih je moguće uneti velike količine podataka, kod kojih se unos ponavlja i sl. su kandidati za batch obradu.

Osnovni načini unosa podataka su:

Tastatura: Ovakav unos ima malu inicijalnu cenu, ali veliku kasniju cenu upotrebe, pošto je to unos koji zahteva da korisnik puno radi. Ako ne bi bilo pomoći od strane računara u proveru podataka (provera sifara i sl.), verovatno bi ovakav način unosa doneo mnogo grešaka. Osnovna prednost ovakvog načina rada je da je to vrlo fleksibilan način rada, posebno u pogledu vrste podataka koji se mogu uneti.

Prepoznavanje karaktera: Osnovni predstavnici ovakvog načina unosa su optičko prepoznavanje karaktera (OCR) i prepoznavanje sa magnetnim mastilom (MICR). Ulazna oprema "čita" ulazni dokument i konvertuje karaktere u oblik koji mašina razume. U ovom slučaju postoji visoka inicijalna cena opreme, ali je kasnija cena upotrebe vrlo niska. To znači da je ovakav način rada pogodan kod unosa velikih količina podataka. Jedna od karakteristika je i manja količina grešaka. Ovakv pristup je ograničen na situacije kada je originalne podatke moguće odštampati, na primer, kod broja čeka i računa.

Optičko prepoznavanje potvrđenih polja: Ovo se koristi kada postoje odštampani dokumenti sa poljima za potvrdu. Potvrđena polja mogu biti pročitana pomoću specijalne opreme, koja je programirana za određena polja. Prisustvo ili odsustvo znaka za potvrdu se prevodi u oblik koji računar može da pročita. Ovde takođe postoji visoka inicijalna cena, dok je cena upotebe mala, a takođe ima i malo grešaka. Ovakav metod unosa je pogodan kod aplikacija velikih količina podataka, kod kojih postoji izbor između zadatih alternativa. Tipične oblasti primene su istraživanja tržišta, ocene testova ili izbor iz magazina.

Čitanje bar koda: Bar kodovi koji se koriste kod identifikacije se štampaju i lepe na predmete. Bar kod se čita pomoću specijalnog čitača, obično pomoću lasera. Bar kod mora biti postavljen na neki predmet, pa je prema tome, ovakav način unosa podataka vezan za materijalna dobra. Primeri su svuda

oko nas, na primer u samouslugama, bibliotekama ili magacinima. Unos podataka je vrlo jednostavan i pri tome nema grešaka.

Unos podataka pomoću glasa: Danas postoji i unos pomoću glasa. Danas postoje određene aplikacije, koje mogu da razumeju ograničen skup reči, na primer izbor nekih stavki iz menija. Dalji rad na usavršavanju prepoznavanja govora može dovesti i do mnogo šire upotrebe ovog načina unosa.

Udaljeni unos: Ako se sistemu može pristupiti iz daljine, na primer, preko Interneta, onda podatke na server mogu da šalju direktno službenici ili kupci. Proces unosa podataka može biti automatizovan, kao što je slučaj kod primene EDI standarda, ili se može koristiti ručni unos, kao što je slučaj kod B2C sajtova.

Kupovina gotovih programa ili izrada po narudžbi

Umesto da se za novi informacioni sistem piše novi program, može se kupiti već gotov. Takve aplikacije pišu specijalizovana softverska preduzeća, tako da je cena relativno manja u odnosu na razvoj od početka. Prednost gotovih aplikacija je i u tome da se brzo implementiraju, da se pre kupovine mogu da pogledaju i da generalno postoji dobra dokumentacija. Prednost je i u tome što su prethodni kupci već izvršili testiranje aplikacije, tako da postoji mogućnost da se ti kupci pozovu i konsultuju pre kupovine.

Nedostatak paketa je u tome što se teško integrišu sa postojećim softverom koji je pisan specijalno za to preduzeće. Može se desiti i da gotovve aplikacije ne odgovaraju u potpunosti specifičnim potrebama preduzeća. Softverska kuća koja prodaje aplikaciju pokušava da u nju ugradi što je više moguće fleksibilnosti, čime se proširuje moguće tržište. Fleksibilnost sa sobom donosi opasnost pojave neefikasnosti.

U poslednje vreme postoji trend da se kupuju aplikacije koje se mogu prilagoditi različitim zahtevima konkretnih preduzeća. Neki od proizvoda su namenjeni specifičnim tržištima, kao što je na primer program za evidenciju studenata, koji se može prodati samo fakultetima. Jezgro sistema se zadržava, ali se mogu dobiti i dodatni moduli, ili funkcionalnosti.

Postoje i primeri koji su mnogo opštiji po nameni. Tu na primer, spada ERP softver, kao što je onaj koji proizvode firme SAP ili Oracle. Ovi softverski sistemi pokušavaju da podrže različite oblasti rada u preduzećima. Za kupca ovakvog softvera se cena smanjuje pošto se ona deli i sa drugim kupcima. Atraktivna je i mogućnost da se paket prilagodi konkretnim zahtevima. Prednost za prodavca je u tome što jezgro softvera može da proda na više mesta. Fleksibilnost prodavcu donosi mogućnost da sklapa nove ugovore, u pogledu održavanja i nadogradnje.

Analitičar će predložiti kupovinu gotovih aplikacija ako:

- Je cena osnovni faktor
- Sistem treba da se brzo implementira
- Zahtevi u pogledu obrade podataka i informacija su standardni
- Preduzeće nema specijalizovano odeljenje za softver, koje bi održavalo i pisalo program.
- Postoje oprobane aplikacije iz te oblasti.

Kakav se hardver koristi

Različiti projekti informacionog sistema imaju različite hardverske zahteve. Nešto od toga treba uzeti u obzir i ranije prilikom izbora metoda unosa podataka i opreme koja se za to koristi. Analitičar treba da donese odluku o opremi za obradu, izlaz, komunikaciju i skladištenje. Ovo u velikoj meri zavisi

od količine obrade koja je potrebna, od složenosti softvera kao i broja korisnika koji sistem koriste. Na hardver takođe utiče i izbor distribuiranog ili centralizovanog sistema.

Analitičar menadžmentu treba da ponudi dve ili tri alternativne opcije. Menadžerima će biti potrebno ne samo objašnjenje alternativnih scenarija, već i procena cene i vremena potrebnog za izvršenje alternativa.

Granice automatizacije

Pored toga što pokazuje kako se podaci kreću od jednog do drugog procesa, dijagram toka podataka ukazuje i na to koji procesi treba da budu automatizovani. Ovo se na dijagramu prikazuje preko granice automatizacije.

U ovoj fazi analitičar treba da predloži nekoliko alternativa. Alternative treba da posluže kao osnova za diskusiju. Rezultat svega ovog treba da bude nekoliko predloga:

- Minimalni sistem, sa najnižom cenom, koji zadovoljava osnovne potrebe.
- Sistem srednje veličine, koja pruža više mogućnosti, ali i više košta.
- Kompletan sistem koji ilustruje kako preduzeće može da u potpunosti iskoristi sve prednosti automatizacije.

Prilikom predstavljanja različitih scenarija analitičar treba da donese odluke u vezi svih stavki koje su pomenute (onlajn ili batch sistem, centralizovani ili distribuiran itd.). Za svaki predlog mora da postoji i procenjena cena i vreme. Prilikom projektovanja sistema analitičar uvek mora da na umu ima ograničenja koja postoje, kao i ciljeve koji su doveli do automatizacije.

Predloženi sistem treba da se menadžmentu predstavi u nekom formalnom obliku. U tom trenutku menadžment treba da izabere jedno rešenje i da zeleno svetlo za početak radova. Tokom daljeg rada se povremeno vrše demonstracije i daju preseći stanja.

Demonstracije

Analitičar u različitim fazama svog rada treba da nekom drugom prezentuje ono što radi. Ovo se radi pomoću demonstracija i formalnih pregleda. Razlika je u tome da demonstracija predstavlja interaktivnu prezentaciju nekog dela sistema i da je vrši analitičar. Cilj je da analitičar prikaže svoje ideje ili probleme i da dobije odgovarajući odgovor.

Demonstracija se vrši pred nekoliko ljudi. Za ilustraciju mogu da se koriste dijagrami tokova podataka, ER dijagrami. Sama demonstracija ne bi trebalo da traje više od 30 minuta. Ostali učesnici mogu biti drugi analitičari ili programeri, korisnici ili menadžment. Demonstracije se obično vrše kada analitičar zapadne u neke probleme i kada mu je potrebna pomoć.

Formalni pregledi

Formalni pregledi se mogu dešavati u različitim trenucima i oni su znak da su u procesu dostignute određene bitne tačke. Uspešan pregled ukazuje na to može da počne sledeća faza, dok neuspešan znači da se posao mora da ponovi.

U formalnom pregledu učestvuje moderator sastanka, tehnička lica, predstavnici korisnika i specijalisti koji će imati koristi od sistema kad počne da radi. Na kraju tu su i analitičari sistema, koji se mogu povremeno uključivati.

Detaljan projekat sistema

Nakon što su donete neke generalne odluke o nivou automatizacije i hardveru i nakon što je menadžment dao odobrenje za nastavak posla, prelazi se na detaljan projekat sistema.

Detaljan projekat sistema sadrži specifikaciju hardvera, softvera, baze podataka, korisničkog interfejsa i termina implementacije. Specifikacija je osnovni dokument koji se dalje koristi. Nakon što se za njega dobije odobrenje, može se pristupiti nabavci hardvera, pisanju ili kupovini softvera, obuci osoblja, razvoju baze podataka itd.

Projekat procesa

Projekat procesa obuhvata definisanje hardverskih i softverskih resursa.

Prilikom definisanja hardvera analitičar treba da razmotri zahteve koji će se postavljati pred procesor. Konkretno, to znači da treba da uzme u obzir količinu transakcija, brzinu odgovora, broj istovremenih korisnika, tipove izlaznih uređaja, potrebnu količinu RAM memorije, složenost programa, kao i oblast do koje će biti potrebna buduća proširenja sistema.

Za specifikaciju programa se mogu koristiti različiti alati. Programi se sastoje od više modula. Module treba napraviti tako da svaki obavlja nezavisnu funkciju, da može da se nezavisno testira i da se kod što lakše prati i razume. Modularni softvera takođe, omogućava kasniju lakšu dopunu novim modulima. Veze između modula se mogu predstaviti strukturnim dijagramima.

Moduli

Modul je u suštini skup instrukcija koje se zajedno izvršavaju. U fazi projektovanja ove instrukcije se mogu definisati u običnom govornom jeziku ili tabelama odlučivanja. Kasnije će ta specifikacija biti prevedena u konkretan programski jezik.

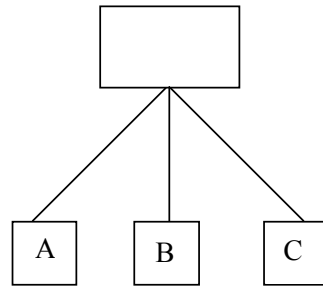
Iz jednog u drugi modul se obično šalju neki podaci, tako da se i tu mogu koristiti dijagrami tokova podataka. Svaki modul se i samostalno modelira i prikazuje. Ta specifikacija modula, zajedno sa rečnikom podataka je dovoljna da programer može da piše kod.

Modularna struktura se dosta razlikuje od tradicionalnih metoda u programiranju, kod kojih je postojao monolitni kod. Integracija funkcija je omogućavala programerima da pišu vrlo efikasan kod (u pogledu upotrebe procesora). Sa druge strane, programe je bilo teško pratiti i vrlo su se često javljali bagovi, za čije je ispravljanje bilo potrebno mnogo vremena. Promena koda u nekoj kasnijoj fazi, radi, na primer, dodavanja nove funkcije je mogla da uvede nepredviđene efekte u druge delove programa. Neka istraživanja su pokazala da preko 50% procenata cene otpada na debugovanje, dopunu i generalno na održavanje softvera.

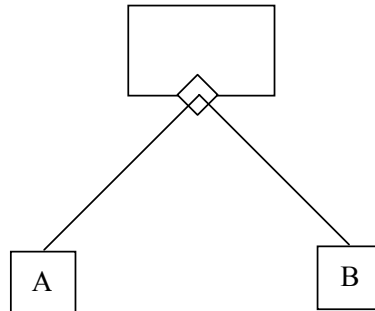
Strukturni dijagrami

Strukturni dijagrami, poput onog prikazanog na sledećoj slici, su grafički prikaz načina na koji su moduli povezani. Za svaki modul se prikazuju podaci koje koristi i koje pruža. Na dijagramu nema više informacija u odnosu na specifikaciju samog modula. Sa druge strane, strukturni dijagram daje sliku sistema koju je lako razumeti. Strukturni dijagram je hijerarhijski.

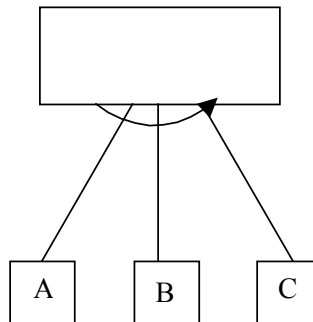
a)



b)



c)



Strukturalni dijagram

Za module se pretpostavlja da pozivaju svoje podmodule i da se izvršavaju sa leva u desno. Svaki modul može da poziva svoje podmodule, i ne smatra se završenim sve dok se svi podmoduli ne završe.

Ako se izvršava jedan ili drugi modul, proces odlučivanja se prikazuje preko romba, a ako se modul izvršava više puta koristi se kružna strelica.

Modularni projekti imaju niz prednosti:

- Lakše je projektovanje programa. Analitičari mogu da na višem nivou definišu osnovne zadatke i da kasnije to razlažu.
- Struktura koja nastaje tokom modularnog projektovanja se sastoji od delova kojima se može upravljati. To znači da su delovi sistema i njihove međusobne veze tako predstavljeni da se vide kao celina.
- Pojedinačni programeri i projektanti mogu da relativno nezavisno rade na zasebnim modulima.
- Upravljanje projektom je olakšano.

- Zasebni moduli se mogu zasebno testirati, tako da se greške lakše uočavaju i ispravljaju.
- Krajnji sistem je modularan. To znači da se pojedini moduli mogu izbaciti, promeniti ili se mogu dodati novi, a da to ne utiče mnogo na ostale module.

Da bi moduli mogli da funkcionišu na ovakav način, moraju biti slabo povezani i kohezioni.

Povezanost modula se može definisati kao činjenica da promene u jednom delu sistema vode ka promenama u drugom delu. Povezanost kod modula treba da bude što manja. Dva modula su slabije povezana ako:

- Je manji broj tipova podataka koji se prosleđuje od jednog do drugog.
- Ako manje dele iste podatke iz baze podataka.

Modul koji je slabo povezan sa drugim modulima ima jednostavan interfejs i zbog toga ga je lakše projektovati, kodirati, testirati i menjati.

Moduli su kohezivni ako:

- Se sastoje od samo jedne funkcije
- U slučaju da imaju više funkcija, te funkcije se izvršavaju redom.
- Obavljaju skup nezavisnih akcija, koje se izvršavaju u istom trenutku. Primer za ovo su rutine za inicijalizaciju.

Prednosti modularnog dizajna se mogu u potpunosti iskazati jedino ako su moduli slabo povezani i interno kohezivni.

Projekat skladištenja podataka

Tokom analize i projektovanja sistema napravljen je konceptualni model podataka. On je zasnovan na entitetima i njihovim vezama. Kasnije se definišu i atributi tih entiteta. Sledeći u nizu je normalizovani model. Tokom projektovanja sistema je donešena odluka da li će se koristiti datoteke ili baza podataka. Bez obzira na to šta se izabere potreban je detaljan projekat skladišta podataka.

Skladištenje podataka u datotekama

Ovakvo skladištenje se generalno bira ako aplikacije ne koriste iste podatke i ako je potrebna brza obrada transakcija. Analitičar treba da napravi neki okvir za sve podatke. To znači da treba da definiše polja u zapisima, njihov tip i dužinu. Zapisi mogu biti fiksne ili promenljive dužine. Promenljiva dužina se javlja ako je moguće da neka polja budu promenljive dužine, ili ako se neka polja mogu da ponavljaju.

Pored tipa zapisa mora se doneti i odluka o organizaciji skladišta, kao i pristupu svakoj datoteci. To će kasnije definisati i brzinu pristupa pojedinačnim zapisima, kao i kasniju lakoću dodavanja i brisanja zapisa. Tipovi skladištenja koji se koriste u ovom slučaju mogu biti sekvencijalna organizacija, sa indeksima, sekvencijalna, ali sa indeksima, po slučajnom principu sa heširanjem itd. Različite organizacije datoteka imaju različite karakteristike.

Analitičar takođe treba da definiše i hardver za skladištenje rezervnih kopija podataka. To će verovatno biti optički ili magnetni disk, a ponekad i magnetne trake. Prilikom izbora hardvera za backup, analitičar mora u obzir da uzme broj datoteka, kao i broj zapisa u svakoj datoteci, organizaciju datoteka koja je izabrana, vreme odgovora, kao i buduće zahteve, posebno u domenu povećanja veličine.

Baze podataka

Ako se podaci treba da dele između različitih aplikacija i ako postoji visok stepen integracije tih podataka, verovatno će se izabrati sistem za upravljanje bazom podataka.

Prilikom izbora modela baze podataka analitičar treba da koristi konceptualni model podataka, koji je ranije razvijen. Relacioni sistemi za upravljanje bazama podataka pružaju veliku fleksibilnost u skladištenju i vađenju podataka. Relativni nedostatak je spor pristup i dosta potrebnog prostora.

Relacione baze podataka su danas najverovatniji izbor za skladištenje podataka. Sistem za upravljanje tim bazama dolazi sa jezicima četvrte generacije, interfejsima za povezivanje sa unakrasnim tabelama i mogućnostima povezivanja sa drugim bazama podataka.

Nakon što se odluči za neku bazu podataka, analitičar treba da napravi konceptualnu šemu. Ovo je sveukupni logički pogled na podatke, uz upotrebu jezika za definiciju podataka, koji koristi baza (obično SQL). Taj model treba da sadrži pravila integriteta i zaštite pristupa. Prilikom prelaska sa normalizovanog na relacioni model, obično svaki entitet i veza postaju tabele. Svaki atribut entiteta će postati kolona u tabeli.

Za svakog korisnika baze podataka treba definisati posebnu šemu. Analitičar treba da definiše koji su podaci potrebni kojem korisniku, da bi mogao da obavlja svoje poslove.

Analitičar treba da definiše i internu šemu baze. To znači da treba da definiše kako se baza podataka fizički smešta na disk. Tu treba definisati gde se smeštaju indeksi, pokazivači, treba definisati veličinu blokova i putanje pristupa. Dobra interna šema će dovesti do efikasne baze podataka.

Projektovanje baze podataka se radi zajedno sa projektovanjem modula, jer oni pristupaju toj bazi. To znači da postoje bliske veze između dijagrama tokova podataka, modela podataka, strukturnih dijagrama, specifikacije modula i rečnika podataka. Kod velikih preduzeća model entiteta može da ima i nekoliko stotina tipova entiteta.

Kao i kod sistema sa datotekama i ovde analitičar treba da proceni koji je hardver potreban.

Projekat komunikacije sa sistemom

Korisnički interfejs

Projekat korisničkog interfejsa je vrlo bitan deo projekta celog softvera. Većina programa zahteva da se unose podaci, kao i da se prikažu neki rezultati. Projekat interfejsa ne sada u dijagrame tokova, ali i njega treba napraviti tako da se u obzir uzmu tipovi korisnika i njihovi zahtevi.

Prilikom projektovanja korisničkog interfejsa u obzir treba uzeti sledeće.

Projekat izgleda ekrana

Format i prezentacija materijala na ekranu monitora utiču i na način na koji korisnici komuniciraju sa sistemom. Loše napravljen izgled može da dovede do grešaka u unosu i do toga da korisnik odbije da koristi sistem.

Ekran treba da osigura je materijal predstavljen na ekranu relevantan, prikazan na logički način i dovoljno jednostavan. Ako podaci treba da se unose iz nekih dokumenata, onda ekran treba da izgleda što sličnije dokumentu. Ekran se organizuje uz pomoć sledećih elemenata:

Meniji: omogućavaju jednostavan izbor

Ispuna obrazaca: Ovo je situacija kada kursor skače iz polja u polje i čeka da se polje ispuni.

Interaktivne komande: Kada korisnik koristi tastere za unos komandi.

Kontrola

Korisnici prave greške. To se dešava prilikom unosa. Takođe i podaci koji se koriste kao osnova za unos mogu da sadrže greške. U korisnički interfejs reba da bude ugrađena kontrola. Osnovna kontrola unosa je:

Grupna kontrola: Kontrola redosleda, kontrola zbirnih podataka, listanje radi vizuelnog poređenja.

Softverska kontrola: Provera cifara, provera opsega, rasporeda i sl.

Kodiranje: U cilju smanjenja unosa podataka može se izvršiti njihovo kodiranje.

Dokumentacija za korisnika

Novi korisnici neće moći samo da sednu ispred računara i počnu da ga koriste, ako nemaju odgovarajuću dokumentaciju i pomoć za upotrebu sistema. Korisnici uvek treba da znaju gde se u menijima nalaze i šta tačno treba da urade. Čak se i eksperti ponekad mogu da izgube u složenim sistemima.

Korisniku stalno treba da bude dostupna pomoć. Treba da postoji generalna pomoć za upotrebu sistema, ali i objašnjenje kako se ona koristi. Može da postoji dugme help, ili može da postoji pomoć vezana za kontekst.

Prilikom unosa podataka se može sugerisati podrazumevana vrednost ili u najmanju ruku tip podatka. Na primer, kod unosa datuma može se napisati

Datum (dan.mesec.godina), a ne samo Datum.

Poruke o greškama

Poruke o greškama treba da pomažu u otkrivanju koja je greška u pitanju. Poruke tipa "Nepoznata komanda", ili "Ne postoji takav parametar" nisu baš od koristi, ali su ipak bolje od poruka tipa "Greška 234". Korisna poruka ne samo da treba da analizira grešku već treba da pomogne korisniku da grešku otkloni.

Vreme odgovora

Generalno vreme odgovora na komandu treba da bude kratko. Dugačko vreme odgovora, posebno kod stalnih korisnika dovodi njihovog iritiranja, grešaka i neprihvatanja sistema. Sa druge strane, kod komplikovanih poslova, posebno kod neiskusnih korisnika, brz odgovor može da poveća greške.

U skladu sa prethodnim analitičar treba da korisnički interfejs projektuje u skladu sa korisnikom koji će ga koristiti. Sa tog aspekta korisnici se mogu podeliti na nekoliko grupa:

Povremeni i stalni korisnici. Stalni korisnici sve više znaju o sistemu kako ga više koriste. Kod njih će verovatno da postoji posebna obuka. Kod takvih korisnika je bitno da vreme odgovora bude kratko, da postoje skraćenice za komande i podrazumevane vrednosti.

Pasivni i interaktivni korisnici. Interaktivni korisnici su oni koji razvijaju sistem ili donose odluke o modelu podrške sistema. Oni žele da imaju mogućnost istraživanja sistema. U ovu kategoriju često spadaju oni koji treba da odgovore na onlajn upite. Ovi korisnici koriste svoje znanje o skraćenicama i koriste ih da bi što brže obavili posao. Sa druge strane postoje pasivni korisnici kojima

treba prikazati menije, obrasce za popunu. Njima treba ograničiti oblasti pristupa. To je obično osoblje za unos podataka.

Počelnici ili eksperti. Ovo nije isto kao povremeni i stalni korisnici. Na primer i početnik i ekspert mogu biti povremeni korisnici sistema. Povremeni korisnici, eksperti će verovatno tražiti posebnu pomoć samo za određene delove sistema, dok će početnici tražiti instrukcije korak po korak.

Specifikacija sistema

Detaljan projekat dovodi do izrade specifikacije sistema. Ovo je detaljan i velik dokument, koji opisuje sistem koji treba da se napravi. Ne postoji format ovog izveštaja koji je opšte prihvaćen, ali je verovatno da će u njemu naći nešto od sledećeg:

- Rezime. Ovo je rezime svih glavnih tačaka specifikacije.
- Opis predloženog sistema i posebno njegovih ciljeva. Mogu da se koriste dijagrami toka podataka. Treba da se objasni šta će sistem da radi i različite funkcije tog sistema.
- Kompletna specifikacija:
 - Programa. Tu treba da postoji specifikacija modula, strukturni dijagrami, zajedno sa podacima za testiranje.
 - Ulaza. Treba prikazati izvorne dokumente, ekrane u programu, strukturu menija i kontrolne procedure.
 - Izlaza. Tu treba prikazati izveštaje, koje će sistem pružati.
 - Skladištenja podataka. Ovo je specifikacija baze ili sistema datoteka.
- Detaljna specifikacija kontrole rada sistema
- Specifikacija svih hardverskih zahteva i traženih performansi.
- Termini implementacije sistema
- Procena cene i ograničenja