

Nada Ličen Cibic

OSNOVE PODATKOVNIH BAZ

81 425 044



računalniški sistemi delta

Nada Ličen Čibic

OSNOVE PODATKOVNIH BAZ

81 425 044

Iskra Delta si pridržuje pravico do razenčevanja tega dokumenta in programskega proizvoda, ki ga s tem učbenikom - priročnikom opisuje.

Iskra Delta ne jamči za morebitne napake v priročniku ali programskem proizvodu in za morebitne posledice, ki bi zaradi takih napak nastale.

KAZALO

KAZALO	1
0. UVOD	0-1
1. ORGANIZACIJA PODATKOV.	1-1
1.1 PRVA STOPNJA ORGANIZACIJE PODATKOV.	1-1
1.2 DRUGA STOPNJA ORGANIZACIJE PODATKOV.	1-3
1.3 TRETJA STOPNJA ORGANIZACIJE PODATKOV.	1-3
ZAČASNI UČBENIK - PRIROČNIK	1-3
IZOBRAŽEVALNEGA CENTRA DELTA	1-3
2. PODATKOVNE BAZE.	2-1
2.1 UVOD V PODATKOVNE BAZE.	2-1
2.2 CILJI IN LASTNOSTI PODATKOVNIH BAZ.	2-5
2.2.1 VEČNAMENSKOST PODATKOV.	2-5
2.2.2 ZAŠČITA PRETEKLEGA DELA.	2-5
2.2.3 JASNOST, LANKA IN FLEKSIBILNA UPORABA.	2-6
2.2.4 ENOSTAVNE SPREMEMBE.	2-6
2.2.5 NIZKA CENA.	2-6
2.2.6 REKUPIRANCA.	2-6
2.2.7 IZVRŠITEV ZAHTEV.	2-7
2.2.8 TOČNOST IN KONZISTENTNOST.	2-7
DO Iskra Delta si pridržuje pravico do razmnoževanja tega dokumenta in programskega proizvoda, ki ga s tem učbenikom - priročnikom opisuje.	
2.2.12 FIZIČNA IN LOGIČNA NEODVISNOST.	2-9
DO Iskra Delta ne jamči za morebitne napake v priročniku ali programskem proizvodu in za morebitne posledice, ki bi zaradi takih napak nastale.	
2.2.16 VISOKOHIVULJSKI PROGRAMERSKI VNOSNIN.	2-9
2.2.17 JEZIK ZA KONČNEGA UPORABNIKA.	2-9
2.2.18 KONTROLA INTEGRITETE.	2-10
2.2.19 ORGONA ZA NAČRTOVANJE IN UREJANJE.	2-10
2.2.20 AVTOMATSKA REORGANIZACIJA.	2-10
2.2.21 RAZVOJ V SMERI DISTRIBUIRANIH OBDELAV.	2-10
2.3 NEODVISNOST PODATKOV.	2-11
2.4 VRSTE PODATKOVNIH BAZ.	2-13
3. SOBNE METODE ZA RAZVOJ RACUNALNIŠKO PODPRTIH APLIKACIJ.	3-1
3.1 INFORMACIJSKO INŽENIRSTVO.	3-2
3.1.1 STRATEŠKO PLANIRANJE POTREB.	3-3
3.1.2 INFORMACIJSKA ANALIZA.	3-4
3.1.3 OBLIKOVANJE PODATKOVNEGA MODELA.	3-4
3.2 SOBNI PRIPOMOČKI ZA RAZVOJ APLIKACIJ.	3-5



KAZALO

KAZALO	1
0. UVOD	0-1
1. ORGANIZACIJA PODATKOV.	1-1
1.1 PRVA STOPNJA ORGANIZACIJE PODATKOV.	1-1
1.2 DRUGA STOPNJA ORGANIZACIJE PODATKOV	1-3
1.3 TRETJA STOPNJA ORGANIZACIJE PODATKOV.	1-3
1.4 ČETRTA STOPNJA ORGANIZACIJE PODATKOV.	1-5
2. PODATKOVNE BAZE.	2-1
2.1 UVOD V PODATKOVNE BAZE.	2-1
2.2 CILJI IN LASTNOSTI PODATKOVNIH BAZ.	2-5
2.2.1 VEČNAMENSKOST PODATKOV.	2-5
2.2.2 ZAŠČITA PRETEKLEGA DELA	2-5
2.2.3 JASNOST, LAHKA IN FLEKSIBILNA UPORABA	2-6
2.2.4 ENOSTAVNE SPREMEMBE	2-6
2.2.5 NIZKA CENA.	2-6
2.2.6 REDUNDANCA.	2-6
2.2.7 IZVRŠITEV ZAHTEV.	2-7
2.2.8 TOČNOST IN KONSISTENTNOST	2-7
2.2.9 PRIVATNOST.	2-7
2.2.10 ZAŠČITA PRED UNICENJEM ALI POŠKODBO	2-7
2.2.11 RAZPOLOŽLJIVOST	2-8
2.2.12 FIZIČNA IN LOGIČNA NEODVISNOST.	2-8
2.2.13 HITROST DOSTOPA IN ISKANJA.	2-8
2.2.14 STANDARDIZACIJA PODATKOV.	2-8
2.2.15 PODATKOVNI SLOVAR	2-9
2.2.16 VISOKONIVOJSKI PROGRAMERSKI VMESNIK	2-9
2.2.17 JEZIK ZA KONČNEGA UPORABNIKA.	2-9
2.2.18 KONTROLA INTEGRITETE.	2-10
2.2.19 ORODJA ZA NAČRTOVANJE IN UREJANJE	2-10
2.2.20 AVTOMATSKA REORGANIZACIJA	2-10
2.2.21 RAZVOJ V SMERI DISTRIBUIRANIH OBDELAV	2-10
2.3 NEODVISNOST PODATKOV.	2-11
2.4 VRSTE PODATKOVNIH BAZ	2-13
3. SODOBNE METODE ZA RAZVOJ RAČUNALNIŠKO PODPRTIH APLIKACIJ.	3-1
3.1 INFORMACIJSKO INŽENIRSTVO	3-2
3.1.1 STRATEŠKO PLANIRANJE POTREB	3-3
3.1.2 INFORMACIJSKA ANALIZA	3-4
3.1.3 OBLIKOVANJE PODATKOVNEGA MODELA	3-4
3.2 SODOBNI PRIPOMOČKI ZA RAZVOJ APLIKACIJ.	3-5

UVOD

Za naš svet je značilna hitra rast stopnje razvoja proizvodnih sil. V takem procesu prihaja tudi do nagle rasti informacij, ki odražajo celotno znanje človeštva. Če hočemo iti v korak s svetom, moramo biti z vsem na tekočem, logično pa je, da je pri taki dinamiki nemogoče spremljati vse dosežke, pa čeprav se omejimo le na neko ožje področje. Vse to vodi v informacijsko krizo, saj so dandanes pomembne hitre in učinkovite informacije. Teh nam klasičen način obdelovanja podatkov (AOP) z obsežnimi papirnimi "outputi", praktično neuporabnimi za končnega uporabnika, ki potrebuje hitro in učinkovito informacijo za odločanje in upravljanje na vseh ravneh, ne more nuditi.

V iskanju rešitve iz informacijske krize je prišlo v računalništvu do razvoja nove veje - to je INFORMATIKA. Ta pojem je nastal iz dveh francoskih besed: informacija in avtomatika (INFORMation automATIQUE).

Informatika raziskuje lastnosti in obnašanje informacij za optimalen pristop in uporabo. Ukvarja se z nastajanjem, zbiranjem, hranjenjem, organizacijo, interpretiranjem, pretvarjanjem in uporabo informacije ter z raziskovanjem izvedbe

in delovanja računalniško podprtih sistemov. Za vse te pa je potrebna ustrezna informacijska podlaga.

Ena takih informacijskih podlag je tudi PODATKOVNA BAZA. Podatki so namreč mnogo stabilnejši od postopkov, ki jih uporabljajo. Podatki neke organizacijske strukture se na primer spremenijo le, če se v temeljih spremeni vrsta poslovanja, postopki pa se v današnji dinamični družbi veliko hitreje spreminjajo. Podatki v podatkovni bazi so organizirani neodvisno od postopkov, z njihovim enkratnim zajemanjem pa se izognemo netočnosti in bistveno zmanjšamo redundanco. Relacije (povezave) med podatki so grajene na njihovih notranjih (inherentnih) lastnostih (tako kot po naravi sodijo skupaj). Enkrat zapisani podatek lahko uporablja več postopkov na različne načine.

Za razvoj kompleksnega informacijskega sistema z integrirano podatkovno bazo moramo temeljito analizirati in proučiti potrebe neke informacijske strukture. Tak pristop imenujemo informacijsko izenirstvo, ki je razdeljeno v več faz. Čimveč skušamo uporabljati sodobne pripomočke za razvoj računalniško podprtih aplikacij, kajti le na ta način bomo računalnik lahko približali končnemu uporabniku in sčasoma prebili masično bariero klasičnih računskih centrov, kjer so delali le za "programiranje poklicani" ljudje.

1. ORGANIZACIJA PODATKOV

Ustrezna organizacija podatkov v informacijskih sistemih je ključnega pomena za njihovo uspešno delovanje. Še zlasti je pomembno čim bolj racionalno zbiranje, hranjenje, obdelava in prikazovanje podatkov. Čeprav sistemi za delo s podatkovnimi bazami - DBMS (Data Base Management System) obstojajo že dokaj dolgo, pa se njihova uporaba ni razmahnila do take mere, kot je bilo pričakovati. Vzrok za to je najbrž predvsem v zahtevi po določeni reorganizaciji obstoječega delovanja, zato so bile bolj pogoste manj zahtevne rešitve, ki niso zahtevale toliko sprememb. To je imelo za posledico dezintegracijo podatkov, podvajanje, nekonsistentnost in pomanjkanje integritete na ravni celovitih organizacijskih sistemov.

Glede na stopnjo organiziranosti podatkov ločimo štiri stopnje:

1.1. PRVA STOPNJA ORGANIZACIJE PODATKOV

Za to stopnjo so značilne datoteke, ki pokrivajo posamezne aplikacije. Pri tem obstaja močna odvisnost podatkov od aplikacijskih programov, ki te podatke uporabljajo. Podatke

definira in zajema programer, ki dela posamezno aplikacijo. Pravidoma ni nobene interakcije med aplikacijami, tako da smo priča velikemu naraščanju tako števila datotek, kakor tudi podatkov v njih. Ta način organizacije podatkov je nedvomno najenostavnejši. Programske rešitve na tej stopnji so večinoma na nivoju fizične organizacije podatkov.

Pri tej stopnji smo ponavadi priča problemom pri naraščanju števila aplikacij, kar sproži naraščanje števila datotek, to pa poveča redundanco podatkov. Na tej stopnji ni nobene povezave med posameznimi datotekami, ki jih uporabljajo različne aplikacije, zato smo priča pojavi nekonsistentnosti podatkov, kajti ob vsaki spremembi posameznega podatka v neki datoteki, bi bilo treba tega spremeniti še v vseh ostalih datotekah, to pa je pri velikem številu datotek skoraj nemogoče obvladovati, saj za vsako datoteko skrbi samo avtor aplikacije, ki jo uporablja, ta pa večinoma nima dostopa do ostalih datotek.

Poleg tega je tak sistem organizacije podatkov tudi zelo dras, kajti vsako spreminjanje zahteva, da vsi programerji spreminjajo v vseh svojih datotekah spremenjen podatek. Visoka je tudi cena dodatnih pomnilniških medijev, ki jih je po določenem času uporabe take organizacije podatkov nužno nabaviti, kajti osnovna konfiguracija postane kaj kmalu nezadostna za hranjenje take količine podatkov.

1.2. DRUGA STOPNJA ORGANIZACIJE PODATKOV

Na tej stopnji se že uporabljajo programski sistemi za delo s podatkovnimi bazami (DBMS - Data Base Management Systems), vendar pa še ni dosežena osnovna zahteva za delo s podatkovnimi bazami, namreč podatkovno orientirani postopki, ampak smo še vedno priča postopkovno orientiranim podatkom.

Edina razlika med prvo in drugo stopnjo organizacije podatkov je v tem, da je zdaj olajšano delo s podatki. Podatkovne baze se namreč gradijo za vsako aplikacijo posebej, torej med posameznimi podatki spet ni nobene povezave, kar potegne za seboj vse slabosti, naštetih že pri prvi stopnji organizacije podatkov. Zaradi naraščanja števila podatkovnih baz s številom aplikacij, ki jih uporabljajo, je vzdrževanje sistemov na tej stopnji organiziranosti izredno drago in zahtevno.

1.3. TRETJA STOPNJA ORGANIZACIJE PODATKOV

Podatki na tej stopnji so opredeljeni in shranjeni ločeno od njihove uporabe. Podatki so večnamenski in si jih deli več uporabnikov oziroma aplikacij. Za izgradnjo take baze je potrebna skrbna informacijska analiza, saj moramo zgraditi stabilen podatkovni model na ravni celotne organizacijske strukture institucije. Na tej stopnji organizacije podatkov je torej prvič

dosežen cilj podatkovne baze: neodvisnosti podatkov od aplikacij, neredundantnost podatkov, njihova večnamenskost in konsistentnost. Stroški vzdrževanja podatkov se vidno zmanjšajo, kajti posamezni podatki se ažurirajo samo enkrat na vseh mestih, kjer nastopajo.

Na tej stopnji organiziranosti podatkov so nujne nekatere organizacijske spremembe v samem delu računskega centra. Glede na to, da so zdaj podatki enotni za vse uporabnike in shranjeni na enem mestu, je nujno imeti administratorja podatkovne baze, katerega naloga je, da skrbi, da delo z bazo podatkov poteka nemoteno, da so podatki v njej točni in konsistentni in da po potrebah uporabnikov dodaja v bazo nove strukture podatkov in povezave med njimi. Če upravljanje s podatkovno bazo ni izvedeno pravilno ali če baza ni bila dobro načrtovana, se na tej stopnji pogosto lahko zgodi, da zasnovana podatkovna baza preide na drugo ali celo prvo stopnjo organiziranosti.

Baze podatkov na tej stopnji organizacije podatkov so največkrat v uporabi v produkcijskih sistemih, kjer se zahteva velika hitrost in učinkovitost pri zajemu podatkov, ažuriranju in manipuliranju z njimi, sam dostop do podatkov pa tu nima tolikšnega pomena.

1.4. ČETRTO STOPNJA ORGANIZACIJE PODATKOV

Za četrto stopnjo organizacije podatkov je značilna potreba po hitrem dostopu do informacij, po čimbolj enostavnem iskanju. Največji poudarek na tej stopnji je na tem, da informacije uporabljamo za podporo odločanju, to pomeni, da je pomembno enostavno oblikovanje vprašanj, poročil in da je odziv na naše zahteve čimhitrejši. Osnova za organizacijo podatkov na tej stopnji je podatkovna baza na tretji stopnji organiziranosti, kar daje podporo produkcijskim sistemom, za nove potrebe pa jo je treba delno preurediti, potrebno je narediti izbor (ekstrakcijo) tistih podatkov, do katerih želimo imeti hiter dostop ali jih zelo pogosto uporabljamo za oblikovanje poročil oziroma podporo odločanju. Da to dosežemo, se največkrat poslužujemo tvorbe invertiranih in indeksnih datotek, preko katerih do posameznih podatkov pridemo hitreje in enostavneje.

Invertirane datoteke so datoteke, v katerih so podatki urejeni po nekem drugem ključu kakor sicer v podatkovni bazi, v indeksnih datotekah pa imamo indekse oziroma naslove posameznih skupin podatkov v bazi, da lahko hitreje pridemo do želenega podatka. Primera indeksne in invertirane datoteke sta prikazana na slikah 1.1 in 1.2

01 02 03 04 05 06 07 08 09 10 11
 ! K1 ! K2 ! K3 ! K4 ! K5 ! K6 ! K7 ! K8 ! K9 ! K10 ! K11 !

K1	01
K5	05
K10	10
:	:

Slika 1.1 Indeksna datoteka

K1	RIBIČ	IGOR	ČRNUČE 106	LJUBLJANA	Z750
K2	KLINAR	HELENA	TITOVA 16	ČRNOMELJ	Z101
K3	FICKO	PIKA	CANKARJEVA 10	M. SOBOTA	DYANE
K4	LAZAR	SIMON	KOSOVELOVA 10	CELJE	MZ250
K5	KOŠIR	IVAN	ŽUGOMERJEVA 2	LJUBLJANA	GOLF
K6	LAMPE	JOŽE	ROŽNA 23	ČRNOMELJ	Z101

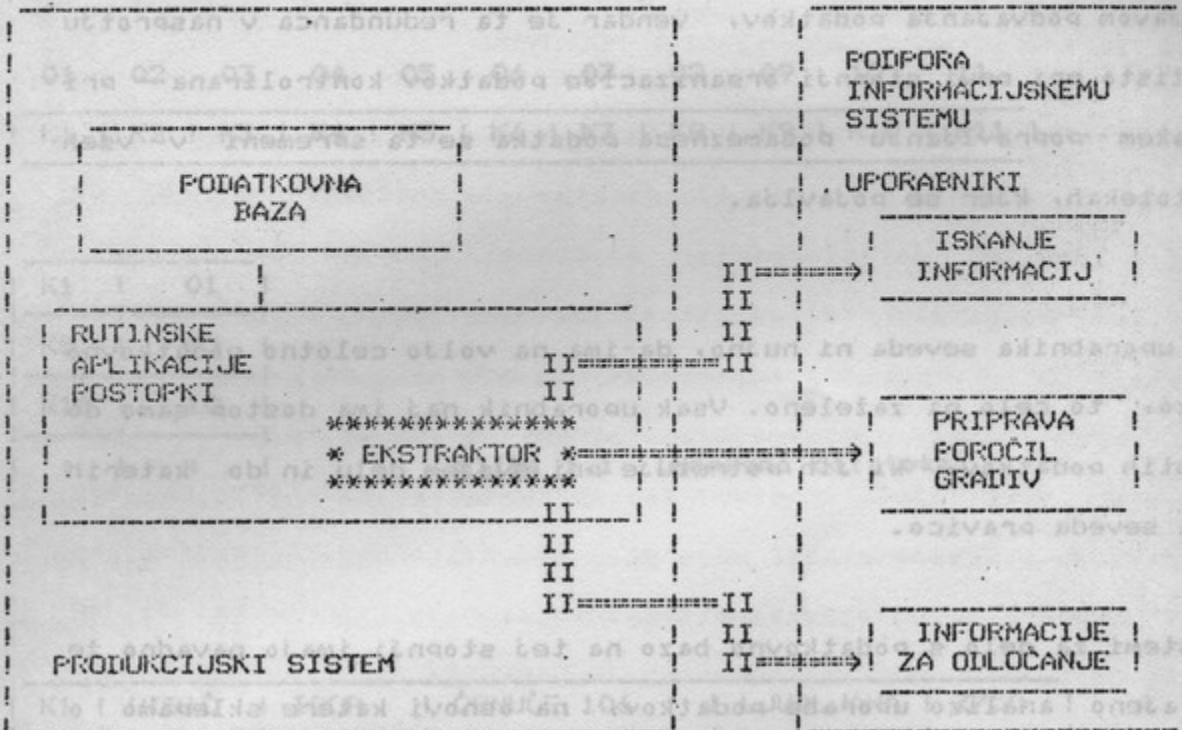
Z750	K1	LJUBLJANA	K1	K5
Z101	K2	ČRNOMELJ	K2	K6
DYANE	K3	M. SOBOTA	K3	
MZ250	K4	CELJE	K4	
GOLF	K5			

Slika 1.2 Invertirana datoteka

Na tej stopnji organiziranosti podatkov se zopet srečamo s pojavom podvajanja podatkov, vendar je ta redundanca v nasprotju s tisto pri prvi stopnji organizacije podatkov kontrolirana - pri vsakem popravljanju posameznega podatka se ta spremeni v vseh datotekah, kjer se pojavlja.

Za uporabnika seveda ni nujno, da ima na voljo celotno podatkovno bazo, to celo ni zaželeno. Usak uporabnik naj ima dostop samo do tistih podatkov, ki jih potrebuje pri svojem delu in do katerih ima seveda pravico.

Sistemi za delo s podatkovno bazo na tej stopnji imajo navadno že verajeno analize uporabe podatkov, na osnovi katere sklepamo o največ uporabljenih podatkih in te nato vključimo v ekstraktor, če še niso, seveda šele potem, ko dobimo privoljenje uporabnikov. Shema take organizacije podatkov prikazuje slika 1.3.



Slika 1.3 Ekstrakcija podatkov na četrtni stopnji organizacije podatkov

Razvoj gre v zadnjem času v smeri distribucije oziroma selitve obdelav k posameznim uporabnikom, pa naj bo to preko inteligentnih terminalov ali mikroračunalnikov. To pa zahteva tudi distribucijo podatkovne baze oziroma distribucijo podatkov v njej do končnih uporabnikov. Vsak uporabnik naj bi torej imel na svojem računalniku samo tiste podatke, ki jih uporablja pogosteje, do ostalih pa naj bi prišel preko omrežja. To pa postavlja zopet nove zahteve pred konstruktorje sistemov za upravljanje s podatkovnimi bazami, drugačne so zahteve glede upravljanja s podatki in zagotovitve konsistentnosti podatkov med

lokalnimi bazami in skupno podatkovno bazo.

Pri odločanju o tem, na kateri stopnji bomo imeli organizirane podatke, je zelo važno poznavati vse zakonitosti in informacijske potrebe področja, za katerega gradimo informacijski sistem. Aplikacije, ki zaradi svoje specifičnosti brez problemov tečejo na sedanji stopnji organizacije podatkov, bi bilo nesmiselno na silo reorganizirati na višjo stopnjo, prav tako je za stroso produkcijske sisteme nesmiselna organizacija podatkov na četrthi stopnji, ker je zanje tretja stopnja organizacije dovolj primerna glede na hitrost organizacije podatkov, pa še manj prostora nam ti podatki zasedejo v pomnilniku.

Produktivnost programerjev narasla le za 15%. Zastajanje pri razvijanju aplikacij je za okoli 100% za evidentiranimi potrebami. Če upoštevamo še vse potencialne, toda doslej še neizpolnjene potrebe.

Na področju programiranja razvili metode, ki naj bi povečale produktivnost programerjev in omogočile čim bolj racionalno in prehodno programiranje. Te metode so:

- strukturirana analiza
- strukturiran design
- strukturirano programiranje

Strukturirana analiza je orientirana v postopke. Nek zaključeno in celotno videlo lesično razgrajujemo na posamezne področja, ki se nato znova, zopet zaključene celote, dokler nimamo celotnega videla razgrajenega na elementarne podprobleme.

2. PODATKOVNE BAZE

2.1 UVOD V PODATKOVNE BAZE

Na razvoj računalniške tehnologije v zadnjih letih je zlasti vplival zelo hiter razvoj strojne opreme, ki mu na žalost ni sledil tudi razvoj programske opreme. Mikroelektronski elementi so v teh letih postali za faktor 1.000 do 10.000 hitrejši, zmogljivejši, zanesljivejši in cenejši, v istem obdobju pa je produktivnost programerjev narasla le za 15%. Zaostajanje pri izdelavi aplikacij ja za okoli 100% za evidentiranimi potrebami, še večji pa je, če upoštevamo še vse potencialne, toda doslej še neevidentirane potrebe.

V ta namen so na področju programiranja razvili metode, ki naj bi povečale produktivnost programerjev in omogočile čim bolj racionalno in presledno programiranje. Te metode so:

- strukturirana analiza
- strukturiran design
- strukturirano programiranje

Strukturirana analiza je orientirana v postopke. Nek zaključen in kompleksen problem logično razgrajujemo na posamezne module, ki so sami zase zopet zaključene celote, dokler nimamo celotnega problema razdeljenega na elementarne podprobleme.

V fazi strukturiranega designa za vsak posamezen proces oziroma elementaren podproblem opredelim programski modul. Za vsak tak modul moramo opredeliti oblike vhodov in izhodov, vse povezave med posameznimi moduli in ustrezne posoje za izvajanje. Vsak programski modul mora biti čim bolj neodvisen zaradi lažjega obvladovanja, popravljanja in vzdrževanja, zato je zelo pomembno, da je strukturirana analiza opravljena kar najbolj natančno. Tako dobimo natančen načrt programskega paketa za problem, ki ga moramo obvladati.

Strukturirano programiranje pomeni formaliziran postopek programiranja. Programske module, ki smo jih definirali v strukturirani analizi in jim opredelili povezave v fazi strukturiranega designa, sprogramiramo in združimo v celoto, ki predstavlja rešitev zastavljenega problema.

Na ta način se je produktivnost pri razvoju aplikacij sicer povečala, vendar samo za kakih 15 do 20%, kar še vedno pomeni precejšnje zaostajanje programske opreme v primerjavi s strojno.

Glavna značilnost teh metod je njihova postopkovna orientiranost, to pomeni da podatke prirejamo postopkom. S tem pride do velikega podvajanja podatkov, saj moramo podatke prirejati za vsak postopek posebej, ne glede na to, če so ti podatki morda že zapisani v datotekah, ki jih uporablja kakšna druga aplikacija. Glavna značilnost teh metod je torej veliko podvajanje (redundanca) podatkov.

Postopki pa so močno podvrženi spremembam in ob vsaki spremembi posameznega postopka je treba spremeniti tudi organizacijo podatkov, ki jih ta postopek uporablja. Poleg tega isti podatki za različne postopke nastopajo na različnih mestih in kontrola konsistentnosti podatkov je zelo težka, če že ne nemogoča. Zgodi se nam torej lahko, da imajo isti podatki na različnih mestih popolnoma nepravilne vrednosti, kajti ob spremembi podatka v eni datoteki bi se morali spremeniti v vseh, to pa je le redkokdaj možno, saj različne aplikacije uporabljajo različni uporabniki.

Taka organizacija podatkov porabi torej zelo veliko dodatnega prostora v pomnilniku, pa tudi časa, zelo velika pa je verjetnost, da so podatki nekonsistentni, na tak način pa uporabnika, ki si želi hitrih in točnih informacij, ne moremo zadovoljiti.

Prav zaradi vseh teh slabosti pomeni uporaba podatkovnih baz pravo revolucijo na področju računalništva in informatike. Kratka definicija podatkovne baze bi se glasila: podatkovna baza je zbirka med seboj pomensko povezanih podatkov, ki so shranjeni na skupnem mestu brez škodljivega ali nepotrebne podvajanja in so namenjeni različnim vrstam uporabe. Podatki so neodvisni od programov, ki jih uporabljajo. Za dostop do obstoječih podatkov, njihovo modificiranje in dodajanje se uporablja skupen in kontroliran pristop. Podatki v podatkovni bazi so tako strukturirani, da omogočajo nadaljnjo rast podatkovne baze, kakor tudi širjenje njene uporabe. Prednosti, ki jih imajo podatkovne

baze pred datotečno organizacijo podatkov, so:

- podatkovna baza je osnova za razvoj aplikacij
- podatki v njej so večnamenski (uporablja jih več uporabnikov)
- obstoječi aplikacijski programi se ne spremenijo, če pride do sprememb v podatkovni bazi
- jasnost pri delu s podatki in enostavnost uporabe podatkov
- fleksibilnost
- možnost za spontana vprašanja
- enostavne spremembe v podatkovni bazi
- cenenost (računalniški prostor, spreminjanje podatkov, aplikacij)
- ni eksplozivne rasti podatkov (manjša redundanca)
- dobre tehnične karakteristike
- natančnost in konzistentnost podatkov
- privatnost in zaščita
- fizična in logična neodvisnost podatkov
- kontrolirana redundanca
- hiter dostop do podatkov
- standardizacija podatkov
- jeziki 4. generacije
- hitra rekonstrukcija
- možnost izboljšanja lastnosti delovanja podatkovne baze
- avtomatizacija načrtovanja podatkovne baze
- možnosti distribucije

2.2 CILJI IN LASTNOSTI PODATKOVNIH BAZ

Podatkovna baza je temeljni kamen razvoja aplikacij v prihodnosti. Z njeno pomočjo bo razvoj aplikacij lažji, hitrejši, cenejši in bolj fleksibilen. O ciljih in lastnostih podatkovnih baz ne moremo govoriti ločeno, kajti cilj je v nekem smislu že tudi lastnost, ko je vračen v sistem za delo s podatkovno bazo. Najosnovnejši cilj je seveda ta, naj bo podatkovna baza shramba podatkov, ki so potrebni za neko organizacijo (organizacija v tem smislu je sistem, katerega delovanje bazira na uporabi informacij). Iz njega so seveda izpeljani vsi ostali cilji in lastnosti.

2.2.1 VEČNAMENSKOST PODATKOV

Različni uporabniki imajo lahko popolnoma različne poglede na iste podatke v podatkovni bazi in jih tudi uporabljajo na različne načine in za različne potrebe.

2.2.2 ZAŠČITA PRETEKLEGA DELA

V primeru, da spreminjamo podatkovno bazo, ne sme to imeti nobenega vpliva na obstoječe programe, ki to podatkovno bazo uporabljajo, ali na losične podatkovne strukture, v katere je bilo vložena veliko človeškega dela.

2.2.3 JASNOST, LAHKA IN FLEKSIBILNA UPORABA

Uporabnik naj na lahek in jasen način izve, kateri podatki so mu dostopni. Dostop do podatkov mora biti čimbolj enostaven. Kompleksnost zgradbe podatkovne baze mora ostati uporabniku skrita, komunicira naj le s sistemom za delo s podatkovnimi bazami. Dostop do podatkov naj bo čimbolj fleksibilen z možnostjo uporabe različnih dostopnih poti. Odgovori na nepričakovana vprašanja naj bodo hitri, z uporabo visokonivojskih jezikov za iskanje ali generatorjev poročil.

2.2.4 ENOSTAVNE SPREMEMBE

Podatkovna baza lahko narašča ali se spreminja, ne da bi to vplivalo na dosedanje načine uporabe podatkov.

2.2.5 NIZKA CENA

Cena shranjevanja in uporabe podatkov je nizka, cena drasih sprememb pa minimizirana, kolikor je to mogoče.

2.2.6 REDUNDANCA

Vsi aplikacijski programi uporabljajo iste podatke, zato ni potrebe po ustvarjanju novih datotek. Na ta način se izognemo redundantnosti podatkov, ki predstavljajo velik problem v datotečni organizaciji podatkov. To pomeni, da je vsak podatek v bazi shranjen samo enkrat, razen v primeru, da obstajajo tehnični

oziroma ekonomski razlogi za redundantnost, ki pa je v tem primeru zaželena in kontrolirana.

2.2.7 IZVRŠITEV ZAHTEV

Zahteve po podatkih so izvršene s hitrostjo, primerno uporabi podatkov.

2.2.8 TOČNOST IN KONSISTENTNOST

V bazi podatkov bo uporabljena kontrola točnosti. Sistem za delo s podatkovno bazo ne bo dovolil v bazi nastopa istih podatkov na različnih stopnjah ažuriranosti.

2.2.9 PRIVATNOST

Uporabniki, ki niso avtorizirani za dostop do posameznih podatkov, do njih ne bodo imeli dostopa. Isti podatki so lahko istočasno zaščiteni na različnih stopnjah proti različnim uporabnikom.

2.2.10 ZAŠČITA PRED UNIČENJEM ALI POŠKODBO

Podatki so zaščiteni pred ognjem, tatovi in drugimi oblikami uničenja. Obstajati mora možnost njihove rekonstrukcije. Urejena mora biti kontrola podatkov (nepravilno spreminjanje), kontrola dostopa (sistem mora biti tako zaščiten, da se ne da obiti pristopne kontrole), uporabnikove akcije morajo biti nadzorovane.

Urejena mora biti avtomatična rekonstrukcija podatkov, če se pojavi napaka ali poškodba v bazi.

2.2.11 RAZPOLOŽLJIVOST

Podatki so hitro na voljo uporabniku ob vsakem času, ko jih le-ta potrebuje.

2.2.12 FIZIČNA IN LOGIČNA NEODVISNOST

Podatki v podatkovni bazi so fizično in logično neodvisni, to pomeni, da spremembe v fizični organizaciji podatkov ne zahtevajo sprememb obstoječih programov, prav tako to velja tudi za dodajanje novih tipov podatkov v bazo.

2.2.13 HITROST DOSTOPA IN ISKANJA

Fristopni mehanizmi in metode naslavljanja morajo biti dovolj hitri, da je možna uporaba podatkovne baze za iskanje po metodi vprašanje - odgovor. Zato je potrebno hitro iskanje po podatkovni bazi. Predvsem je važno, da so ti sistemi zasnovani na interaktivni osnovi.

2.2.14 STANDARDIZACIJA PODATKOV

Potreba po standardizaciji znotraj organizacije, ki bo uporabljala isti podatkovni model, je zelo velika. Med posameznimi enotami so potrebni natančni dogovori o formatih in

definicijah podatkov, sicer je velika verjetnost, da bodo podatki, definirani po posameznih enotah, med seboj nekompatibilni.

2.2.15 PODATKOVNI SLOVAR

Za lažjo uporabo podatkovne baze je nujno potreben podatkovni slovar, v katerem so definirani vsi podatki (atributi), ki v bazi nastopajo.

2.2.16 VISOKONIVOJSKI PROGRAMERSKI VMESNIK

Za uporabo podatkov iz podatkovne baze v aplikacijskih programih mora obstajati programerski vmesnik, ki bo programerju aplikacij omogočil uporabo enostavnih in dovolj močnih ukazov in mu prihranil uporabo kompleksno zgrajenih datotek in kompliciranega adresiranja.

2.2.17 JEZIK ZA KONČNEGA UPORABNIKA

Končnemu uporabniku je treba omogočiti, da v nekaterih primerih obide fazo aplikacijskega programiranja in dobi rezultat svoje zahteve po lažji poti. V ta namen služita jezik za iskanje po podatkovni bazi (query language) in generator poročil (report generator).

2.2.18 KONTROLA INTEGRITETE

Že pri vnosu podatkov mora biti opravljena (kjer je to mogoče) kontrola tipa, dolžine, in drugih elementov, ki jih je možno definirati in kontrolirati.

2.2.19 ORODJA ZA NAČRTOVANJE IN UREJANJE

Administratorju podatkovne baze ali njenemu načrtovalcu morajo biti na voljo orodja, ki mu omogočajo ocenjevanje in izboljševanje lastnosti baze.

2.2.20 AVTOMATSKA REORGANIZACIJA

Če hočemo izboljšati lastnosti že obstoječe podatkovne baze, morajo obstajati orodja za avtomatično reorganizacijo in prenos podatkov.

2.2.21 RAZVOJ V SMERI DISTRIBUIRANIH OBDELAV

Sistem za delo s podatkovno bazo naj omogoča razvoj v tej smeri, da podpira tudi distribuirane obdelave.

2.2.18 - KONTROLA INTEGRITETE

ni spre- memb v aplik. progr.	ni spre- memb v log. shemi	ni spre- memb v fiz. org.	
+	+	+	-dodamo nov aplikacijski program, ki uporablja obstoječe podatke
+	+	+	-aplikacijski program zahteva drugačno predstavitev obstoječih podatkov
+			-dodamo nov aplikacijski program z novimi podatki
+	+	+	-sprememba vrednosti podatkov.
+			-sprememba logične organizacije (dodamo nove povezave)
+			-združitve dveh podatkovnih baz
+	+		-izboljšanje fizične organizacije
+	+		-premestitev podatkov na drugo V/I enoto
+	+	+	-sprememba programske ali strojne opreme za DBMS

Slika 2.2 Vpliv sprememb na vse tri poglede

Različni standardi imenujejo te tri poglede različno, zato je prav, da imena omenimo (slika 2.3).

	ANSI/SPARC	CODASYL	IBM
posled končne- sa uporabnika	eksterna shema	subschema	program specifi- cation block
globalni pogled	konceptualna shema	schema	logical DB description
fizični pogled	interna shema	physical data	physical DB description

Slika 2.3 Imena posledov po posameznih standardih

2.4 VRSTE PODATKOVNIH BAZ

Sistemi za delo s podatkovnimi bazami naredijo fizično zgradbo podatkovne baze uporabniku sicer transparentno, vendar je prav, da poznamo vsaj osnove organizacije naših podatkov v bazi. Osnovni gradnik podatkovne baze je zapis (entiteta), ki je sestavljen iz enega ali več ključev, po katerih ta zapis iščemo, in atributov, ki ga opisujejo. Slika 2.4 prikazuje opis zapisa Občan in ta zapis z različnimi vrednostmi.

OBČAN

! mat.št.obč.!	! priimek !	! ime !	! mestu !	! ulica št. !
! 2313456234 !	! HVALA !	! BOJAN !	! LJUBLJANA !	! CANKARJEVA 3 !
! 1267453590 !	! GANTAR !	! JANA !	! CELJE !	! GOTSKA 16 !
! 0978046723 !	! ČUK !	! TOMAŽ !	! MARIBOR !	! CELOVŠKA 123 !

Slika 2.4 Opis entitete in entiteta z vrednostmi

Logično zgradbo podatkovne baze opišemo z opisom zapisov in povezav med njimi. Te delimo na več različnih tipov:

- ENA : ENA enemu zapisu prvega tipa pripada natanko en zapis drugega tipa in obratno
- ENA : MNOGO enemu zapisu prvega tipa pripada več zapisov drugega tipa in enemu zapisu drugega tipa pripada natanko en zapis prvega tipa
- MNOGO : ENA enemu zapisu prvega tipa pripada natanko en

zapis drugega tipa in enemu zapisu drugega tipa
pripada več zapisov prvega tipa

- MNOGO : MNOGO enemu zapisu prvega tipa pripada več zapisov
drugega tipa in obratno

Z uporabo teh in še nekaterih drugih pravil zgradimo nato podatkovni model, ki predstavlja logično zgradbo podatkovne baze. Fizična zgradba nas kot uporabnike ne zanima posebno, važno je le, da vemo kateri tip podatkovne baze uporabljamo. Ločimo tri tipe podatkovnih baz, ki se razlikujejo po organizaciji povezav med posameznimi zapisi:

- relacijska podatkovna baza
- mrežna podatkovna baza
- hierarhična podatkovna baza

Relacijske podatkovne baze so navadno najbolj priljubljene, ker so nam najbližje glede na naše razmišljanje. Podatki v njih so namreč urejeni podobno kot v tabelah, zato si jih ni težko predstavljati.

Hierarhične baze so zrajane na osnovi drevesne strukture, podobne njim so mrežne, le da te dovoljujejo tudi to, da ima en otrok več očetov, kar v hierarhični bazi ni mogoče.

Za relacijske baze je značilna velika fleksibilnost pri oblikovanju baz podatkov, medtem ko so hierarhične in mrežne baze učinkovitejše in hitrejšje pri obdelavi podatkov.

3. SODOBNE METODE ZA RAZVOJ RAČUNALNIŠKO PODPRTIH APLIKACIJ

Prvotna raba računalnikov se je omejevala zlasti na računanje zapletenih problemov s področja matematike, mehanike, fizike. To je značilnost najzgodnejšega obdobja uporabe računalnikov. Naslednja faza, za katero je karakteristična avtomatska obdelava podatkov, je imela za posledico razvoj produkcijskih računalniških sistemov s poudarkom na hitri in učinkoviti obdelavi velike množice podatkov v zelo formaliziranih sistemih zlasti s področij knjigovodstva, računovodstva, bančnega poslovanja itd.

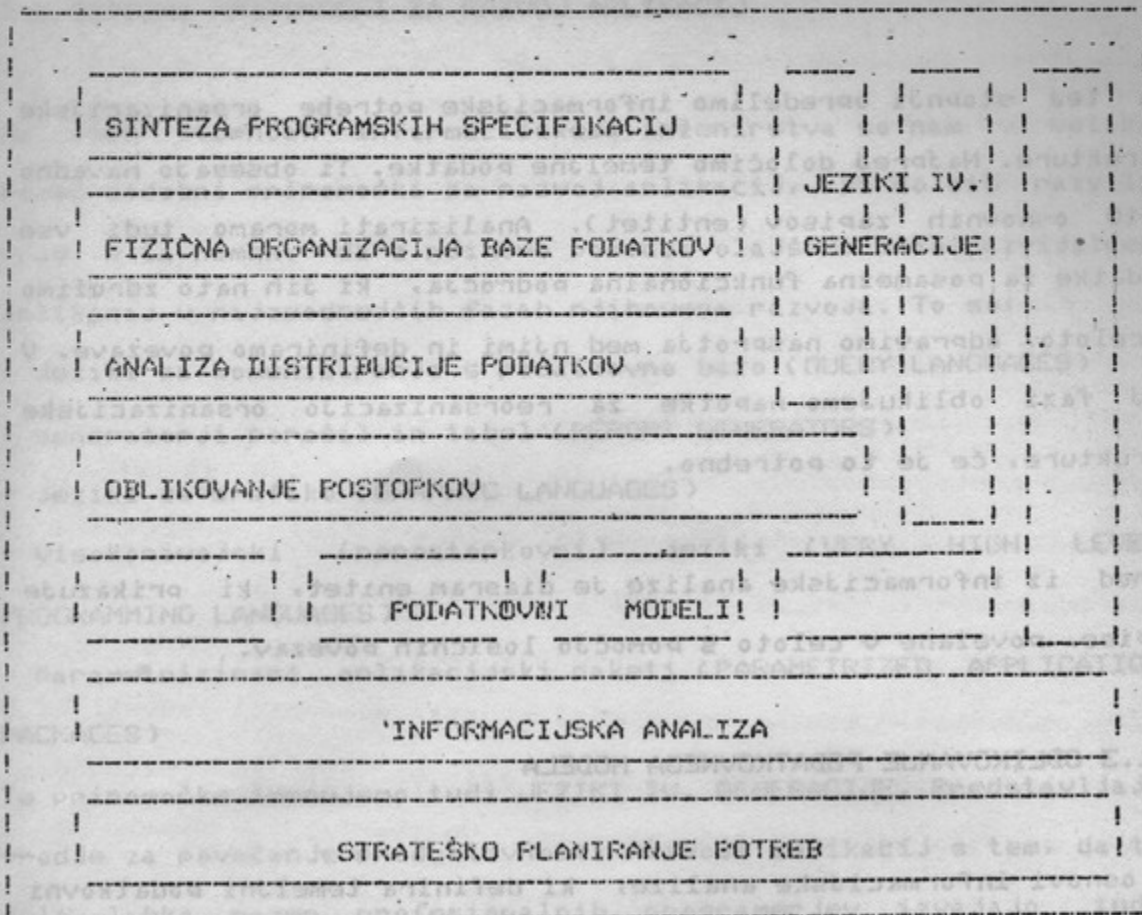
Razvoj računalništva je privedel do zahteve po izgradnji kompleksnejših sistemov, po premiku od produkcijskih k informacijskim sistemom za potrebe upravljanja in odločanja na vseh ravneh. To pa je bilo onemogočeno predvsem zaradi pomanjkanja sodobnih programskih metod zlasti za komuniciranje z bazami podatkov in interaktivni dialog z računalnikom.

V zadnjem času pa gre razvoj prav smeri izgradnje teh sodobnih pripomočkov, ki po eni strani omogočajo veliko povečanje produktivnosti razvijalcev aplikacij, prav tako pa tudi uporabo računalnika s strani končnih uporabnikov brez računalniškega predznanja.

3.1 INFORMACIJSKO INŽENIRSTVO

Pri konvencionalnem razvoju aplikacij je prihajalo do največjih težav prav v prvih fazah njihovega razvoja. Te faze so dolgotrajne, zahtevajo veliko intenzivnega dela in prav v njih so težave najposostejše, poleg tega pa je te težave najtežje odpravljati. Tudi strukturirana analiza in strukturiran design teh težav nista odpravila, posebno še pri trajanju dela zaradi zahtev po kompleksni spremljajoči dokumentaciji in zaradi premajhnega odziva uporabnikov, ki niso dovolj sodelovali pri razvoju aplikacij. Zato so se v zadnjih letih precej razmahnile metode, ki naj z avtomatizacijo pomagajo pri razvoju aplikacij tudi v njihovem najzgodnejšem obdobju. Faze sodobnih metod za razvoj informacijskih sistemov, ki jih imenujemo informacijsko inženirstvo, prikazuje slika 3.1.

Temeljna razlika med strukturiranimi metodami in metodami informacijskega inženirstva je v tem, da gre tu za izgradnjo sistema od vrha navzdol (topdown), za osnovo je strateško planiranje potreb.



Slika 3.1 Informacijsko inženirstvo

3.1.1 STRATEŠKO PLANIRANJE POTREB

V tej prvi fazi moramo opredeliti namen in cilj organizacije, za katero želimo zgraditi informacijski sistem. Ugotoviti moramo:

- namen organizacije
- proizvodi in storitve
- stranke
- razvojni načrti

3.1.2 INFORMACIJSKA ANALIZA

Na tej stopnji opredelimo informacijske potrebe organizacijske strukture. Najprej določimo temeljne podatke. Ti obsegajo navadno 5-10 osnovnih zapisov (entitet). Analizirati moramo tudi vse podatke za posamezna funkcionalna področja, ki jih nato združimo v celoto, odpravimo nasprotja med njimi in definiramo povezave. V tej fazi oblikujemo napotke za reorganizacijo organizacijske strukture, če je to potrebno.

Izhod iz informacijske analize je diagram entitet, ki prikazuje zapise, povezane v celoto s pomočjo logičnih povezav.

3.1.3 OBLIKOVANJE PODATKOVNEGA MODELA

Na osnovi informacijske analize, ki definira temeljni podatkovni model za neko organizacijo, naredimo podrobnejše podatkovne modele, ki upoštevajo vse nastopajoče zapise, opredelimo njihove attribute, ključe (atribut, ki enolično določa zapis) in podrobne opise vseh podatkov glede načina zapisa, velikosti, frekvence itd.

Te faze so osnova za oblikovanje postopkov, analizo rabe in distribucije podatkov, pripravo fizične organizacije podatkov in programskih specifikacij.

3.2 SODOBNI PRIPOMOČKI ZA RAZVOJ APLIKACIJ

Na vseh stopnjah informacijskega inženirstva so nam v veliko pomoč sodobni pripomočki za razvoj aplikacij, ki so jih razvili prav v ta namen, da z njihovo pomočjo olajšamo delo razvijalcem aplikacij v najzgodnejših fazah njihovega razvoja. To so:

- jeziki za komuniciranje s podatkovno bazo (QUERY LANGUAGES)
- generatorji poročil in tabel (REPORT GENERATORS)
- jeziki za grafiko (GRAPHIC LANGUAGES)
- visokonivojski (nepostopkovni) jeziki (VERY HIGH LEVEL PROGRAMMING LANGUAGES)
- parametrizirani aplikacijski paketi (PARAMETRIZED APPLICATION PACKAGES)

Te pripomočke imenujemo tudi JEZIKI IV. GENERACIJE. Predstavljajo orodje za povečanje produktivnosti razvoja aplikacij s tem, da to delo lahko razen profesionalnih programerjev izvajajo tudi neposredni uporabniki, obenem pa so tudi sredstvo za učinkovit in neposreden dialog s podatkovno bazo, zasotavljanje informacij v realnem času in dajejo s tem dejansko podporo procesom planiranja, upravljanja in odločanja.

Prav tako moramo poudariti učinkovitost sodobnih metod v vseh fazah cikla razvoja aplikacij, zlasti pa v dosedaj delovno in konceptijsko najbolj intenzivnih, to je pri sistemski analizi in načrtovanju sistema. Težave, ki so bile v teh fazah najbolj značilne, predvsem zaradi njihovega trajanja, sodelovanja uporabnikov, priprave ustrezne dokumentacije, vzdrževanja, ob uporabi sodobnih pripomočkov v veliki meri odpadejo ali se

zmanjšajo. Še posebej to velja za področji odpravljanja napak in vzdrževanja, ki po novem načinu dela ne predstavljata težav, saj je sprotno izpopolnjevanje, modificiranje in popravljanje aplikacij vsrajeno v sistem dela, prav tako pa se sistem v znatni meri sam dokumentira in tako odpadejo težave zaradi nezadostne dokumentacije.

V razvoju računalniških aplikacij je zrelo pomemben korak vpeljava prototipov. Na vseh področjih tehnike je ta metoda povsem običajna v inovacijski verigi priprave novih proizvodov. Ugotovimo lahko, da je bil pri konvencionalnem načinu razvoja aplikacij prototip člen, ki je resnično manjkal in so zaradi tega nastajale napake, končni uporabniki pogosto niso bili zadovoljni z razvitim informacijskim sistemom ali pa tak informacijski sistem pogosto sploh ni bil uporaben. Veliko laže je napake in nefunkcionalnosti nekoga sistema namreč usotavljati v njesovi opredeljeni funkciji, potem ko je sistem izdelan in ga je moč preizkušati in testirati. Prav tako je bilo pri konvencionalnem načinu zelo težko usotoviti dejanske želje uporabnikov, ker jih je bilo težko pritesniti k sodelovanju. Za nov način nastajanja informacijskih sistemov pa je značilna spontanost z intuitivnim pristopom uporabnikov. Metoda prototipov omogoča, da končni uporabniki v sodelovanju s sistemskim analitikom, v neposrednem dialogu z računalnikom opredelijo svoje zahteve na primer po obliki vnašanja podatkov, po obliki poročil in odgovorov na zaslону, ki dejansko ustrezajo njihovim zahtevam. Ko končni uporabnik vidi na zaslону obliko poročila, lahko neprimerno laže usotovi, ali mu to poročilo ustreza, lahko zahteva spremembe,

dokler ne doseže zaželene oblike. To pri vnaprejšnjem abstraktnem opredeljevanju teh zahtev ni bilo možno. Sodobni pripomočki omogočajo neposredno izdelavo oblik vhodov in izhodov sistema in s tem torej prototipov ter njihovo sprotno dopolnjevanje in verifikacijo s strani končnih uporabnikov. Pri tem so vse modifikacije vsrajane v način samega dela, tako da razvoja ni potrebno dokončno zamrzniti kot pri konvencionalnem načinu dela, temveč možnosti sprememb obstojajo še naprej, ko je sistem že v normalni uporabi. Prototip se izdeluje postopno, s tem da je prvi osnovni modul, ki zajema najbolj strateške operacije, potrebno izdelati čimhitreje in ga potem v najkrajšem času dograditi do neke, za končnega uporabnika sprejemljive oblike. Ko je prototip dograjen do neke določene faze, lahko optimaliziramo tiste dele programa, ki so kritični glede na zmožljivosti računalnika.

Uporaba sodobnih pripomočkov za razvoj aplikacij, zlasti prototipov, zahteva določene tehnološke pogoje:

- interaktivno uporabo računalnika
- učinkovit sistem za upravljanje s podatkovnimi bazami s sodobnimi pripomočki - jeziki četrte generacije
- baze modelov za delo s podatki, ki vključujejo najrazličnejša matematična, statistična, simulacijska in druga aplikacijska orodja za uporabo podatkov v najrazličnejše specifične namene.

Povečanje produktivnosti, ki ga je moč doseči s sodobnimi metodami, je izredno zlasti pri jezikih za komuniciranje z bazami podatkov, z generatorji poročil, kjer je možno doseči 50 do 100 krat večjo produktivnost kot z ustaljenimi metodami, če pa

upoštevamo celoten cikel razvoja aplikacij, so prihranki oziroma povečanje produktivnosti v razredu 5 do 10 krat.

Specifičnost izgradnje informacijskega sistema je predvsem v tem, da v njej sodelujejo končni uporabniki kot enakopravni svetovalci in načrtovalci le-tega. Pomembno je neposredno delo končnih uporabnikov z računalnikom, da bi na ta način čimhitreje pridobili najvažnejše informacije za pomoč pri planiranju, upravljanju in odločanju.

