

UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

PODATKOVNA SKLADIŠČA  
V MEDNARODNEM PODJETJU

Ljubljana, april 2007

Dušan Pirc

## IZJAVA

Študent Dušan Pirc izjavljam, da sem avtor tega magistrskega dela, ki sem ga napisal pod mentorstvom prof. dr. Andreja Kovačiča in skladno s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega/specialističnega dela na fakultetnih spletnih straneh.

V Ljubljani, dne 4. aprila 2007

Podpis: \_\_\_\_\_

# Kazalo

<b>1.</b>	<b>UVOD.....</b>	<b>1</b>
1.1	PROBLEMATIKA IN NAMEN DELA .....	1
1.2	NAMEN IN CILJ DELA.....	2
1.3	METODE DELA .....	2
<b>2.</b>	<b>ODLOČANJE V POSLOVNEM PROCESU .....</b>	<b>3</b>
2.1	ZAKAJ UPRAVLJAVCI POTREBUJEJO POMOČ INFORMACIJSKE TEHNOLOGIJE PRI ODLOČANJU? .....	6
2.2	KAKO UGOTOVITI INFORMACIJSKE POTREBE UPRAVLJAVCA?.....	6
2.3	ALI JE LAHKO DELO UPRAVLJAVCA POVSEM AVTOMATIZIRANO? .....	7
2.4	TRADICIONALNI SISTEMI ZA PODORO ODLOČANJA .....	9
<b>3.</b>	<b>PODATKOVNA SKLADIŠČA .....</b>	<b>11</b>
3.1	DEFINICIJA PODATKOVNEGA SKLADIŠČA .....	12
3.2	CILJI PODATKOVNEGA SKLADIŠČA.....	15
3.2.1	<i>Kratkoročni cilji .....</i>	<i>16</i>
3.2.2	<i>Dolgoročni cilji .....</i>	<i>17</i>
3.3	ARHITEKTURE PODATKOVNIH SKLADIŠČ .....	18
3.3.1	<i>Centralizirana arhitektura .....</i>	<i>18</i>
3.3.2	<i>Distribuirana arhitektura.....</i>	<i>19</i>
3.3.3	<i>Federativna arhitektura .....</i>	<i>21</i>
3.4	INFORMACIJSKI EKOSISTEM .....	24
3.5	RAZVOJ IN VZDRŽEVANJE PODATKOVNEGA SKLADIŠČA .....	24
3.5.1	<i>Projektno vodenje gradnje podatkovnega skladišča.....</i>	<i>25</i>
3.5.2	<i>Merjenje uspešnosti projekta .....</i>	<i>27</i>
3.5.3	<i>Kritični dejavniki uspeha projekta.....</i>	<i>29</i>
3.5.4	<i>Merjenje rezultatov .....</i>	<i>32</i>
3.5.5	<i>Metodologija razvoja podatkovnega skladišča.....</i>	<i>33</i>
3.5.6	<i>Vzdrževanje podatkovnega skladišča.....</i>	<i>40</i>
<b>4.</b>	<b>PODATKOVNO SKLADIŠČE V PRAKSI.....</b>	<b>43</b>
4.1	PLANIRANJE IN VODENJE PROJEKTA GRADNJE PODATKOVNEGA SKLADIŠČA .....	43
4.1.1	<i>Zgodovina razvoja podjetja in njegovega informacijskega sistema .....</i>	<i>44</i>
4.1.2	<i>Projekt gradnje sistema za podporo odločanju.....</i>	<i>52</i>
4.2	POTEK GRADNJE DIS .....	54
4.2.1	<i>Glavni razvojni koraki.....</i>	<i>55</i>
4.2.2	<i>Uspešnosti prvega koraka projekta DIS.....</i>	<i>58</i>
4.2.3	<i>Nadaljnji projekti razvoja sistema za podporo odločanja .....</i>	<i>59</i>
4.3	VSEBINA PODATKOVNEGA SKLADIŠČA .....	61
4.3.1	<i>Meritve .....</i>	<i>65</i>
4.3.2	<i>Dimenzije.....</i>	<i>68</i>
4.3.3	<i>Posebnosti .....</i>	<i>74</i>
4.4	UČINKI GRADNJE PODATKOVNEGA SKLADIŠČA .....	77
4.4.1	<i>Združeni podatki.....</i>	<i>77</i>
4.4.2	<i>Pridobivanje podatkov .....</i>	<i>78</i>
4.4.3	<i>Uporaba podatkovnega skladišča.....</i>	<i>80</i>
<b>5.</b>	<b>ZAKLJUČEK .....</b>	<b>80</b>

6. SLOVAR KRATIC, POJMOV IN IZRAZOV .....	83
7. LITERATURA .....	84
8. VIRI .....	86

## Kazalo slik

Slika 1: Dvofazni proces odločanja.....	4
Slika 2: Faze odločanja .....	5
Slika 3: Sistemi za podporo upravljanja .....	8
Slika 4: Centralizirana arhitektura .....	19
Slika 5: Distribuirana arhitektura .....	20
Slika 6: Federativna arhitektura .....	21
Slika 7: Vseorganizacijski razvoj.....	26
Slika 8: Kontrolni izpis prenosa faktur .....	47
Slika 9: Zaslonska slika informacijskega sistema PIS – skupna prodaja lastnih artiklov.....	48
Slika 10: PIS - prodaja lastnih artiklov na izbrani državi v kilogramih.....	49
Slika 11: PIS - prodaja v izbrani državi po mesecih v kilogramih. ....	49
Slika 12: Podatki o dnevni prodaji po trgih v kilogramih, SIT in DEM.....	51
Slika 13: Zgradba sistema za podporo odločanja.....	60
Slika 14: Plan prodaje po mesecih v kilogramih (številke so spremenjene!) .....	62
Slika 15: Prejeta naročila izbranega kupca v četrtem četrtletju .....	63
Slika 16: Plan in prodaja po mesecih v kilogramih za izbrani trg .....	65
Slika 17: Časovna dimenzija s prikazom lastnosti elementa 04.04.2002 .....	69
Slika 18: Dimenzija Partner s prikazom lastnosti elementa »ACV Containerdienst GmbH«.	70
Slika 19: Dimenzija Blagovne skupine s prikazom lastnosti elementa "Barva za vlažne zidove 18 L" .....	71
Slika 20: Dimenzija Skupine produktov s prikazom lastnosti elementa "Paleta" .....	72
Slika 21: Dimenzija Izvor s prikazom lastnosti elementa "Videm" .....	73
Slika 22: Dimenzija Prodajalec s prikazom lastnosti za element "Slovenija" .....	73
Slika 23: Dimenzija Računi s prikazom lastnosti za element "IZ2006/00069" .....	74
Slika 24: Prikaz posameznega posla .....	76

## Kazalo tabel

Tabela 1: Finančni plan razvoja DIS.....	54
Tabela 2: Dokumenti, ki nastanejo ob poslu in so v podatkovnem skladišču .....	77
Tabela 3: Čas, potreben za izdelavo novega poročila .....	79

## 1. UVOD

Povečevanje obsega poslovanja, vedno bolj dinamično in kompleksno poslovno okolje ter napredek informacijske tehnologije povečujejo razpoložljivost in množičnost podatkov, ki so na voljo upravljavcu. Velika količina podatkov lahko hitro preseže zmoglosti človeka, da bi te podatke obdelal in na njihovi podlagi izpeljal zaključke (Šinigoj, 2000, str. 469)!

### 1.1 Problematika in namen dela

Kako torej zmanjšati količino podatkov, ki jo mora pregledati upravljavec?

Ena od rešitev, ki so jo podjetja uporabljala v preteklosti, je izdelovanje periodičnih poročil, pogosta so dnevna, tedenska, mesečna, četrletna in letna poročila. Poročila so bila praviloma strogo formalna, višje ravni upravljanja so določile obliko, vsebino in način njihove izdelave. Pripravljavci poročil so poročila pripravljali po vnaprej predpisanih postopkih in jih predajali naročnikom. Kaj se je s poročili dogajalo, jih ni zanimalo in tudi niso imeli povratnih informacij. Navadno niso niti vedeli, zakaj so izdelovali poročila.

Naročniki poročil so naročila dobili v ustaljenih terminih. Lahko se je zgodilo, da so bila posredovana prepozno, ko so bile odločitve že sprejete, ali prezgodaj, ko poročilo še ni bilo aktualno in se je vsebina do takrat, ko bi bila koristna, že pozabila. Naročniki v poročilih velikokrat niso dobili podatkov, ki so jih želeli, naročati so morali dodatna poročila. Ta poročila pa niso nujno zajemala tistih podatkov, ki so jih naročniki v resnici želeli. Zato so naročili nova poročila, ki so se vedno bolj kopičila. Na koncu so bili naročniki zasuti s poročili, v katerih sicer so vsi potrebni podatki, vendar je bila količina prevelika, da bi bila obvladljiva.

S pojavom osebnih računalnikov, predvsem pa s pocenitvijo celotne informacijske infrastrukture, ki je tako postala dostopna za manjše organizacije in večje število uporabnikov, so upravljavci dobili novo možnost za nadzorovanje dogajanja v svoji organizaciji. Najprej so papirna poročila zamenjala poročila v elektronski obliki, vendar so ta imela vse hibe, ki so jih imela papirna. S povečevanjem zmogljivosti informacijskih sistemov so lahko uporabniki sami oblikovali svoja poročila neposredno iz podatkovnih baz za sprotno obdelavo transakcij, krajše transakcijskih ali operativnih sistemov (ang. *On Line Transactional Processing*, krat. *OLTP*). Vendar oblikovanje poročil iz operativnega sistema navadno presega znanje uporabnika informacijskega sistema, je časovno zamudno in lahko povzroči motnje pri samem delovanju operativnega sistema (Inmon, 1993, str. 13).

Povečevanje zmogljivosti informacijskih sistemov je omogočilo razvoj podatkovnih skladišč, v katerih so podatki shranjeni zato, da jih uporabniki lahko analizirajo in niso namenjeni sprotnim potrebam. Najpomembnejša prednost podatkovnih skladišč je ta, da uporabnikom omogočajo samostojno iskanje po podatkovnem skladišču in pridobivanje podatkov, ki jih tisti trenutek potrebujejo za svoje delo. Seveda so na voljo tudi vnaprej pripravljena poročila, ki jih pripravijo pripravljavci poročil za končne uporabnike.

Upravljalci mednarodnih podjetij morajo nadzorovati delovanje mednarodnega podjetja kot celote in tudi njegovih sestavnih delov (Rozman, 1993, str. 106). Pridobivanje podatkov iz zemljepisno oddaljenih podjetij, ki imajo pogosto med seboj slabo združljive informacijske sisteme, je težko. Operativni sistemi posameznih podjetij so prilagojeni zahtevam lokalnega okolja in pridobivanje poročil neposredno iz njih je za uporabnike v matični lokaciji večkrat celo nemogoče. Če pa podatke iz informacijskih sistemov posameznih podjetij združimo v **skupno podatkovno skladišče**, je pridobivanje podatkov iz njega veliko lažje.

Naj povzamem – danes lahko upravljalci pridobijo potrebne podatke za svoje odločitve iz informacijskega sistema z vnaprej pripravljenimi poročili in orodji za izdelavo poročil, ki jih lahko zgradijo sami, ko jih potrebujejo.

### 1.2 Namen in cilj dela

Zaradi prepočasnega poročanja, ki je bilo velikokrat tudi preobširno ali preskopo, so uporabniki zahtevali sistem za pomoč pri upravljanju. Namen naloge je rešiti težave s pravočasnim pridobivanjem tistih podatkov, ki jih upravljalci potrebujejo za sprejemanje odločitev.

Cilj magistrskega dela je gradnja podatkovnega skladišča, ki bo omogočilo uporabnikom hitrejšo pridobivanje podatkov, potrebnih za sprejemanje odločitev. Prek teoretičnega spoznavanja podatkovnih skladišč želim prikazati njihov pomen pri upravljanju.

Dodatno pozornost bom posvetil podatkovnim skladiščem v mednarodnih podjetjih, pri katerih so podatkovna skladišča še posebej pomembna, saj omogočajo uporabnikom, da uporabljajo podatke za podjetje kot celoto in podatke za njegove posamezne, nepovezane dele.

Hipoteza magistrskega dela je, da z uvedbo podatkovnega skladišča skrajšamo čas za pridobivanje podatkov, ki jih uporabniki potrebujejo za sprejemanje odločitev.

Poročila, ki jih uporabniki sami pridobijo iz podatkovnega skladišča, so osredotočena na podatke, ki jih zanimajo, in so zato zanje razumljivejša in preglednejša. Skladišče podatkov zniža stroške za izdelavo poročil.

### 1.3 Metode dela

Izdelava magistrske naloge bo sestavljena iz dveh faz. V prvi fazi sem proučil dostopno literaturo o podatkovnih skladiščih v slovenskem in tujem jeziku. Literature v slovenskem jeziku s tega področja skoraj ni, zato sem se opiral predvsem na tujo literaturo. Teoretična spoznanja so bila osnova za hipotezo naloge in podlaga za razvoj podatkovnega skladišča v mednarodnem podjetju.

Druga faza je praktičen razvoj podatkovnega skladišča. Pri razvoju podatkovnega skladišča sem kot vodja projektne skupine vodil razvoj in gradnjo skladišča. Pri tem mi je bilo v pomoč znanje o projektnem vodenju, pridobljeno med študijem, uporabil pa sem tudi izkušnje iz sodelovanja pri predhodnih projektih razvoja informacijskih storitev. Del projekta razvoja podatkovnega skladišča je tudi pogovor z uporabniki, ki so bili pomemben vir pri potrjevanju hipoteze.

## 2. Odločanje v poslovnem procesu

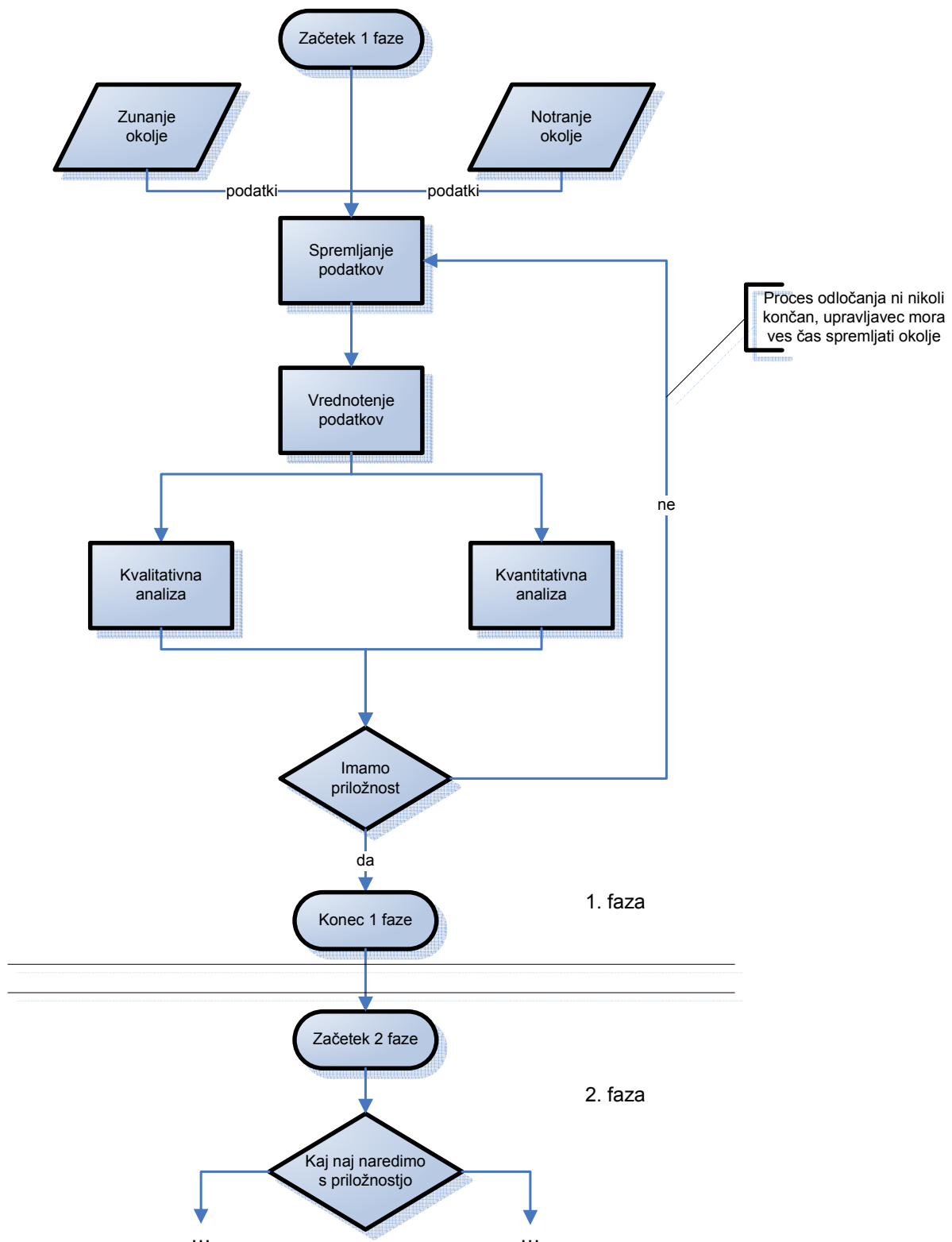
Definicija »odločati se« v Slovarju slovenskega knjižnega jezika je »pri izbiri dati čemu prednost«. Že iz te definicije vidimo, da moramo za odločanje imeti več možnosti (izbir), med katerimi izberemo eno (dati prednost). V vsakodnevnem življenju vsak dan sprejmemo veliko odločitev, navadno povsem podzavestno.

*Upravljanje* (management) je proces, s katerim dosežemo cilje organizacije z uporabo virov. Viri so ljudje, denar, energija, material, prostor, čas ... Vire imamo za vhod v proces, medtem ko je izhod izpolnjevanje zastavljenih ciljev (Turban, 2000, str. 435).

Odločanje upravljavca lahko razdelimo v dve fazi (Turban, 2000, str. 436). V prvi fazi prepozna problem in/ali priložnost. V drugi fazi se odloči, kaj bo storil v zvezi s problemom oziroma priložnostjo. Slika 1 prikazuje proces in tok podatkov v njem. Vidimo, da v proces prihajajo notranji in zunanji podatki. Notranji podatki prihajajo iz same organizacije, zunanji pa iz različnih virov, kot so časopisi, revije, splet in osebni odnosi. Zaradi velike količine podatkov, ki so na voljo, je treba vire skrbno preiskati, da zberemo *primerne* podatke. Zbranim podatkom pripišemo njihovo *vrednost* in jih kvalitativno oziroma kvantitativno analiziramo. Podatke interpretirata (razlagata) upravljavec ali skupina, ki se odločita, ali obstaja problem oziroma priložnost.

Če problem oziroma priložnost obstajata, se preneseta v drugo fazo kot vhod vanjo. V drugi fazi se preiščejo mogoče rešitve, med katerimi upravljavec ali skupina izbereta eno.

Slika 1: Dvofazni proces odločanja



Vir: Turban, 2000, stran 436.

Izvajanje procesa odločanja je brez podpore računalnikov težko. Glavni razlog je informacijska zasičenost, na voljo imamo preveč podatkov, med katerimi težko najdemo

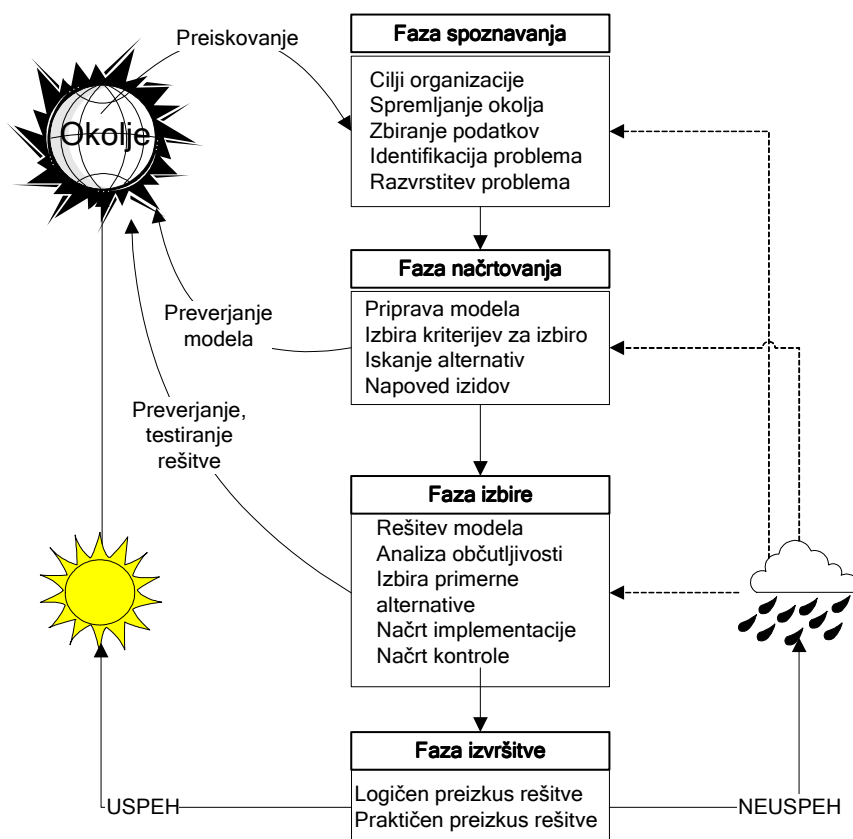


pomembne. Nekateri podatki, ki so nam na voljo, so povsem nepomembni. Okolje in poslovni procesi se ves čas spreminjajo, na voljo imamo veliko alternativ, odločitve morajo biti hitre.

Uspešno upravljanje je odvisno od uspešnega izvajanja upravljaljskih funkcij planiranja, organiziranja, vodenja in kontroliranja (Rozman, 1993, str. 71). Za uspešno izvajanje teh funkcij se morajo upravljalci ves čas *odločati*.

Odločanje je, ne glede na to, ali gre za vsakodnevno odločanje ali za razumske odločitve, sistematičen proces iz štirih glavnih faz (Turban, 2000, str. 438). V *fazi spoznavanja* upravljaavec razišče stanje in definira problem oziroma priložnost. V *fazi načrtovanja* zgradi model, ki poenostavi problem. V modelu upošteva predpostavke, ki poenostavijo resničnost in upošteva razmerja med spremenljivkami. V *fazi izbire* izbere eno od mogočih rešitev in teoretično preveri, ali ustreza pogojem. Ko najde izvedljivo možnost, jo v *fazi izvršitve* izvede v praksi. Če je rešitev uspešna, je problem rešen in nadaljuje z novimi odločitvami. V primeru neuspeha se vrne na eno od predhodnih faz procesa ali celo na začetek procesa (Slika 2).

**Slika 2: Faze odločanja**



Vir: Turban, 2000, stran 438.

## 2.1 Zakaj upravljavci potrebujejo pomoč informacijske tehnologije pri odločanju?

Osnova vsake dobre odločitve so dobri podatki. Podatki so potrebni v vsaki fazi odločitvenega procesa. Sprejemanje odločitev brez pomoči informacijske tehnologije postaja vedno težje zaradi naslednjih trendov (Turban, 2000, str. 441):

- Število možnih rešitev se stalno povečuje zaradi sprememb v tehnologiji, izboljšanih komunikacij, širitve tržišča in elektronskega poslovanja. Več kot imajo upravljavci možnih rešitev, več podatkov marajo preiskati in jih primerjati med seboj.
- Veliko odločitev morajo upravljavci sprejeti v časovni stiski. Pogosto ne morejo ročno obdelati vseh potrebnih podatkov v času, ki jim je na razpolago za odločitev.
- Zaradi nestalnega okolja, v katerem se upravljavci odločajo, potrebujejo za dobre odločitve podrobne analize, ki velikokrat zahtevajo uporabo modelov. Ročno procesiranje modelov zahteva veliko časa in je podvrženo napakam.
- Pogosto potrebuje upravljavec za svoje odločitve oddaljene vire, tako podatke kot ljudi. Vire potrebuje hitro in brez velikih stroškov.

Ti trendi povzročajo težave v procesu odločanja in pri tem lahko pomagajo orodja za podporo odločanja.

## 2.2 Kako ugotoviti informacijske potrebe upravljavca?

Eden ključev uspeha informacijskega sistema je, da zagotovi uporabnikom *prave podatke ob pravem času*. Ugotavljanje informacijskih potreb upravljavcev ni preprosta naloga, je pa zelo pomembna. Ena od možnosti je prototipni pristop. V prvi fazi pridobimo od upravljavca z intervjujem podatke o njegovih informacijskih potrebah. V drugi fazi zgradimo prototip z želenimi podatki. Upravljavec ga ovrednoti in predlaga izboljšave, ki se upoštevajo pri nadgradnji prototipa. Zanka preizkušanja in sprememb se večkrat ponovi, tako da na koncu pridemo do podrobnih zahtev, ki jih vključimo v sistem za podporo.

Informacijske potrebe lahko poiščemo tudi na drugačne načine, ena možnost je pogovor z uporabniki in opazovanje njihovega dela. Sledi nekaj primerov, s katerimi si lahko pomagamo pri ugotavljanju informacijskih potreb (Turban, 2000, str. 442).

1. Upravljavca vprašamo, katera vprašanja bi rad zastavil, ko se vrne s tritedenskega dopusta.
2. Poiščemo glavne cilje v kratkoročnih in dolgoročnih načrtih organizacije in poiščemo informacijske potrebe zanje.
3. Upravljavce, posebej še glavnega upravljavca, vprašamo, za katere podatke še posebej ne želijo, da jih vidi konkurenca.
4. Z opazovanjem ali intervjujem izvemo, katere podatke upravljavci v resnici uporabljajo.
5. Upravljavcem omogočimo neposreden dostop do poročil in jih prosimo, naj preuredijo sistem in poročila na način, ki bo najbolj ustrezal njihovim potrebam. Upravljavcem je navadno veliko lažje povedati, kaj je narobe s predstavljenim sistemom, kakor to, kakšen naj bi bil sistem, ki bi jim zadovoljeval njihove potrebe.

### 2.3 Ali je lahko delo upravljavca povsem avtomatizirano?

Vsakodnevni odločitveni proces zahteva od upravljavca veliko časa. Avtomatizacija nekaterih procesov prihrani čas, poveča doslednost in omogoča boljše odločitve. Bolj avtomatiziramo odločitev, boljše je to za organizacijo. Pri tem se postavi vprašanje, ali lahko povsem avtomatiziramo upravljavčevo delo?

Na splošno lahko najbolj avtomatiziramo delo srednjih upravljavcev (ang. *Mid-level Management*). Njihove odločitve so večinoma rutinske in kot take se lahko avtomatizirajo. Upravljavci na nižji ravni imajo manj odločitvenih nalog, ampak nadzorujejo, izobražujejo in motivirajo sodelavce. Odločitve upravljavcev na najvišjem nivoju so najmanj rutinske in kot take jih je najtežje avtomatizirati.

Seveda, tudi če bi lahko povsem avtomatizirali odločitvene naloge upravljavcev, to ne pomeni, da bi avtomatizirali njihovo delo, ki poleg odločanja vsebuje še mnogo drugega.

Informacijske tehnologije za podporo upravljavcem različni avtorji različno poimenujejo. Turban (Turban, 2000, str. 442) pod nazivom Sistemi za podporo upravljanja (ang. *Management Support Systems*) združuje štiri vrste orodij:

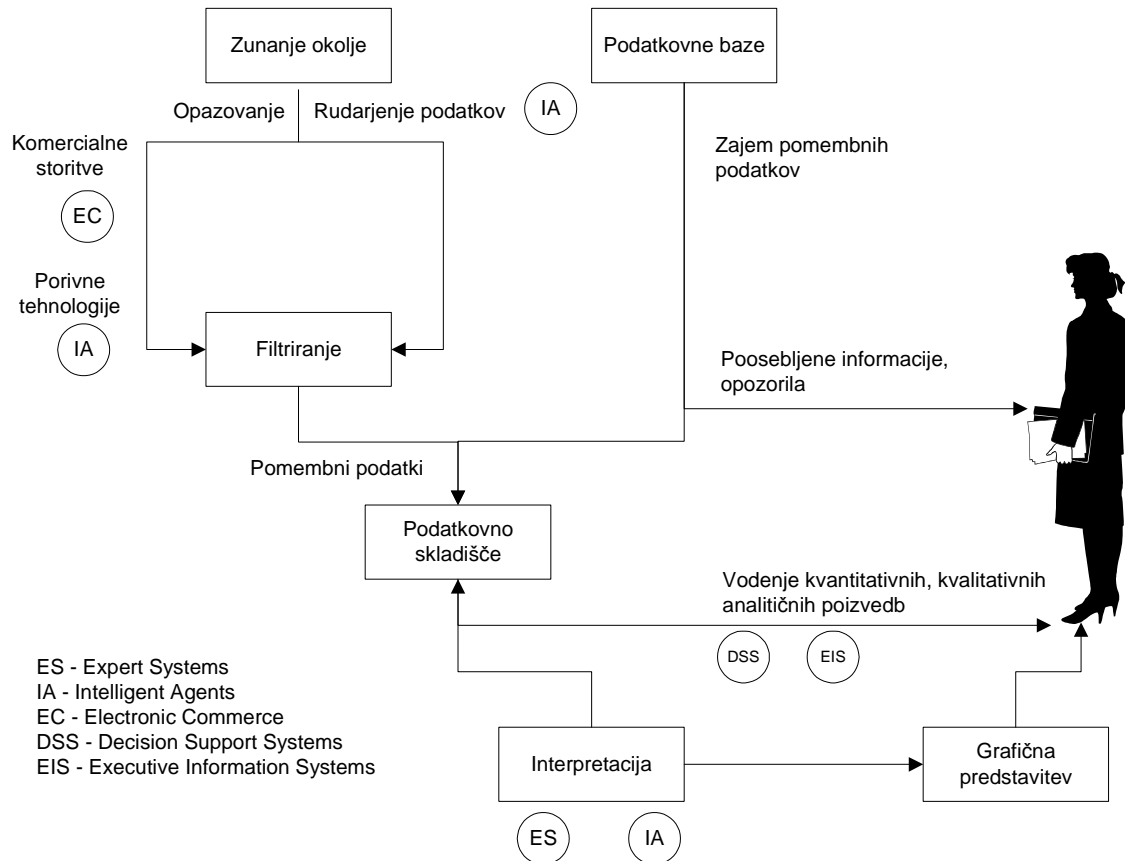
1. Orodja za podporo odločanja (ang. *Decision Support Systems*).
2. Orodja za podporo odločanja vrhnjih upravljavcev (ang. *Enterprise Support Systems*).
3. Orodja za podporo skupinskemu odločanju (ang. *Group Decision Support Systems*).
4. Inteligentni sistemi (ang. *Intelligent Systems*).

Orodja za podporo odločanja, ki so v uporabi že od srede sedemdesetih let dalje, so namenjena analitičnim, kvantitativnim odločitvam. Orodja za podporo odločanja vrhnjih upravljavcev so bila razvita v osemdesetih letih in podpirajo predvsem informacijske potrebe upravljavcev. Tretja skupina orodij pomaga pri odločanju upravljavcev v skupinah. Inteligentni sistemi zajemajo orodja umetne inteligence, ki pomagajo upravljavcem.

Našteta orodja lahko uporabljamo neodvisno eno od drugega, čeprav jih navadno kombiniramo med seboj.

Na Slika 3 vidimo, da morajo upravljavci najti podatke, izločiti nepomembne podatke in pomembne podatke interpretirati. Pri tem so jim v pomoč različni sistemi za podporo odločanja.

**Slika 3: Sistemi za podporo upravljanja**



Vir: Turban, 2000, stran 443.

Vsa orodja za svoje delovanje potrebujejo podatke, ki izvirajo bodisi iz organizacije ali iz okolja. Upravljanje podatkov v organizaciji je zahtevna naloga iz več razlogov (Turban, 2000, str. 478).

- Količina podatkov v organizaciji raste eksponentno. Organizacije hranijo pretekle podatke, ki jim nenehno dodajajo vedno večje količine tekočih podatkov.
- Podatki so razpršeni v organizaciji in posamezniki oziroma jih posamezne službe zbirajo z različnimi metodami.
- Samo majhen del podatkov je pomemben za podporo določeni odločitvi.
- Surovi podatki so shranjeni v različnih računalniških sistemih, podatkovnih bazah, formatih, računalniških in človeških jezikih.
- Podatki so pogosto shranjeni v različnih strežnikih na različnih lokacijah.
- Zakoni v zvezi s podatki se pogosto spreminjajo in so različni med državami.
- Varnost, kakovost in neoporečnost podatkov so zelo pomembni in so lahko hitro ogroženi.

## 2.4 Tradicionalni sistemi za podporo odločanja

Kmalu po uvedbi računalnikov v procese so upravljavci začeli iskati podatke, ki so se hranili v na novo avtomatiziranih procesih. Te zahteve na začetku niso bile prioritete, ker so bili sistemi namenjeni operativnemu delu. Kot sem že omenil, so se sistemi kmalu razdelili na dve veji – na operativne sisteme in na sisteme za podporo odločanju.

Tradicionalni sistemi za podporo odločanja so nastali kot rezultat zahteve upravljavcev po določenih podatkih iz operativnih sistemov. Navadno so zahteve izpolnili z namenskimi aplikacijami, ki so reševale konkretne zahteve uporabnikov. Nastale so in še nastajajo množice programov, ki izdelujejo poročila za omejen krog uporabnikov.

Tradicionalni sistemi za podporo odločanja imajo več pomanjkljivosti. Nekatere pomanjkljivosti so naštetje v nadaljevanju (Adelman, 2000, str. 29).

### **Pomanjkljivosti tradicionalnih sistemov za podporo odločanja**

#### **Oddelčni pogled na podatke**

Najpogostejši način informacijske podpore danes je, da del oddelka informacijske tehnologije podpira enega ali več poslovnih oddelkov. Oddelek znotraj oddelka za informacijsko tehnologijo je lahko formalno definiran, ali pa je to le neformalna skupina znotraj oddelka.

Drugi način informacijske podpore je uporaba lastnih virov za informacijsko podporo. To so lahko zaposleni znotraj enote ali zunanji sodelavci (ang. *outsourcing*).

V obeh načinih samo skupina ljudi podpira določeni oddelek. Medsebojnega sodelovanja med skupinami ni, ali pa je omejeno.

Rešitve, ki tako nastanejo, so povsem zadovoljive za oddelek, za katerega je rešitev nastala. Tudi sistemi za podporo odločanja zadoščajo potrebam naročnika, saj so narejene po njegovih zahtevah. Rešitve so slabo uporabne ali povsem neuporabne za druge oddelke, ki niso sodelovali pri razvoju. Rešitve, ki so nastale v različnih oddelkih, so medsebojno slabo povezljive ali celo povsem nepovezljive.

#### **Nerazumevanje podatkov**

Novi uporabniki tradicionalnih sistemov za podporo odločanju imajo velikokrat težave z razumevanjem podatkov v njih. To ne velja samo za uporabnike, ki so v organizaciji novi, ampak celo za uporabnike, ki so prišli iz drugega oddelka.

Imena za podatke v sistemih so se s časom spreminjala, tako kot sta se spreminjala aplikacija in operativni sistem, iz katerega črpa podatke. Novi uporabniki morajo proučiti dokumentacijo, velikokrat se zgodi, da spremembe sploh niso dokumentirane ... Čas, ki sta ga uporabnik in mentor porabila za prilagajanje, bi lahko porabila veliko koristneje.

### **Različne definicije podatkov**

Ker so sistemi za podporo odločanja nastajali neodvisno drug od drugega, v različnih oddelkih, in ker so jih ustvarjali različni ljudje, so podatki interpretirani s pogledom oddelka in ne celotne organizacije.

Primerjanje podatkov med različnimi oddelki je težko, ker je treba uskladiti pomen posameznih podatkov in pomen vrednosti.

### **Podvojenost podatkov**

Posledica oddelčnega razvoja je tudi, da so isti podatki v različnih podatkovnih bazah po različnih oddelkih (ang. *redundancy*). Ne gre le za to, da ima to za neposredno posledico povečanje stroškov, ki so potrebni za vzdrževanje prevelike količine podatkov (npr. prostor na disku) in vzdrževanje več aplikacij, ki skrbijo za delo s temi podatki. Večji problem je to, da se podatki, ki bi morali biti enaki, razlikujejo. Ko en oddelek odpravi napako v podatkih, tega drugi oddelek ne bo storil, ker o napaki verjetno ni bil niti obveščen. Tako imamo v organizaciji podatke, za katere ne vemo, ali so pravi, oziroma ne vemo, kateri so pravi.

### **Neskladna poročila**

Posledica različnih definicij podatkov in podvojenosti so neskladna poročila (ang. *inconsistent reports*). Poročila iz oddelkov, ki bi morala dati enak rezultat, se medsebojno razlikujejo. Krivda je navadno pripisana oddelku, ki je izdelal poročila, čeprav je lahko razlog v definiciji podatkov in njihovi točnosti.

Takim poročilom kmalu ne zaupa nobena stran – nastajajo nove aplikacije, katerih rezultat pa je spet vprašljiv.

### **Uporabniki ne zaupajo poročilom**

Če ni prave komunikacije med oddelkom informacijske tehnologije in uporabniki, se lahko zgodi, da pride do različnega razumevanja podatkov že znotraj skupine. Ni nujno, da oddelek informacijske tehnologije ne razume poslovnega pogleda na podatke, pogosto tudi uporabniki ne razumejo, kako se podatki zbirajo in obdelujejo v operativnem sistemu.

Pogosto to nerazumevanje pripelje do situacije »mi proti njim«, kar povzroči nemalo težav pri sodelovanju.

Uporabniki poročilom ne zaupajo in jih tudi ne uporabljajo. Potreben je nov sistem, ki pa bo imel iste težave, če ne bo rešen problem komunikacije med uporabniki in razvijalci.

### **Napake v podatkih**

Napake v podatkih, pogosto se uporablja izraz »umazani podatki« (ang. *dirty data*), so pogost problem pri odločanju. Čeprav se vsi zavedajo obstoja napak v podatkih, je izgovorov, ko je treba podatke očistiti, vedno dovolj, od pomanjkanja časa do tega, da to ni pomembno.

Uporabniki, ki vedo, kaj je narobe s podatki, lahko pravilno interpretirajo tudi poročila, ki temeljijo na njih. Uporabnike, ki tega ne vedo, podatki zlahka zavedejo. Potrebujemo čas, da se naučijo, kako izločiti vpliv nepravilnosti v podatkih.

Napake v podatkih so lahko razlog, da so poročila neskladna in jim uporabniki ne zaupajo.

**Podatki se ne delijo**

Deljenje podatkov v tradicionalnih sistemih, ki so operativno usmerjeni, ni pogosto. Uporabniki, ki imajo popoln nadzor nad »svojim« sistemom, v katerem sami definirajo podatke, postopke pridobivanja, čiščenja in vzdrževanja, nimajo želje za deljenjem teh podatkov. Deljenje pomeni sodelovanje drugih uporabnikov in s tem sklepanje kompromisov. Tudi postopki so, če sodeluje več ljudi oziroma oddelkov, daljši. Posameznega oddelka ne zanima, da bo moral drugi oddelk ponovno razviti celoten sistem, ki bo zajemal večino podatkov v že obstoječem sistemu. S stališča organizacije pa je to nepotrebno zapravljanje virov.

Druga razloga za slabo pripravljenost za deljenje podatkov sta strah pred izgubo moči in izpostavljenost. Če tudi drugi oddelki lahko vidijo in uporabljajo »tvoje« podatke, je oddelk izgubil del moči, ker informacije v resnici pomenijo moč. Seveda pa lahko drugi oddelki iz podatkov vidijo tudi morebitne napake v oddelku. Tukaj ne gre za poskuse prevare, ampak preprosto za interpretacijo istih podatkov – kozarec je lahko napol poln ali napol prazen.

Posledica nedeljenja podatkov so seveda podvojenost podatkov, pa tudi različne obrazložitve istih podatkov, neskladna poročila, ki jim uporabniki ne zaupajo, in napake v podatkih.

**Nezdružljivost podatkov**

Tudi, če so oddelki pripravljeni deliti podatke, a ima vsak oddelk svoje samostojne sisteme, so podatki med njimi težko združljivi (ang. *integration*).

Za združevanje podatkov iz različnih sistemov je treba narediti premostitvene aplikacije, za katere je treba zagotoviti dodatne vire – ki pa lahko postanejo nov vir napak.

**Pomanjkanje zgodovinskih podatkov**

S premikom sistemov za podporo odločanja iz operativnega odločanja v taktično in strateško odločanje, se je pojavila potreba za primerjanjem daljših časovnih obdobj. Tradicionalni sistemi za podporo odločanja velikokrat nimajo zgodovinskih podatkov (ang. *historical data*), saj jih oddelki za svoje odločanje ne potrebujejo. Če pa sistem zgodovinske podatke hrani, nima orodij za preprosto izdelavo časovnih primerjav.

Zaradi številnih pomanjkljivosti tradicionalnih sistemov za podporo odločanja se je razvil sistem, ki poskuša zgoraj naštete pomanjkljivosti odpraviti.

**3. Podatkovna skladišča**

V zadnjih letih lahko izraz podatkovno skladišče (ang. *Data Warehouse*, krat. *DW*) slišimo vsakič, ko beseda nanese na sisteme za podporo odločanju. Ker gre za pogosto uporabljen termin, je seveda tudi veliko različnih definicij, kaj pravzaprav podatkovno skladišče je.

### 3.1 Definicija podatkovnega skladišča

Splošna definicija (*Data Warehouse*, 2004) pravi, da je podatkovno skladišče centralno skladišče vseh ali vsaj pomembnega dela podatkov, ki jih zbirajo različni poslovni sistemi v organizaciji. Tipično je podatkovno skladišče nameščeno na osrednjem strežniku podjetja. Podatki iz različnih operativnih sistemov in drugih virov se selektivno zbirajo in organizirajo v podatkovni bazi skladišča podatkov za uporabo v analitičnih aplikacijah in poizvedovanja uporabnikov.

Večina avtorjev se strinja, da je »podatkovno skladišče« izumil William H. Inmon, ki ga opredeljuje (Inmon, 1998, str. 51) kot arhitekturno strukturo, ki podpira upravljanje podatkov in je:

- subjektno usmerjena (ang. *subject oriented*),
- združljiva (ang. *integrated*),
- časovno spremenljiva (ang. *time-variant*),
- obstojna (ang. *non-volatile*),
- vsebuje podrobne in združene podatke.

Upravljavcem podatkovno skladišče pomaga pri odločanju in tako podpira proces strateškega planiranja v organizaciji.

#### **Subjektno usmerjena**

Subjektno usmerjena arhitektura podatkovnega skladišča pomeni, da je skladišče organizirano okoli glavnih entitet organizacije (Inmon, 1998, str. 52), kot so:

- kupci,
- produkti,
- dobavitelji,
- posli,
- naročila,
- dobave,
- računi ...

Podatkovno skladišče torej ni funkcionalno ali aplikacijsko orientirano. To omogoča časovno dolgo uporabo podatkov podatkovnega skladišča brez pomembnega spreminjanja njegove organizacije ali strukture. Ta lastnost podatkovnega skladišča je kritična, če upoštevamo količino zgodovinskih podatkov, ki je shranjena v skladišču.

#### **Združljiva**

Združevanje (integracija) podatkov v podatkovnem skladišču se nanaša na fizično združljivost in povezljivost podatkov v skladišču. Podatki imajo tako ne glede na izvor poenotene ključne, oblike zapisov, skupne povezave, poenoteno poimenovanje, merske enote ... Združevanje podatkov ni mogoče, če bi podatke iz operativnega sistema samo kopirali v podatkovno



skladišče. Namesto tega mora proces, ki polni podatkovno skladišče, surove podatke iz operativnega sistema spremeniti in prilagoditi za uporabo v podatkovnem skladišču.

### **Časovno spremenljiva**

Časovna spremenljivost podatkovnega skladišča pomeni, da je vsak zapis točen v določenem časovnem trenutku. Časovno spremenljivost dosežemo z izdelavo posnetkov stanja okolja, ki ga preslikujemo v podatkovno skladišče. Lahko rečemo, da podatkovno skladišče vsebuje samo veliko posnetkov stanja (ang. *snapshot*) in nič drugega.

Vsak posnetek velja (je točen) samo v določenem trenutku in njegova uporaba pred tem časom ali po njem je (lahko) zavajajoča in nepravilna.

Časovno spremenljivost podatkov dosežemo s časovnim elementom v ključu strukture podatkov. Časovni element so lahko leto, četrletje, mesec, teden, dan ali še kakšna druga časovna enota. Časovni element dodamo v ključ zapisa pri vnosu v podatkovno skladišče.

Dva pogosta načina označevanja časovne spremenljivosti sta *stanje* in *dogodek*.

Zapis, ki uporablja *stanje*, ima oznaki »od« in »do«, ki omejujeta čas, ko je zapis veljaven. Primer je pogodba kupca, ki velja v določenem časovnem obdobju.

Zapis, ki uporablja *dogodek*, ima časovno oznako, ki določi čas, ko so podatki v zapisu veljavni. Primer takega zapisa je naročilo z datumom.

### **Obstojna**

Obstojnost ali nespremenljivost vsebine podatkovnega skladišča pomeni, da se zapisi, ko so zapisani v podatkovnem skladišču, ne spreminjajo več. Če se spremenijo, je to izredni dogodek in ne vsakodnevna praksa.

Spremembe v okolju, ki morajo biti zapisane v podatkovnem skladišču, se vanj prenašajo z omenjenimi posnetki stanja.

Podatkovno skladišče vedno omogoča pregledovanje preteklih stanj. Časovno obdobje, za katero se hranijo podatki, je lahko tudi daljše od deset let.

### **Vsebovanje podrobnih in združenih podatkov**

Podrobni podatki predstavljajo podrobne podatke osnovnih operacij organizacije. To so podatki, ki opisujejo stranke, produkte, dejavnosti v skladišču, prodaji, nabavi in tako naprej.

Pri združenih podatkih ločimo dve vrsti združenih podatkov – *profil* in *javni združeni podatki*.

*Profil* nastane, ko v procesu vnosa surovih podatkov te združimo v en sam združen zapis. Profil uporabljamo, ko nas ne zanimajo vse podrobnosti o dogodku, ali pa bi bila količina podrobnih podatkov tako velika, da podatkovno skladišče tega ne bi zmoglo obvladovati. Primer je izdaja surovin iz skladišča v proizvodnjo. V programu, ki vodi skladišče, se beleži vsaka posamezna izdaja vsake surovine posebej. V podatkovnem skladišču potrebujemo samo podatek o tem, koliko je bilo skupaj izdane surovine na posamezen dan. V fazi polnjenja podatkovnega skladišča tako združimo posamezne izdaje v en sam zapis.

*Javni združeni podatki* nastanejo, ko posamezna organizacijska enota v organizaciji pripravi podatke, ki jih potem uporabljajo v celotni organizaciji. Primer takih podatkov je bilanca stanja, ki jo pripravi finančna služba v določenih intervalih, npr. vsak mesec ali vsako četrletje. V bilanci je samo nekaj podatkov o prihodkih, odhodkih, v njej so tudi finančni kazalci. V podatkovno skladišče se zapiše samo majhen del podatkov, ki jih je finančna služba potrebovala za izračun. Izbrani podatki so na voljo za vse, ki lahko uporabljajo podatkovno skladišče.

### **Področno podatkovno skladišče**

Področno podatkovno skladišče (ang. *Data Mart*, krat. *DM*) je skladišče podatkov, zbranih iz operativnih sistemov in drugih izvorov, ki služi določeni skupini uporabnikov (*Data Warehouse*, 2004). Obsega lahko izveček iz podatkovnega skladišča, ali pa je bolj specializiran. Področno podatkovno skladišče mora izpolnjevati posebne zahteve določene skupine uporabnikov tako po vsebini, prikazovanju in načinu uporabe podatkov. Uporabniki področnega podatkovnega skladišča pričakujejo, da bodo podatke uporabljali na njim domač način.

V praksi se izraza podatkovno skladišče in področno podatkovno skladišče večkrat prekrivata. Večina avtorjev pa se strinja, da pri izdelavi področnega podatkovnega skladišča načrtovalci izhajajo iz analize potreb uporabnikov, pri izdelavi podatkovnih skladišč pa iz analize podatkov, ki že obstajajo, in načina, kako se ti podatki lahko pridobijo za uporabo v podatkovnem skladišču.

Podatkovno skladišče je torej središčna, centralizirana, baza podatkov, ki je sicer lahko fizično razdeljena. Področno podatkovno skladišče pa je baza podatkov, ki lahko izhaja iz podatkovnega skladišča, ni pa to nujno in je prilagojena potrebam omejene skupine uporabnikov.

Na splošno lahko trdimo, da je podatkovno skladišče strateško usmerjeno, področno podatkovno skladišče pa taktično. Strukture podatkovnega skladišča so namenjene neznani uporabi, strukture področnega podatkovnega skladišča pa so načrtovane za znane, specifične namene.

Pogosto avtorji uporabljajo izraz *podatkovna tržnica* za področno podatkovno skladišče, vendar menim, da je slednji izraz primernejši od dobesednega prevoda iz angleškega jezika.

Inmon (Inmon, 1998, str. 70) definira področno podatkovno skladišče kot podmnožico podatkovnega skladišča, ki je prilagojena potrebi uporabnikov določenega oddelka. Dopusča možnost, da se področno podatkovno skladišče uporablja v več oddelkih, ki imajo enake ali podobne zahteve.

Področna podatkovna skladišča med seboj neposredno ne izmenjujejo podatkov. Če se podatki morajo izmenjavati, potem se izmenjujejo prek podatkovnega skladišča. Če sledimo temu načinu izmenjave podatkov, imamo zagotovljeno združljivost podatkov med področnimi podatkovnimi skladišči in podatkovnim skladiščem.

Področna podatkovna skladišča so po gradnji podatkovnega skladišča zelo privlačna za uporabo iz več razlogov. Oddelek, ki uporablja področno podatkovno skladišče, ima lahko popoln nadzor nad podatki in procesiranjem področnega podatkovnega skladišča. Če oddelek uporablja (skupno) podatkovno skladišče, se mora prilagajati vsem uporabnikom skladišča, deliti mora skupne vire in to je včasih lahko zelo omejujoče. Podatke v lastnem področnem podatkovnem skladišču oddelek lahko prilagodi svojim potrebam – lahko spremenijo ključe, združujejo in razdružujejo podatke, dodajajo nove izračunane podatke.

Področno podatkovno skladišče mora biti v primerjavi s podatkovnim skladiščem prilagodljivo in dostopno. Ker vsebuje v primerjavi s podatkovnim skladiščem veliko manj podatkov, je odzivnost področnega podatkovnega skladišča boljša.

V nasprotju z Inmonom pa Kimball (Kimball, 1998. str. 19) definira podatkovno skladišče kot unijo področnih podatkovnih skladišč, pri kateri podatkovno skladišče nastane iz že zgrajenih področnih podatkovnih skladišč.

Manjša podjetja, v slovenskem merilu pa sem lahko štejemo tudi večino naših velikih podjetij, navadno uporabljajo samo eno, skupno podatkovno skladišče in nimajo izdelanih področnih podatkovnih skladišč.

V delu dosledno uporabljam izraz »podatkovno skladišče« za zaključeno, samostojno strukturo. Izraz »področno podatkovno skladišče« pa za podatkovno skladišče, ki je na nek način povezano z drugimi strukturami, oziroma je del neke večje celote. Pri tem je področno podatkovno skladišče lahko tudi samo povsem samostojno delujoče podatkovno skladišče.

### 3.2 Cilji podatkovnega skladišča

Glavni cilj podatkovnega skladišča mora biti odpravljanje težav v organizaciji (Adelman, 2000, str. 37). Ta težava je lahko znižanje dohodka, nezmožnost sledenja konkurenci, visoki stroški proizvodnje, izgubljanje strank, zamujene priložnosti ...

Prva napaka, ki se ji moramo izogniti pri gradnji podatkovnih skladišč, je, da poskusimo rešiti vse na podatke vezane težave naenkrat. Cilje gradnje podatkovnega skladišča razdelimo na dve skupini – na kratkoročne in dolgoročne cilje.

Kratkoročni cilji so uresničljivi s samo z gradnjo podatkovnega skladišča in prinesejo takojšne prednosti uporabnikom. Dolgoročni cilji pa se uresničijo šele z dolgoročno uporabo podatkovnega skladišča.

Poudariti velja, da morajo biti tako kratkoročni kot dolgoročni cilji, ki jih želimo doseči z gradnjo podatkovnega skladišča, v skladu s strateškimi cilji organizacije. Če si za cilj gradnje zadamo proučevanje navad kupcev, da jim bomo lahko prodali dražje izdelke, strateški cilj organizacije pa je zniževanje stroškov, naš projekt ne bo imel podpore vrhnjih upravljavcev in bo obsojen na neuspeh.

### 3.2.1 Kratkoročni cilji

Kratkoročne cilje mora podatkovno skladišče uresničiti takoj, ko je projekt gradnje končan. Kratkoročni cilji morajo odpraviti najbolj pogoste napake, ki veljajo za tradicionalne sisteme za podporo odločanja.

#### **Odprava napak v podatkih**

Podatki, ki jih želimo prenesti v podatkovno skladišče iz operativnih sistemov, vsebujejo napake. Če napak ne bomo odpravili, uporabniki ne bodo zaupali podatkom, ki jih bodo pridobili iz podatkovnega skladišča in ga zato tudi ne bodo uporabljali.

Čiščenje podatkov (ang. *data cleaning*) je časovno zahtevna naloga, za katero potrebujemo tudi veliko drugih virov. Zato je gradnja podatkovnega skladišča daljša in dražja, kar pa je v nasprotju z željo, da se projekt gradnje konča čim prej in s čim manjšimi stroški. Treba je narediti kompromis med željo po podatkih brez napak in stroški, ki so potrebni za gradnjo podatkovnega skladišča (Adelman, 2000, str. 39).

#### **Zmanjšanje neskladnih poročil**

Pogosta slabost tradicionalnih sistemov za podporo odločanja so neskladna poročila in podatkovno skladišče mora taka poročila zmanjšati na najmanjše možno število.

Nekatere neskladnosti v poročilih bomo odpravili že z odpravo napak v podatkih. Izvor neskladnosti pa je tudi napačna raba podatkov, ki izvira iz različnega tolmačenja pomena podatkov. Odpravljanje različnih tolmačenj podatkov zahteva sporazum o pomenu podatkov med vsemi uporabniki in razvijalci.

#### **Združevanje podatkov iz različnih virov**

Podatkov iz različnih operativnih virov navadno ne moremo neposredno združiti med seboj, ker se podatki v različnih sistemih identificirajo na različne načine. Poleg tega so lahko podatki različno podrobni, nanašajo se na različna časovna obdobja in podobno.

V podatkovnem skladišču so podatki iz različnih virov združeni in jih lahko sočasno uporabljamo. Seveda pa je treba pri polnjenju skladišča s podatki iz različnih virov poskrbeti za njihovo pravilno združevanje.

#### **Zajem in objava opisnih podatkov**

Opisni podatki (ang. *Metadata*) so bili pred uvedbo podatkovnih skladišč sestavni del dokumentacije, ki pa jo uporabniki le redko preberejo. V podatkovnih skladiščih so opisni podatki nujni že pri gradnji skladišča, brez njih pa tudi ni mogoča uporaba podatkovnega skladišča.

V podatkovno skladišče združujemo podatke iz različnih virov, s katerimi delajo različni uporabniki, ki imajo za iste podatke vsak svoj naziv. Ti nazivi lahko drugim uporabnikom pomenijo čisto nekaj drugega, razvijalcem pa nekaj tretjega. Zato se je treba že pri gradnji podatkovnega skladišča dogovoriti, kaj pomeni določen podatek oziroma določen opis.

Ko pa so opisni podatki zajeti, jih je treba objaviti tako, da jih uporabniki lahko preprosto uporabljajo.

### **Deljenje podatkov**

Deljenje podatkov med uporabniki iz različnih delov organizacije je navadno cilj gradnje podatkovnega skladišča. Da bi bilo deljenje uspešno, moramo, kot sem že omenil zgoraj, podatke očistiti in se dogovoriti o pomenu podatkov (opisni podatki).

Uporabniki iz različnih delov organizacije se ne razlikujejo le po tem, katere podatke potrebujejo, ampak tudi po tem, do kakšne globine (podrobnosti) jih zanimajo. Zato moramo določiti največjo podrobnost, ki jo bodo vsebovali podatki v podatkovnem skladišču, da bomo zadovoljili vse uporabnike.

### **Spajanje zgodovinskih in trenutnih podatkov**

Podatkovno skladišče vsebuje zgodovinske podatke, da lahko z njihovo pomočjo iščemo trende.

Podatki, ki so nam na voljo v operativnem sistemu, navadno obsegajo le podatke za obdobje zadnjih nekaj let. Zgodovinskih podatkov pri vsakdanjem delu ne potrebujemo, ker lahko dodatno obremenjujejo sistem, in so celo nezaželeni. Poleg tega se zaradi prilagajanja operativnega sistema različnim zahtevam spreminjata oblika in vsebina podatkov.

Pri zgodovinskih podatkih se moramo odločiti, kako dolgo zgodovino podatkov uporabniki potrebujejo. Pri vnosu zgodovinskih podatkov moramo še posebno pozornost nameniti čiščenju podatkov, ker so starejši podatki navadno manj natančni.

## **3.2.2 Dolgoročni cilji**

Dolgoročne cilje lahko s podatkovnim skladiščem uresničimo, če so izpolnjeni kratkoročni cilji in če podatkovno skladišče ves čas nadgrajujemo, tako da sledi potrebam uporabnikov (Adelman, 2000, str. 47).

### **Uskladitev različnih pogledov na iste podatke**

Z uresnitvijo kratkoročnih ciljev zmanjšanja števila neskladnih poročil in deljenje podatkov, smo že pričeli z usklajevanjem različnih pogledov na iste podatke. Predvsem zaradi pomanjkanja časa in pripravljenosti na spremembe je pogosto nemogoče uveljaviti vse uskladitve pri prvi gradnji podatkovnega skladišča.

Z nadaljnjo gradnjo podatkovnega skladišča poskušamo zmanjšati število različnih pogledov na iste podatke.

### **Izdelava celotne slike podatkov v organizaciji**

Pri gradnji podatkovnega skladišča izdelamo tudi logični podatkovni model podatkov, ki jih bomo vključili v podatkovno skladišče. Ko dodajamo podatke v podatkovno skladišče, tudi povečujemo obseg podatkov, ki so zajeti v logičnem modelu. Sčasoma bo logični model obsegal vse podatke v organizaciji – če bomo v podatkovno skladišče vključili vse podatke v organizaciji.

### **Ena vstopna točka do vseh podatkov organizacije**

Z eno vstopno točko želimo doseči, da so vsi podatki v organizaciji zbrani tako, da velja:

- Vsi podatki v podatkovnem skladišču so dosegljivi prek skupnega uporabniškega vmesnika ali vstopne točke.
- Na voljo so standardna orodja za izdelavo poizvedb in poročil, ki so preprosta za uporabo.
- Fizična lokacija podatkov je transparentna za uporabnike.
- Podatki so enotni, brez napak in skladni ali vsaj uskladjivi.
- Rezultati poizvedb so skladni ali vsaj uskladjivi.

Če želimo izpolniti ta cilj, je treba uskladiti tako tehnično arhitekturo informacijskega sistema kot tudi podatkovno arhitekturo. Tehnična arhitektura, ki omogoča izpolnitev ciljev, je trinivojska arhitektura – aplikacijski nivo, orodja za dostop do podatkov (ang. *Data Access Tools*) in podatkovna struktura. Pri podatkovni arhitekturi je treba paziti, da ločimo logični fizični podatkovni model. Ker je možno, da bodo fizične podatkovne baze različne, moramo poskrbeti za mapiranje med njimi.

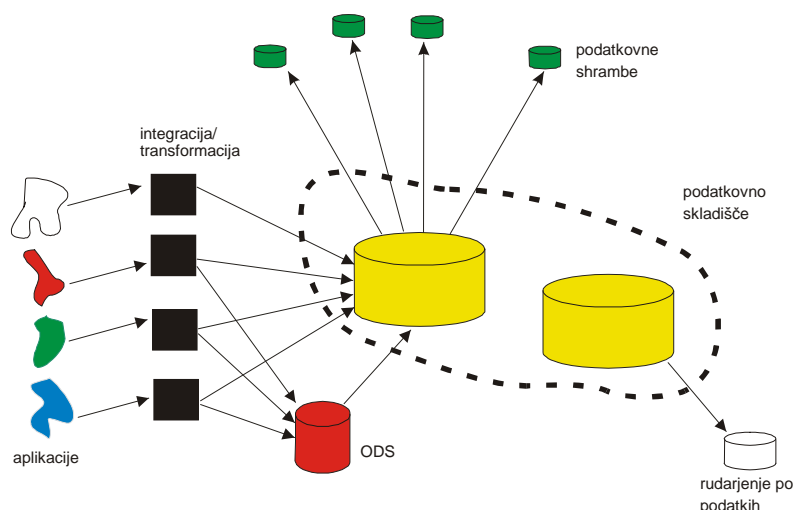
## **3.3 Arhitekture podatkovnih skladišč**

Podatkovno skladiščenje je kljub pogosti in uspešni uporabi še vedno v fazi razvoja, v kateri razvijalci iščejo nove rešitve. Hiter razvoj strojne, programske in komunikacijske opreme omogoča nove rešitve in tudi nove arhitekture podatkovnih skladišč. Izbira arhitekture podatkovnega skladišča je pomembna odločitev načrtovalcev v začetku razvoja podatkovnega skladišča. Izbira arhitekture je zelo pomembna, saj je od nje odvisen podatkovni model, vrstni red razvojnih korakov in vloga področnih podatkovnih skladišč. Različni avtorji praviloma zagovarjajo eno arhitekturo in opozarjajo samo na pomanjkljivosti drugih arhitektur.

Golob predstavlja tri arhitekture podatkovnih skladišč – centralizirano, distribuirano in federativno (Golob, 2001).

### **3.3.1 Centralizirana arhitektura**

Centralizirano arhitekturo v svojih delih kot edino arhitekturo za podatkovna skladišča predstavlja Inmon. V središču centralizirane arhitekture podatkovnega skladišča je podatkovno skladišče celotnega organizacijskega sistema (organizacije), ki se polni iz operativnih podatkovnih baz ter operativne podatkovne shrambe (ang. *Operational Data Storage*, krat. *ODS*). Podatkovno skladišče je tudi vir podatkov področnih podatkovnih skladišč. V taki arhitekturi so področna podatkovna skladišča povsem odvisna struktura, saj se polnijo samo s podatki, ki izvirajo iz centralnega podatkovnega skladišča (Slika 4).

**Slika 4: Centralizirana arhitektura**

Vir: Golob, 2001.

Razvojni cikel podatkovnega skladišča je podatkovno voden, skladišče nastane na podlagi podatkov, ki jih imamo na voljo iz operativnih sistemov, in ne na podlagi zahtev uporabnikov. Zahteve in potrebe uporabnikov mora podatkovno skladišče vedno izpolnjevati, ker pri gradnji v njem zberemo »vse« podatke.

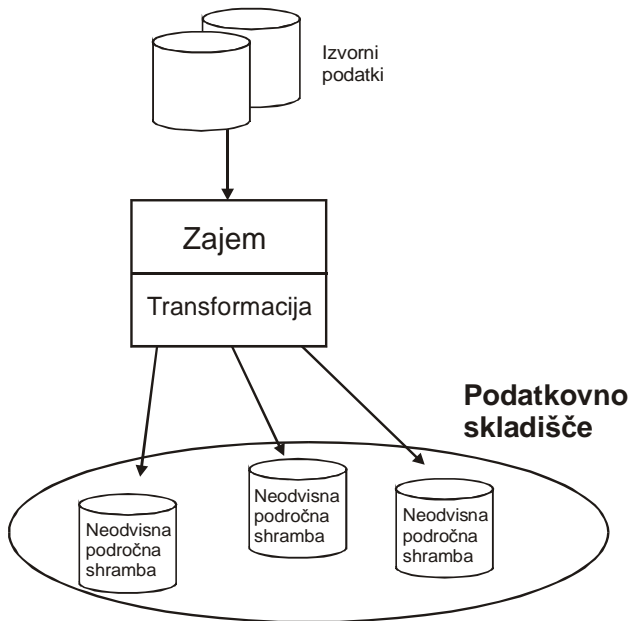
Podatki v centraliziranem podatkovnem skladišču so normalizirani, v redkih izjemah delno denormalizirani. Delno denormaliziranost uvedemo le v primeru, da z njo dosežemo boljšo učinkovitost sistema. Nekaj razlogov za denormalizacijo:

- Redundantni podatki se redno uporabljajo skupaj z drugimi podatki (npr. opis objekta).
- Izračunani podatki se uporabljajo pogosto (npr. zaloga blaga, čeprav je to izpeljan podatek).
- Če vemo, da bodo skupine podatkov pogosto uporabljene skupaj, zgradimo zanje skupni prostor (npr. zbirne mesečne podatke lahko shranimo skupaj in s tem povečamo učinkovitost uporabe).
- Če se bo verjetnost dostopa do posameznih delov podatkov (atributov) bistveno razlikovala, ločimo take podatke na več delov (npr. naslov partnerja in naziv partnerja).

Za tehniko modeliranja uporabljamo entitetno-relacijski model.

### 3.3.2 Distribuirana arhitektura

V nasprotju s centralizirano arhitekturo, pri kateri so področna skladišča podmnžice podatkovnega skladišča, je pri distribuirani arhitekturi podatkovno skladišče unija področnih skladišč (Kimball, 1998. str. 19). Področno skladišče se vede kot specializirano podatkovno skladišče za oddelčne, funkcionalne ali krajevne potrebe (Slika 5).

**Slika 5: Distribuirana arhitektura**

Vir: Golob, 2001.

Organizacija začne razvoj podatkovnega skladišča z gradnjo področnih podatkovnih skladišč, ta pa skupaj poveže v logično podatkovno skladišče celotne organizacije. Za podatkovno skladišče s centralizirano arhitekturo lahko rečemo, da nastaja od zgoraj navzdol, za distribuirano podatkovno skladišče pa, da od spodaj navzgor.

Seveda ne moremo med seboj povezati poljubnih področnih podatkovnih skladišč. Dimenzijski modeli področnih podatkovnih skladišč morajo biti usklajeni med seboj, zato da so usklajeni tudi znotraj skupnega podatkovnega skladišča.

Značilnost *skladne dimenzije* (ang. *Conformed Dimension*) je, da je njen pomen enoličen, ne glede na to, s katero tabelo dejstev je povezana. V skladni dimenziji je vsak podatek predstavljen samo enkrat. Načrtovalci podatkovnega skladišča morajo skladne dimenzije uskladiti pred gradnjo področnih podatkovnih skladišč. Podatke o skladnih dimenzijah morajo objaviti, tako da jih dosledno uporabljajo vsi načrtovalci. Samo na ta način lahko združena področna skladišča delujejo kot (skupno) podatkovno skladišče.

Struktura področnih podatkovnih skladišč je denormalizirana, v določenih primerih delno normalizirana. Osnovni podatkovni model je dimenzijski.

Sama arhitektura omogoča hitro gradnjo prvega področnega skladišča. Prvo skladišče ima lahko velik pozitiven vpliv in ga je relativno lahko zgraditi. Tak pristop omogoča, da dobimo podporo vodstva in uporabnikov. To omogoča nadaljevanje projekta – gradnjo novih področnih podatkovnih skladišč in združitve le-teh v skupno podatkovno skladišče.

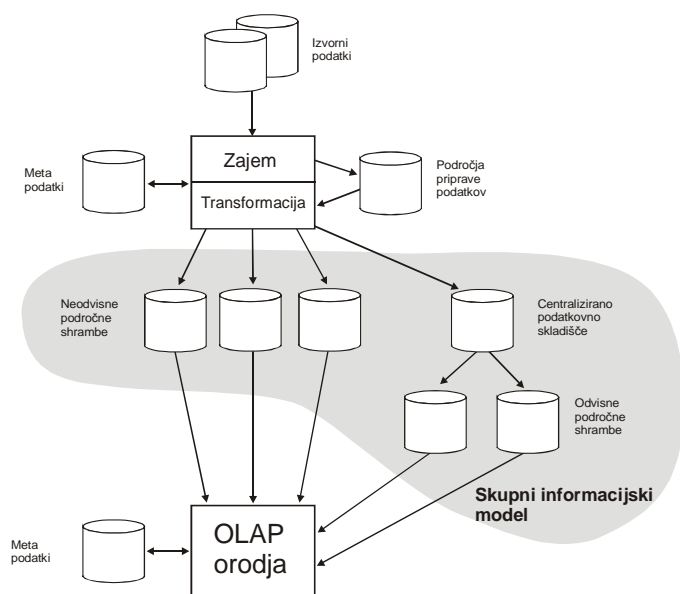


### 3.3.3 Federativna arhitektura

Federativno podatkovno skladišče (ang. *Federated Data Warehouse*) je hibridna rešitev, temelječa na skupnem poslovnem modelu (ang. *Common Business Model*) in področjih priprave podatkov (ang. *Information Staging Areas*), ki so v skupni rabi (Golob, 2001). Federativna arhitektura zagotavlja nizke stroške in hitro povrnitev vloženi sredstev, saj uporablja neodvisna področna skladišča, za katere pa ni potrebna poznejša podatkovna integracija v skupno podatkovno skladišče (White, 2004).

Arhitekturo federativnega podatkovnega skladišča prikazuje Slika 6.

**Slika 6: Federativna arhitektura**



Vir: Golob, 2001

#### Skupni poslovni model

Federativno podatkovno skladišče podpira iterativni razvoj podatkovnega skladišča, ki ga sestavljajo med seboj neodvisna področna podatkovna skladišča. Ključ do združitve podatkov v federativnem podatkovnem skladišču je *skupni poslovni model*, ki se hrani in vzdržuje v podatkovnem skladišču. Izdelava in uporaba skupnega poslovnega modela zagotavlja doslednost v uporabi imen podatkov in poslovnih definicij na vseh področjih, ki jih pokriva projekt podatkovnega skladišča. Z gradnjo novega področnega podatkovnega skladišča moramo skupni poslovni model osvežiti, tako da vedno velja za celotno podatkovno skladišče. Seveda je treba spremembe v skupnem poslovnem modelu prenesti v vsa področna skladišča, ki sestavljajo podatkovno skladišče.

Če zgradbo področnega skladišča narekujejo podatki, ki so na voljo v operativnem sistemu, potem skupni poslovni model osvežujemo vzporedno z razvojem posameznih področnih skladišč.

Če razvoj narekujejo zahteve uporabnikov, potem najprej razvijemo skupni poslovni model in ga nato uporabimo pri razvoju področnih skladišč. Seveda moramo v tem primeru zgraditi vsa področna skladišča na podlagi skupnega poslovnega modela.

Rešitve proizvajalcev omogočajo obe vrsti razvoja. Če razvijamo podatkovno skladišče tako, da z gradnjo novega področnega skladišča spreminjamo skupni poslovni model, je pomembno, da rešitev omogoča spremembe poslovnega modela in združevanje področnih poslovnih modelov v skupni poslovni model. Razvoj dodatno pospešijo predloge poslovnih področij (ang. *Business Area Templates*), ki dokumentirajo poslovno metriko in poslovna pravila. Dokumentacijo potrebujejo uporabniki pri analiziranju in modeliranju poslovnih procesov. Predloge poslovnih procesov pogosto vsebujejo tudi podatkovni model, ki omogoča izdelavo prilagojenih podatkovnih skladišč.

Skupni poslovni model lahko uporabimo tudi pri gradnji orodij za poslovno odločanje in analitičnih aplikacij. Veliko naprednih izdelkov uporablja množico poslovnih pogledov (ang. *Business View*) ali poslovnih meta podatkov, ki olajšajo uporabnikom dostop do podatkov v podatkovnem skladišču. Skupni poslovni model lahko pomaga pri izdelavi tega sloja, saj so v njem dokumentirane potrebe poslovne inteligence (ang. *Business Intelligence*) in načini, kako te potrebe izpolnjujejo podatki iz podatkovnega skladišča.

### **Področje priprave podatkov**

Razvijalci praviloma za vsako novo področno podatkovno skladišče zgradijo nove aplikacije (rutine) za zajem in preoblikovanje podatkov. Te aplikacije so redko združene z aplikacijami za gradnjo drugih, neodvisnih področnih skladišč. S povečevanjem neodvisnih področnih skladišč se povečuje tudi število neodvisnih aplikacij za zajem in preoblikovanje podatkov.

Vzdrževanje aplikacij zahteva mnogo virov, še večji problem pa je koordinacija izvajanja teh aplikacij. Rešitev je razbitje procesiranja na več korakov, kar vključuje tudi razvoj rutin, ki zajemajo podatke iz operativnega sistema in jih shranjujejo v *področje priprave podatkov*. Podrobni podatki se iz tega področja uporabijo za polnjenje neodvisnih področnih podatkovnih skladišč. Orodja za zajem, preoblikovanje in polnjenje (ang. *Extract Transform Load*, krat. *ETL*) preoblikujejo podatke v obliko primerno za področno skladišče.

Pri dodajanju novih področnih skladišč lahko uporabimo obstoječe rutine za polnjenje iz operativnega sistema in podatke, ki so že v področju priprave podatkov. Seveda je včasih treba te rutine nadgraditi in v področje priprave podatkov dodati nove podatke, vendar ostaja večina rutin in področja nespremenjena. Ta način zelo dobro deluje v federativnem podatkovnem skladišču, kjer se za gradnjo rutin za zajem podatkov in področje priprave lahko uporabi skupni poslovni model.

Področje priprave podatkov lahko uporabimo tudi za izboljšanje kakovosti zajetih podatkov. Orodja za zajem, preoblikovanje in polnjenje večkrat ne morejo popraviti napak, ki nastanejo pri izvornih podatkih. Težavo lahko rešimo tako, da podatke prečistimo že pri pripravi podatkov. S prečiščenimi podatki nato polnimo področno podatkovno skladišče.

Uporaba orodij za prečiščevanje in orodij za povratni inženiring (ang. *Data Reengineering Tools*) pred orodji za zajem, preoblikovanje in polnjenje lahko znatno zmanjša težave s kakovostjo podatkov v podatkovnem skladišču, kar pripomore k večjemu uspehu projekta

gradnje podatkovnega skladišča. Tak pristop imenujemo tudi analizno voden (ang. *analysis-led*) in je nasproten ETL-vodenemu (ang. *ETL-led*). Seveda pa tudi analizno voden pristop ne more rešiti vseh težav s kakovostjo podatkov.

V federativno podatkovno skladišče lahko združujemo regionalno področna skladišča in namenska podatkovna skladišča (Jindal, 2004, str. 4).

Pri regionalnem združevanju ima organizacija lokalna področna podatkovna skladišča za lokalne potrebe in globalno področno podatkovno skladišče za potrebe organizacije kot celote. Razlika med sistemi (lokalnimi in globalnim) je v vrsti zajetih podatkov. V globalnem sistemu so podatki v najbolj zgoščeni obliki in referenčni podatki. Referenčni podatki vsebujejo skladne dimenzije (ang. *Conformed Dimension*) in skupne podatke na nivoju organizacije (npr. menjalne tečaje valut). Lokalni sistemi vsebujejo podatke, ki jih potrebujejo za lokalne potrebe, praviloma so to bolj podrobni podatki. Referenčni podatki so osnova za integracijo lokalnih in globalnega podatkovnega skladišča. Referenčni podatki morajo biti skupni med različnimi področnimi podatkovnimi skladišči. Enolična in dosledna definicija referenčnih podatkov med vsebujočimi področnimi podatkovnimi skladišči zagotavlja »eno resnico« v federativni arhitekturi podatkovnega skladišča.

Pri združevanju namenskih področnih podatkovnih skladišč ima organizacija izdelana podatkovna skladišča, ki so namenjena za določene subjekte in so izdelana v zaprtih rešitvah (ang. *Packaged solutions*) ali posebnih aplikacijah. Federativno podatkovno skladišče je tu velik dežnik, ki zagotavlja temelje in okolje za poslovne analize in odločitve v neenotnem okolju.

Federativna podatkovna skladišča so novejša rešitev, ki je proizvajalci rešitev še ne podpirajo množično, čeprav je kar nekaj razlogov za izbiro te arhitekture pri razvoju podatkovnega skladišča (Jindal, 2004, str. 3).

Neorganska rast organizacij je mnoge arhitekture potisnila v nezavidljiv položaj. Združitve in pripojitve organizacij so danes normalen dogodek. Vsak tak dogodek prinese v organizacijo ogromno podatkov iz pridobljenega dela, največji izziv pa je združitev teh podatkov v obstoječi poslovni informacijski sistem. Nima smisla zavreči pridobljenega, povsem delujočega podatkovnega skladišča, zato so informacijski oddelki prisiljeni uporabiti federativno arhitekturo podatkovnega skladišča.

Organizacije dnevno potrebujejo medpodročne analize (ang. *Cross-functional analysis*). Medpodročne analize so poudarjene pri zaključevanju projekta podatkovnega skladišča. Federativno podatkovno skladišče shranjuje podatke o večini, če ne o vseh področjih delovanja organizacije. Ti podatki so lahko shranjeni v enem področnem podatkovnem skladišču ali v več področnih podatkovnih skladiščih. Federativno podatkovno skladišče podpira medpodročne analize z uporabo skupnih dimenzij med različnimi (neodvisnimi) področnimi skladišči. Skupne dimenzije so rezultat uporabe skupnega podatkovnega modela pri gradnji posameznih področnih skladišč.

Gradnja podatkovnega skladišča federativne arhitekture je v primerjavi z gradnjo podatkovnega skladišča centralizirane arhitekture mnogo hitrejša. Eden glavnih razlogov je, da

federativna arhitektura poskuša integrirati obstoječi sistem z uporabo skupnega okvira. Ne zahteva gradnje skupnih temeljev, kar je dolgotrajen proces.

Ko gradimo centralizirano podatkovno skladišče, se pogosto srečamo z nasprotujočimi si interesi, ki želijo prilagoditi podatkovno skladišče svojim zahtevam. Temu se ni moč izogniti in ti interesi vplivajo na implementacijo celotnega podatkovnega skladišča. Federativna arhitektura izhaja iz že obstoječega operativnega informacijskega sistema in poslovnega modela, ki ga lahko združimo v skupni poslovni sistem brez večjih konfliktov.

### 3.4 Informacijski ekosistem

Orodja za podporo odločanju so le eno od informacijskih orodij, ki omogočajo uspešno delovanje organizacije. V organizaciji imamo tako podatkovna skladišča, področna podatkovna skladišča, podatkovno rudarjenje (ang. *Data Mining*), internet in intranet, relacijske baze, orodja za podporo skupinskemu odločanju, inteligentne sisteme ...

Vsako od orodij, tehnik in sistemov nam olajša oziroma izboljša delovanje na določenem področju, težave pa se pojavijo, ko želimo posamezna orodja združiti v celovit sistem.

Potrebujemo okolje, informacijski ekosistem (ang. *Information ecosystem*), ki bo usklajevalo uporabo različnih informacijskih tehnologij ter pospeševalo komunikacijo in medsebojno izmenjavo dela, podatkov, procesa in znanja kot del simbiotske povezave (Inmon, 1998, str. 2).

Informacijski ekosistem je sistem z različnimi komponentami, od katerih vsaka organizacija uporablja direktno, hkrati pa deluje v dogovoru z drugimi komponentami, ki tako skupaj ustvarjajo skupno, uravnoteženo okolje. Podobno kot naravni ekosistem mora biti tudi informacijski ekosistem sposoben prilagajanja spremembam njegovih »prebivalcev«. S časom se zaradi sprememb v informacijskem ekosistemu spreminjajo razmerja med posameznimi komponentami in tudi njihove povezave. Včasih se spremembe pojavijo v na videz povsem nepovezanih delih, včasih so te spremembe celo uničujoče.

### 3.5 Razvoj in vzdrževanje podatkovnega skladišča

Uspešna izvršitev zahtevne naloge zahteva od izvrševalca, da »pred začetkom« dela premisli, kako se bo naloge lotil. Od faze premišljevanja je pogosto najbolj odvisno, s kakšnim rezultatom in kako hitro bo naloga opravljena. Zakaj je zveza »pred začetkom« v narekovajih? Ker premišljevanje seveda že sodi v samo reševanje naloge in ni nekaj, kar se dogaja pred izvrševanjem.

Gradnja podatkovnega skladišča je v vseh pogledih zahtevna naloga in zato je še bolj pomembno, da se je lotimo po temeljitem premisleku. Seveda pa podatkovno skladišče ni namenjeno eni sami osebi, ampak celotni organizaciji, včasih še širše. Zato je treba h gradnji pristopiti projektno.

### 3.5.1 Projektno vodenje gradnje podatkovnega skladišča

Vsi avtorji, ki pišejo o podatkovnih skladiščih, se strinjajo, da je projektni način gradnje podatkovnega skladišča edini mogoči način. Podatkovno skladišče mora biti tak vir podatkov, da generične rešitve, ki ne bi zahtevale prilagoditev končnim uporabnikom, niso na voljo. Podatkovna skladišča so tudi del večjih paketov, ki skrbijo za celotno poslovanje organizacije, vendar jih v fazi uvajanja prilagajajo zahtevam uporabnikov.

Na samem začetku naj povem, da se projekt gradnje podatkovnega skladišča razlikuje od »klasičnih« projektov gradnje programskih paketov za operativno delovanje organizacije po visoki dinamičnosti oziroma spremenljivosti.

Na začetku projekta niso znane želje uporabnikov, ki si navadno nekaj želijo, vendar tega ne znajo prenesti razvijalcem. Po drugi strani tudi razvijalci, če je to njihovo prvo podatkovno skladišče, šele med projektom spoznavajo novo tehnologijo in njene možnosti ter omejitve. Zato so spremembe ciljev projekta neizogibne ves čas njegovega trajanja.

**Tradicionalni razvoj programske opreme** je posledica notranje organiziranosti organizacije. Organizacije imajo združena sorodna opravila v poslovne funkcije. Te so razdeljene na manjše, obvladljive enote – oddelke. Ena ali več enot informacijske tehnologije skrbi za eno poslovno enoto. Sodelovanje je pri klasičnih projektih razvoja programskih paketov preprosto – poslovna in informacijska enota sestavita projektno skupino, ki izvede projekt. Rezultat se uporablja v poslovni enoti, informacijska enota skrbi za delovanje in vzdrževanje paketa. Druge poslovne enote razvijajo s »svojimi« informacijskimi enotami produkte, ki zadovoljujejo njihove potrebe.

Če se lotimo razvoja podatkovnega skladišča na ta način, bomo prišli do, pogojno rečeno, področnih podatkovnih skladišč. Posamezna področna skladišča bodo zadovoljevala potrebe poslovne enote, ki je sodelovala pri nastanku le tega.

Problem nastane, ko želimo ta področna skladišča povezati skupaj v enotno podatkovno skladišče za celotno organizacijo. Preprost primer – prodajna in nabavna enota sta zgradili ločeni podatkovni skladišči. Ko ju želimo združiti, ugotovimo, da se poslovni partnerji pojavljajo dvakrat – kot kupci in kot dobavitelji. Seveda želimo na nivoju organizacije takega partnerja videti le enkrat, želimo vedeti, koliko smo mu prodali in koliko smo kupili od njega. Rešitev takih problemov je v gradnji vmesnikov med njimi. Razvoj, posebej pa vzdrževanje teh vmesnikov, je lahko dražji od razvoja podatkovnega skladišča. Vmesniki pa so lahko tudi nov vir napak.

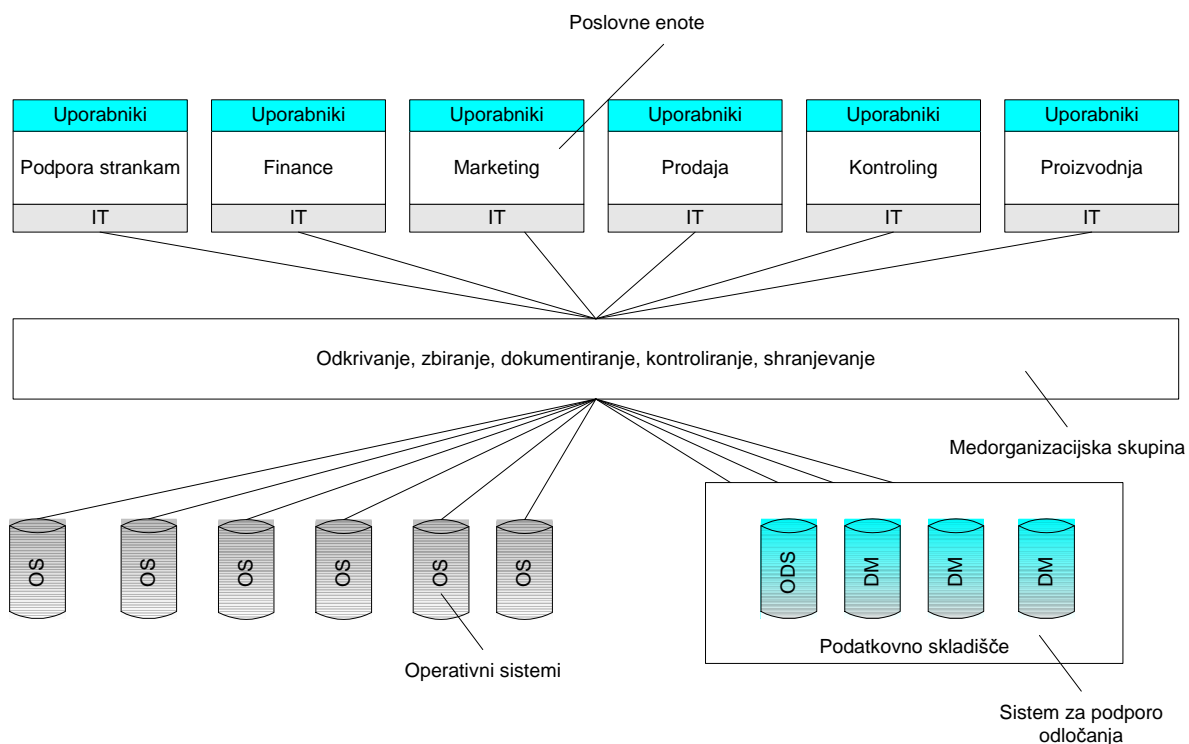
Razvoja podatkovnega skladišča se moramo lotiti na drugačnih temeljih – potrebno je sodelovanje **vseh** poslovnih enot in **vseh** informacijskih enot – **vseorganizacijski razvoj** (ang. *Cross-organizational development*). Vseorganizacijski razvoj ne postavlja pod vprašaj notranje organiziranosti organizacije, ampak samo spreminja način razvoja programske opreme, predvsem pa vodenje projekta.

## Vseorganizacijski razvoj

Vseorganizacijski razvoj seveda ni namenjen samo podatkovnim skladiščem, primeren in priporočljiv je tudi za razvoj operativnih sistemov (Adelman, 2000, str. 6). Z vseorganizacijskim razvojem lahko najdemo temeljne podatke in poslovno inteligenco, ki se skriva v različnih oddelkih organizacije. Podatke, ki so v različnih operativnih sistemih ali v ločenih, težko združljivih področnih podatkovnih skladiščih, moramo zbrati, proučiti, integrirati, razvrstiti in shraniti na lahko dostopno mesto (Slika 7).

Rezultat vseorganizacijskega razvoja ni eno samo podatkovno skladišče, ki zadovoljuje potrebe vseh v organizaciji. Rezultat so podatkovna skladišča, ki so med seboj združljiva. Projekti lahko tečejo zaporedno, v daljšem časovnem obdobju, pomembno je, da so tako nastali sistemi na koncu medsebojno združljivi.

**Slika 7: Vseorganizacijski razvoj**



Vir: Adelman, 2000, stran 6.

Čeprav so prednosti vseorganizacijskega razvoja jasne, je treba vedeti, da ima tak razvoj velik vpliv na velikost projekta – poveča se zapletenost projekta, podaljša čas trajanja, za izvedbo potrebujemo več virov, skratka projekt je zato dražji. Zato je naloga projektne vodje, da skrbno nadzoruje načrtovanje in izvedbo projekta. Treba je postaviti in se držati področij, ki jih projekt pokriva, skrbno je treba nadzorovati spremembe v projektu in si prizadevati za čim boljšo komunikacijo med udeleženci projekta.

### 3.5.2 Merjenje uspešnosti projekta

Projektni način vodenja gradnje podatkovnega skladišča seveda ni zadosten pogoj za to, da bo naloga uspešno zaključena. Uspeh podatkovnega skladišča ni lahko dokazati, ker so cilji navadno bolj kvalitativne kot kvantitativne narave. Je pa lažje opredeliti, kdaj projekt ni bil uspešen, oziroma je propadel (Adelman, 2000, str. 58):

- Izčrpana so bila sredstva, ki so bila namenjena za projekt.
- Uporabniki niso zadovoljni s kakovostjo podatkov in z orodji za delo.
- Podatkovno skladišče uporablja samo del uporabnikov.
- Odzivnost sistem ni zadostna.
- V podatkovno skladišče ni mogoče dodajati novih podatkov.
- Pretvorba podatkov in polnjenje podatkovnega skladišča traja predolgo.

Zaradi različnih razlogov tudi projekte, ki bi morali soditi med neuspešne, razglasijo za uspešne.

Merila za uspeh projekta morata skupaj določiti oddelek za informacijsko tehnologijo in poslovni pokrovitelj projekta. Merila morajo biti taka, da si vsi sodelujoči v projektu prizadevajo za njihovo izpolnitev in ko projekt doseže uresničitev zastavljenih meril, ga lahko razglasimo za uspešnega. Ta merila so lahko sledeča (Adelman, 2000, str. 59).

#### Donosnost naložbe

Donosnost naložbe (ang. *Return Of Investment*, krat. ROI) je količina vrnjenih sredstev iz naložbe. Naložba so v našem primeru sredstvu vložena v projekt gradnje podatkovnega skladišča. Donosnost naložbe se lahko poveča zaradi:

- Zmanjšanja stroškov zaradi boljšega upravljanja z zalogami, zmanjšanja stroškov zaradi neproduktivnega oglaševanja in podobno.
- Povečanja produktivnosti tako v oddelku informacijske tehnologije kot pri uporabnikih podatkovnega skladišča. Analitik lahko porabi do 90 odstotkov časa za pridobivanje podatkov in samo 10 odstotkov za njihovo analizo. Uspešno podatkovno skladišče lahko ti vrednosti obrne. Oddelek informacijske tehnologije bo še vedno pripravljal zapletena poročila in poročila iz operativnih sistemov, vendar bodo ta poročila lahko hitreje pripravljena.
- Povečanje dohodka kot posledica večje prodaje, ker so tržniki na podlagi podatkovnega skladišča pravilno našli ciljne kupce ob pravem času.

Zmanjšanje stroškov, povečanje produktivnosti in prihodka je težko neposredno povezati s podatkovnim skladiščem. Ko se uvaja podatkovno skladišče, organizacija ne miruje – razvija se, nanjo vplivajo tako notranji kot zunanji faktorji, ki vplivajo tako na stroške kot prihodke. Dejanski vpliv podatkovnega skladišča je zato težko, če ne nemogoče, izolirati in prikazati.

### **Podatkovno skladišče se uporablja**

Merilo, ki je lahko za razumevanje in ga je preprosto meriti, je uporaba podatkovnega skladišča. Merimo število uporabnikov in število poizvedb, ki so jih naredili v podatkovnem skladišču. Če uporabniki ves čas uporabljajo podatkovno skladišče in izdelujejo poročila, potem je to dober znak, da smo izdelali uporabno podatkovno skladišče in je projekt uspel.

Če po začetnem spoznavanju podatkovnega skladišča število uporabnikov in njihovih povpraševanj pada, potem lahko sklepamo, da podatkovno skladišče ni v uporabi in je projekt neuspešen.

### **Podatkovno skladišče je uporabno**

Uporabniki podatkovno skladišče uporabljajo, vendar poleg podatkov iz njega uporabljajo pri svojem delu tudi stare vire podatkov, ki naj bi jih podatkovno skladišče nadomestilo.

Uporaba starih virov podatkov ne pomeni nujno, da smo zgradili neuporabno podatkovno skladišče. Mogoče je treba uporabnike samo naučiti, kako lahko pridobijo zelene podatke v podatkovnem skladišču.

Lahko pa smo naredili napako pri gradnji podatkovnega skladišča, ki ne vsebuje vseh podatkov, ki jih uporabniki potrebujejo. Ali je projekt uspešen, je odvisno od tega, ali bomo lahko manjkajoče podatke dodali v obstoječe podatkovno skladišče.

### **Projekt je bil končan v roku**

Projekt, ki je bil dokončan v roku, je uspešen, če so bila izpolnjena tudi druga merila uspešnosti.

Je pa to merilo lahko tudi zavajajoče, če se je časovni načrt določil še pred tem, ko sta bila določena obseg in zahtevnost projekta. Izpolnjevanje časovnega roka je primerno merilo, če v roku upoštevamo dogovorjene spremembe, ki vplivajo na trajanje projekta.

### **Projekt ni presegel proračuna**

Podobno kot za doseganje roka projekta, velja tudi za njegov proračun. Če se projekt med izvajanjem spreminja, potem moramo spremeniti tudi njegov proračun in tak proračun je primerno merilo uspešnosti.

Treba je poudariti, da je proračun projekta težko oceniti. Treba se je dogovoriti, kateri stroški se bodo upoštevali v proračunu projekta (samo programska, strojna oprema in razvoj, izobraževanje uporabnikov, vzdrževanje opreme in podobno). Nekateri stroški se bodo nadaljevali, tudi ko bo projekt formalno že zaključen (vzdrževanje strojne opreme, licence za programsko opremo, stroški polnjenja skladišča z novimi podatki).

### **Povečano zadovoljstvo uporabnikov**

Cilj projekta mora biti zadovoljstvo uporabnikov podatkovnega skladišča. Če so uporabniki zadovoljni z zmožnostmi, odzivnostjo in kakovostjo podatkov v podatkovnem skladišču, lahko zaključimo, da je bil projekt uspešen. Zadovoljstvo uporabnikov lahko merimo z intervjuji.

Pomembno pa je, da so uporabniki seznanjeni z lastnostmi, ki jih bo podatkovno skladišče imelo. Zahteve uporabnikov morajo biti dokumentirane v projektni dokumentaciji.



Nezadovoljstvo uporabnikov, ker podatkovno skladišče ne izpolnjuje njihovih želja, ki jih niso izrazili v času gradnje podatkovnega skladišča, ne more biti razlog, da bi ocenili projekt kot neuspešen.

### **Dodatne zahteve po podatkih in funkcionalnostih**

Projekt je skoraj zanesljivo uspel, če želijo podatkovno skladišče uporabljati tudi uporabniki, ki niso bili vključeni v projekt – npr. iz drugih oddelkov. To pomeni, da se je o uporabnosti podatkovnega skladišča razvedelo in ga uporabniki s pridom izkoriščajo.

Če so obstoječi uporabniki zadovoljni s podatki, ki so jim na voljo v podatkovnem skladišču ter načinom dela z njimi in si želijo dodati nove podatke ali funkcionalnosti, tedaj lahko sklepamo, da je bil projekt uspešen. Podatkovno skladišče uporabniki aktivno uporabljajo in odkrivajo nove možnosti njegove uporabe.

Zahteva po novih podatkih ali funkcionalnostih lahko pomeni tudi, da smo v fazi razvoja spregledali zahteve oziroma želje uporabnikov. V tem primeru ne moremo sklepati, da je bil projekt uspešen.

### **Uresničitev zastavljenih ciljev**

V poglavju 3.2 smo definirali cilje, ki jih mora izpolniti podatkovno skladišče. Nekateri cilji so bili z gradnjo izpolnjeni v celoti, nekateri le deloma. Za merilo uspešnosti se moramo dogovoriti, kako bomo točkovali doseganje ciljev. Skupna ocena bo merilo uspešnosti projekta.

Pri tem mora imeti največjo težo problem, zaradi katerega smo podatkovno skladišče začeli graditi. Če smo ta problem z izdelavo podatkovnega skladišča rešili, potem je projekt uspešen.

### **Podatkovno skladišče je sprožilo spremembe v organizaciji**

Hitre spremembe okolja, v katerem deluje organizacija, zahtevajo hitre in pravilne odgovore. Odločitve so možne samo, če imamo na voljo pravočasno pravilne podatke. Če te odločitve pomenijo tudi spremembe v organizaciji, je bil projekt gradnje podatkovnega skladišča uspešen.

### **3.5.3 Kritični dejavniki uspeha projekta**

Če želimo, da bo gradnja podatkovnega skladišča uspešen projekt, mora biti izpolnjenih nekaj dejavnikov. Odsotnost doseganja kritičnih dejavnikov uspeha navadno obsodi projekt na neuspeh. Če projektni vodja ve, kaj je pomembno za projekt, se lahko pravilno odloči glede proračuna, virov, urnika in vodenja projekta (Adelman, 2000, str. 63).

### **Jasna pričakovanja uporabnikov**

Razvijalci podatkovnega skladišča večkrat ne želijo ali ne upajo povedati uporabnikom, kakšne lastnosti bo imelo podatkovno skladišče. Uporabnikova pričakovanja so lahko povsem drugačna od zmožnosti podatkovnega skladišča.

Zato je treba pričakovanja uporabnikov uskladiti z dejanskimi možnostmi in to dokumentirati v projektni dokumentaciji. Uporabniki morajo v naprej vedeti, kakšne lastnosti lahko pričakujejo od podatkovnega skladišča.

- **Odzivnost** (ang. *performance*). Uporabniki se morajo zavedati, da vse poizvedbe ne bodo izvedene v trenutku. Zapletene poizvedbe, ki bodo združevale več velikih tabel, se bodo lahko izvedle šele po nekaj minutah ali celo urah.
- **Razpoložljivost** (ang. *availability*). Pričakovana dosegljivost podatkovnega skladišča mora biti določena – npr. od 7. do 20. ure med delavniki. Dodan mora biti tudi odstotek planirane razpoložljivosti v tem terminu – npr. 97 %.
- **Funkcije** (ang. *function*). Uporabniki morajo vedeti, katere podatke bo vsebovalo podatkovno skladišče in katera vnaprej definirana poročila jim bodo na voljo. Za podatke morajo vedeti, kako podrobni bodo in kako bodo združeni.
- **Zgodovinski podatki** (ang. *historical data*). Uporabniki morajo vedeti, kako dolgo se bodo hranili zgodovinski podatki in kako podrobni bodo.
- **Točnost podatkov** (ang. *accuracy*). Pri točnosti podatkov moramo poleg natančnosti podatkov doseči tudi, da uporabniki razumejo pomen podatkov.
- **Pravočasnost** (ang. *timeliness*). Pri pravočasnosti moramo določiti, kako hitro po dogodku bomo podatke prenesli v podatkovno skladišče (npr. tri dni po koncu meseca) in kako pogosto bomo podatke v njem osveževali.
- **Urnik** (ang. *schedule*). Uporabniki morajo vedeti, kdaj lahko pričakujejo, da bodo začeli uporabljati podatkovno skladišče.
- **Pomoč** (ang. *support*). Določiti je treba sistem pomoči, ko uporabniki naletijo na problem. Na koga se bodo obrnili v primeru težav, kako bo to osebje poznalo podatke in orodja.

### **Sodelovanje uporabnikov**

Razvoj podatkovnega skladišča brez sodelovanja končnih uporabnikov je obsojen na neuspeh. Več verjetnosti za uspeh ima projekt, pri katerem so se zahteve uporabnikov zbrale na začetku projekta in so se potem skušale uresničiti. Ker pa uporabniki praviloma svojih zahtev ne definirajo dobro, je lahko rezultat nekaj povsem drugega od zahteve.

Če pa dosežemo, da uporabniki sodelujejo v projektu ves čas njegovega izvajanja, je verjetnost za uspeh projekta zelo velika. Uporabniki naj bi sodelovali ne le pri definiranju zahtev in pri preizkušanju, ampak tudi v vseh drugih fazah projekta.

Ravno sodelovanje uporabnikov, njihova pripadnost projektu, je eden od glavnih kazalnikov uspešnosti projekta.

**Dober pokrovitelj projekta**

Priporočljivo je, da je pokrovitelj projekta iz poslovnega dela organizacije. Mora biti dovolj vpliven, da lahko zagotovi potrebna sredstva in druge vire, ki so potrebni za projekt.

Pokrovitelj mora biti predan projektu, težave, ki se bodo pojavile v teku projekta, mu ne smejo rabiti kot povod za končanje projekta ali za odtegnitev podpore.

Najpomembneje je, da ima pokrovitelj veliko potrebo po podatkovnem skladišču. Uspešno dokončan projekt mu bo pomagal rešiti obstoječi problem, ali pa bo njemu oziroma njegovemu oddelku prinesel prednosti.

**Usposobljenost projektne skupine**

Projekt brez usposobljene skupine, ki se posveča projektu, je obsojen na neuspeh. Pri tem je treba poudariti, da ni dovolj, da so člani skupine dobro usposobljeni za razvoj podatkovnega skladišča, te sposobnosti članov morajo tudi biti dodeljene projektu. Pri vseorganizacijskem razvoju se lahko hitro zgodi, da člani, ki so sicer del skupine, ne morejo sodelovati v projektu takrat, ko bi jih potrebovali.

**Stvaren časovni načrt**

Najpogostejši razlog za neuspeh projektov je nestvaren časovni načrt, ki je največkrat vsiljen vodji projekta. Vsiljeni časovni načrti so navadno postavljeni glede na želje vodstva in ne temeljijo na dejanskih možnostih.

**Dober sistem nadzora sprememb v projektu**

V projektih vedno prihaja do sprememb in tudi pri razvoju podatkovnega skladišča prihaja do njih. Projektni vodja mora znati obvladovati spremembe in jih upoštevati pri nadaljnjem delu.

**Izbira pravih orodij**

V projektu se bo uporabljalo več različnih kategorij orodij. Poleg orodij za poizvedbe, s katerimi bodo delali končni uporabniki, so tu še podatkovne baze, orodja za preobrazbo podatkov, modeliranje podatkov, administracijo. Orodja niso poceni, v ceno je treba prišteti še izobraževanje in vzdrževanje.

Pri izbiri orodij je najpomembnejše, če lahko izbiramo vsa orodja, da za njihovo medsebojno delovanje ni treba vzpostavljati posebnih vmesnikov.

**Skupne definicije pomena podatkov**

Vsak oddelek, ki sodeluje v projektu, ima svoje definicije poslovnih izrazov in ti izrazi se velikokrat razlikujejo med oddelki. Izrazi so le redko dokumentirani in vsi mislijo, da so to splošno uveljavljeni izrazi.

Če ni mogoče postaviti skupne definicije izrazov, je izraze treba dokumentirati ter z njimi seznaniti razvijalce in uporabnike.

### **Jasno definirana transformacijska pravila**

Podatki, ki so v podatkovnem skladišču, so v večini primerov preoblikovani iz osnovnih podatkov. Ta preoblikovanja je treba jasno definirati in jih tudi predstaviti uporabnikom.

### **Izobraženi uporabniki**

Vsi uporabniki, ki bodo uporabljali podatkovno skladišče, morajo biti izobraženi za delo. Izobraževanje mora biti prilagojeno nivoju njihovega dela. Poleg spoznavanja samega orodja s katerim bodo delali, morajo uporabniki poznati podatke. Zahtevnejši uporabniki morajo podrobneje poznati zgradbo podatkov.

### **3.5.4 Merjenje rezultatov**

Samo z nadzorovanjem poteka projekta in merjenjem rezultatov lahko ugotovimo, kako uspešen je projekt (Adelman, 2000, str. 66). Merila so tako subjektivna kot objektivna, nekatere meritve imajo lahko negativen vpliv na merjeni sistem (npr. na odzivnost ali stabilnost sistema).

### **Uporabnost**

Podatkovno skladišče mora izpolnjevati zahteve, ki so jih postavili uporabniki. Z orodji, ki so jim na voljo, morajo pridobiti vse podatke, ki jih potrebujejo.

### **Kakovost podatkov**

Podatkovno skladišče brez kakovostnih podatkov je neuporabno. Kakovost lahko preverimo tako, da vprašamo uporabnike, ali so poročila točna. Uporabniki navadno ne vedo odgovora, zato jih prosimo, da poročila občasno ročno preverijo.

### **Odzivnost sistema**

Pri odzivnosti računalniškega sistema lahko merimo čas, potreben za izvedbo poizvedbe, čas, potreben za izdelavo poročila, čas, potreben za nalaganje oziroma osveževanje podatkov in porabo virov v sistemu.

Podatkovna skladišča so na voljo v notranjih omrežjih, zato je treba meriti tudi obremenitev računalniškega omrežja pri izvajanju poizvedb oziroma poročil. Če so uporabniki krajjevno oddaljeni, je treba meriti tudi odzivnost medkrajevne omrežne povezave.

### **Zadovoljstvo uporabnikov**

Po uvedbi podatkovnega skladišča je treba uporabnike anketirati in ankete izvajati periodično. S tem merimo spreminjanje zadovoljstva uporabnikov.

### **Število poizvedb**

Poizvedovalna orodja lahko merijo število poizvedb, ki so jih sprožili oddelki oziroma posamezniki. Število je vedno zanimivo tako za razvijalce kot tudi za upravljalce.

### **Kateri podatki v podatkovnem skladišču se uporabljajo**

S spremljanjem, kateri podatki se uporabljajo, lahko ugotovimo, ali obstajajo v skladišču podatki, ki jih nihče ne uporablja. To je lahko posledica nepopolnega oziroma napačnega zajema zahtev uporabnikov, ki teh podatkov v resnici sploh ne potrebujejo.

Včasih tudi razvijalci napolnijo podatkovno skladišče z vsemi možnimi podatki, da bi se izognili željam uporabnikov po podatkih, ki jih niso zahtevali v razvoju projekta.

Velikokrat uporabniki zahtevajo »vse podatke, za vedno in takoj«, ne da bi se v resnici zavedali, kako lahko s tako zahtevo obremenijo sistem.

Podatki, ki se ne uporabljajo, ne sodijo v podatkovno skladišče. Zaradi takih podatkov je vzdrževanje sistema dražje in zahtevnejše, odzivnost pa slabša.

### **Dosežene prednosti**

Že pred začetkom projekta se ocenijo tako merljive kot nemerljive prednosti, ki jih bo prineslo podatkovno skladišče. Taka prednost je lahko, da se bo novo poročilo izdelalo hitreje in se bo povečala produktivnost.

Merljive prednosti je treba izmeriti, nemerljive pa oceniti. Ne moremo pričakovati, da se bodo prednosti pokazale takoj po uvedbi podatkovnega skladišča, počakati je treba nekaj mesecev in šele nato preveriti, ali smo dosegli zastavljene cilje.

### **3.5.5 Metodologija razvoja podatkovnega skladišča**

Razvoj programske opreme mora biti načrten, metodičen. Podatkovna skladišča seveda niso nobena izjema. Podatkovnega skladišča ne moremo zgraditi vsega naenkrat, potrebne so iteracije, ponovitve razvoja. Podatkovno skladišče bo zraslo v dobro definiran, čvrst in kakovosten sistem po iterativnem razvoju z majhnimi razvojnimi koraki.

Čeprav je koncept iterativnega razvoja lahko razumeti, ga je v praksi večkrat težko uresničiti. Uporabniki so navajeni, da vse zahtevajo že v začetku, ker druge možnosti navadno niso imeli ali je bila časovno preveč oddaljena. Tako zahtevajo **vse podatke z vsemi podrobnostmi** operativnega nivoja, ki morajo biti na voljo **takoj in za vedno**.

Take zahteve je skoraj nemogoče izpolniti. Uporabniki dobro poznajo svoje področje dela, ki je tudi relativno stabilno, zato lahko dobro definirajo potrebe za operativne sisteme, ki jih uporabljajo pri vsakodnevnem delu. V primeru podatkovnih skladišč pa lahko s podatki delajo, kar želijo, in ves čas odkrivajo nove možnosti sistema. Postavljajo nove zahteve po podatkih in spreminjajo že obstoječe podatke. Tako je treba ves čas spreminjati podatkovno skladišče in večje kot je, več časa in virov je potrebnih za spreminjanje (Adelman, 2000, str. 194).

Zahtevi **vse in takoj** je težko izpolniti že v operativnih sistemih, v podatkovnih skladiščih pa je to še težje. V podatkovnem skladišču **vse** pomeni podatke iz različnih operativnih sistemov, podatke, ki niso bili nikoli združeni in njihova kakovost ni bila nikoli preverjena. Nastopajo lahko podatki iz virov zunaj organizacije, s katerimi se prvič srečajo tako poslovni uporabniki kot tudi razvijalci. Čas in trud, ki sta potrebna za združitev in morebitno čiščenje podatkov, zanikata besedo **takoj**.

Nenazadnje je z velikostjo projekta povezana tudi zahtevnost vodenja in tveganost za neuspeh projekta.

### **Prvi projekt**

Prvi projekt oziroma prvi prototip podatkovnega skladišča je najpomembnejši od vseh projektov podatkovnega skladišča in »mora« uspeti, saj se bodo uporabniki, ki ne želijo novega orodja, le na ta način prepričali o njegovi koristnosti.

Prvi projekt je tudi najtežji – v njem sodelujejo uporabniki in strokovnjaki, ki šele spoznavajo posebnosti razvoja podatkovnega skladišča. Izbere in vključi se nova tehnologija, tako strojna kot programska oprema. V organizaciji se bodo, verjetno prvič, vpeljevali novi standardi, skupna poimenovanja podatkov, treba se bo dogovoriti za okrajšane oznake podatkov ...

Da bo projekt uspel, je treba še dosledno upoštevati dejavnike, ki pripomorejo k uspehu (Adelman, 2000, str. 200).

Projekt naj bo majhen, vendar velikost projekta ni merljiva z neko absolutno količino – npr. število gigabajtov podatkov, število poročil ali število uporabnikov. Ali je projekt velik ali majhen, je odvisno od razpoložljivega časa in virov, ki so na voljo za projekt.

Podatki naj se v podatkovno skladišče črpajo samo iz nekaj virov. Na ta način bodo preobrazba, čiščenje in integracija podatkov lažji. Podobno se je treba omejiti na malo podatkovnih elementov. Seveda si prizadevamo, da so podatki, ki jih bomo uporabili, že na izvoru čim bolj brez napak.

Čeprav bo prvi projekt majhen, je pomembno, da bo njegov rezultat prinesel nove možnosti pri uporabi podatkov, ki jih bo zajel. Podatki naj ne bodo kritični za delovanje organizacije, če bo projekt končan pozneje, kot je bilo načrtovano.

Projekt mora imeti izkušenega projektne vodjo in močnega pokrovitelja. Seveda sodelovanje uporabnikov skozi vse faze projekta pripomore k njegovemu uspehu.

Čeprav natančno preverimo vire podatkov in njihovo kakovost, spoznamo in razumemo želje ter zahteve uporabnikov ter imamo dobro in izurjeno skupino razvijalcev, ni najboljše, da razvijemo projekt do končne rešitve. Po izkušnjah uporabniki v prvih mesecih resnične uporabe produkta zahtevajo veliko sprememb v njem. Te spremembe so lahko povsem kozmetične narave, lahko pa zahtevajo veliko sprememb v produktu. Če je produkt že končan

in umeščen v tako imenovano produkcijsko okolje<sup>1</sup>, je te spremembe težko izvesti. Zato je bolje razviti prototip, ki ga uporabniki preizkusijo in tudi uporabljajo pri vsakodnevnom delu. Prototip, ki ga razvijemo, je že pravi sistem, ki ga ni treba spreminjati v dokončen produkt. Ker pa sodi v razvojno okolje, je na njem možno lažje in hitreje uveljavljati spremembe, ki jih zahtevajo uporabniki. Čas za preizkušanje prototipa mora biti določen v projektni dokumentaciji, po njem pa prototip umestimo v produkcijsko okolje. Razvoj prvega podatkovnega skladišča je tako končan.

Pri prototipnem razvoju grozi nevarnost, da bodo stalne zahteve po spremembah pripeljale razvoj v spiralo smrti. Nove in nove zahteve po spremembah, ki jih nekritično uresničujemo v prototipih, so zanesljiv način, da bo projekt neuspešen. Prototip in s tem končna rešitev ne bo nikoli popoln, vedno bodo obstajale želje in zahteve po spremembah. Zato morajo biti vse zahteve in njihove rešitve omejene z viri in časom.

### **Glavni razvojni koraki**

Tradicionalni projekti so razdeljeni na faze (ang. *phase*), kjer se mora ena faza končati in zaključiti, da se lahko začne naslednja. Pri prototipnem razvoju imamo namesto faz glavne razvojne korake, s katerimi merimo potek projekta. Po zaključku glavnega razvojnega koraka imamo nek rezultat, mejnik (ang. *milestone*). V glavnem razvojnem koraku je združenih več tradicionalnih razvojnih faz projekta, ki pa se lahko ponavljajo. Če imamo pri tradicionalnih projektih faze, kot so planiranje, definiranje zahtev, analiza stanja in tako dalje, imamo pri razvoju podatkovnega skladišča korake, ki so usklajeni z rezultati razvoja, kot so grajenje podatkovne baze, izdelava vnaprej izdelanih poizvedb in podobno (Adelman, 2000, str. 208).

### **Vzpostavitveni dokument**

Glavni cilj tega koraka je izdelava »Vzpostavitvenega dokumenta« (ang. *Project Agreement Document*) s podrobnim načrtom projekta podatkovnega skladišča. Za izdelavo podrobnega dogovora je treba skupaj z uporabniki analizirati vhodne datoteke, preveriti razpoložljiva orodja, začeti z gradnjo infrastrukture, določiti vloge in odgovornosti. Vzpostavitveni dokument je skupni izdelek razvijalcev in uporabnikov, za katerega pa se morajo zavedati, da se bo skozi projekt spreminjal. Seveda pa se morajo spremembe uveljavljati z vnaprej dogovorjenimi postopki.

---

<sup>1</sup> Produkcijsko okolje je okolje, v katerem se izvajajo operativni procesi. Nasprotno je razvojno okolje tisto, ki je namenjeno razvoju in testiranju produktov. V produkcijskem okolju lahko vsaka ne dovolj dobro načrtovana sprememba poruši delovanje programov in s tem motnjo v delovanju organizacije.

Vzpostavitevni dokument naj vsebuje:

- **Cilje podatkovnega skladišča** (glej 3.2). Pri tem je posebej pomembno, da so cilji projekta usklajeni s strateškimi cilji organizacije. V dokumentu naj bodo zajeti tako strateški cilji organizacije kot cilji podatkovnega skladišča.
- **Zahteve uporabnikov in njihova rešitev.** Podrobno morajo biti opisane zahteve uporabnikov, ki jih razporedimo bodisi v skupino operativnih poročil bodisi v skupino taktičnih ali strateških analiz. Sem sodi tudi visokonivojski diagram, ki prikazuje predlagano rešitev in potrebne podatkovne baze za njihovo rešitev. Prikazani so tudi možni vhodni podatki in njihov tok v podatkovne baze, ki predstavljajo rešitev težav.
- **Analizo stroškov in upravičenosti.** Potrebe uporabnikov in njihova rešitev morajo biti ovrednotene. Če je mogoče, velikokrat ni, je treba narediti analizo ROI.
- **Grob opis funkcionalnosti.** Ko imamo definirane potrebe in rešitve, je treba definirati obseg projekta ob upoštevanju dejstva, da je projekt, ki bi rešil vse probleme, veliko bolj tvegan, kot majhen projekt, ki je usmerjen k reševanju samo dela problemov. Za točno definiranje funkcionalnosti bo treba izvesti več ponovitev analiz. Koliko funkcionalnosti bomo lahko vključili, je odvisno od več dejavnikov – koliko virov je treba zajeti, koliko podatkov je treba pridobiti iz virov, koliko napak lahko pričakujemo v izvornih podatkih, koliko novih tehnologij bo treba vpeljati, kakšna je usposobljenost razvojne skupine in nenazadnje od velikosti proračuna in časa, ki je na voljo za projekt.
- **Grob logični model podatkov podatkovnega skladišča.** Logični model podatkovnega skladišča nam pomaga pri določanju podatkovnih elementov, ki jih pridobivamo iz izvornih podatkov, iskanju poslovnih pravil, pokaže nam povezave med podatki in njihovo morebitno združevanje. Grob model se v naslednjih korakih s podrobnejšimi analizami dopolnjuje. Lahko ga uporabimo tudi kot osnovo za fizično modeliranje podatkovne baze.
- **Opis infrastrukture.** Uporabniki podatkovnega skladišča vidijo samo njim namenjene funkcionalnosti in podatke. Toda v projekt je treba vključiti tudi infrastrukturo, na kateri deluje podatkovno skladišče. Ločimo razvojno in tehnično infrastrukturo. Razvojna infrastruktura vsebuje nakup ali razvoj metodologije, standardov, nove vloge in odgovornosti osebja ... Tehnična infrastruktura vključuje nove računalniške sisteme, nove ali dodatne sisteme za upravljanje s podatkovnimi bazami, nova orodja za razvoj, nova orodja za iskanje po podatkih ... Ker celotne infrastrukture ne potrebujemo naenkrat, je smiselno vključevati zahtevano infrastrukturo takrat, ko jo zares potrebujemo.
- **Dogovor o zgodovinskih podatkih.** Večina uporabnikov želi zgodovinske podatke že ob samem začetku dela s podatkovnim skladiščem in ne želi, da se naberejo šele sčasoma. Če je treba naložiti tudi zgodovinske podatke, je to dodatna ovira pri razvoju. Velikokrat so stari podatki v operativnem sistemu v drugačni obliki, kot so trenutno, in je zanje treba pripraviti posebna orodja za vključevanje v podatkovno skladišče. Treba je tudi ugotoviti in se dogovoriti o tem, katere zgodovinske podatke uporabniki v resnici potrebujejo in samo ti se vključijo v podatkovno skladišče.



- **Seznam vhodnih virov.** V tej fazi projekta ni mogoče natančno določiti, katere vhodne vire (datoteke) bomo potrebovali pri gradnji skladišča. Uporabniki in člani razvojne skupine imajo navadno dokaj dober občutek, katere vire bomo uporabili, zato seznam teh vključimo v dokument.
- **Oceno kakovosti podatkov.** Po določitvi vhodnih podatkov je priporočljivo preveriti vrednosti podatkov, ki smo jih identificirali z logičnim modelom podatkov. Oceniti je treba napake, ki se pojavljajo v podatkih, poiskati izjeme od poslovnih pravil, polja, ki se uporabljajo za različne namene ... Na podlagi teh podatkov lahko ocenimo delo, ki bo potrebno za čiščenje podatkov.
- **Opis odgovornosti uporabnikov.** Pri razvoju klasičnih informacijskih rešitev je odgovornost končnega uporabnika samo v tem, da definira zahteve in izvede nekaj pogovorov z razvijalci. Nato samo čaka na produkt, ki ga bodo razvijalci razvili. Pri razvoju podatkovnega skladišča to ni dovolj – uporabnik mora aktivno sodelovati ves čas. Ne samo, da mora biti na voljo razvijalcem za pogovore, sodelovati mora tudi pri definiranju standardov, pri medresorskem usklajevanju in mora se dodatno izobraževati, da bo lahko uporabljal novo orodje.
- **Dogovor o razpoložljivosti.** Podobno kot pri operativnem sistemu zahtevajo uporabniki, da je podatkovno skladišče na voljo določen čas. Ker pa je to sistem namenjen za pomoč pri odločanju, navadno ni smiselno zahtevati, da je na razpolago 24 ur na dan, 7 dni v tednu. Izjeme so centralizirana podatkovna skladišča organizacij, ki imajo pisarne v vseh časovnih pasovih in potrebujejo podatke ves čas. Sicer pa mora biti podatkovno skladišče na voljo toliko časa, da se lahko izvedejo planirana opravila in imajo uporabniki dovolj časa za pridobivanje podatkov. Za potrebe izdelave rezervnih kopij in polnjenja podatkov si lahko dovolimo nedelovanje tudi prek vikenda.
- **Varnostne zahteve.** Čeprav je podatkovno skladišče namenjeno pomoči pri odločanju, je predpostavka, da ga ni treba varovati pred nepooblaščenno uporabo, napačna. Varovati je treba sistem kot celoto (kdo ga lahko uporablja), podatkovno bazo in orodja za delo. Posebno pozornost je treba nameniti varovanju detajlnih podatkov. Uporabnik ima lahko dostop do združenih podatkov, nima pa dostopa do osnovnih podatkov. Primer je pregled plač, ko skupno vsoto plač oddelkov uporabniki lahko vidijo, ne smejo pa videti posamezne plače zaposlenega.
- **Opis knjižnice poročil in povpraševanj.** Večina podatkovnih skladišč vsebuje knjižnico vnaprej opredeljenih parametriziranih poročil in povpraševanj, ki so na voljo uporabniku. Opredeliti je treba, kaj bo vključeno v knjižnico in kdo bo odgovoren za njeno gradnjo in vzdrževanje.
- **Dogovor o orodjih za dostop do podatkov.** Orodja za dostop do podatkov je treba izbrati čim bolj na začetku projekta. Treba se je dogovoriti, katere vrste orodij za dostop do podatkov se bo uporabljalo, katera merila se bodo upoštevala pri odločanju med orodji in postopek za izbor. Določiti je treba seznam ljudi, ki bodo sodelovali pri izboru orodij. Če so orodja že izbrana, jih je treba navesti.

- **Seznam vlog v projektni skupini in njihove obveznosti.** Dokument mora vsebovati seznam vlog in njihove obveznosti v projektu. Vloge morajo biti dodeljene članom skupine, takoj ko jih sprejmemo v skupino.
- **Grob načrt projekta.** Napredek projekta bomo lahko pozneje primerjali s tem osnovnim načrtom projekta. V njem določimo tudi postopke za konsolidacijo skupnega načrta z načrti razvojnih skupin.
- **Kaj ni področje projekta.** Pogosto se zgodi, da si vsak od udeležencev sestanka na svoj način razlaga dogovore. Zato je treba v dogovor zapisati tudi stvari, za katere je bilo dogovorjeno, da ne bodo zajete v projektu.
- **Predpostavke in znane omejitve.** Celotni vzpostavitevni dokument temelji na predpostavkah in znanih omejitvah, zato jih je treba dokumentirati. Pričakovati moramo, da se bodo predpostavke in omejitve spremenile.
- **Postopki za uveljavljanje sprememb.** Spremembe v klasičnih projektih razvoja programske opreme niso zaželeno, pri razvoju podatkovnega skladišča pa so skoraj nujnost. Same spremembe je treba imeti pod nadzorom – definirati je treba postopke za umeščanje sprememb v projekt.
- **Ovire in njihovo obvladovanje.** Čeprav smo se na projekt dobro pripravili, vedno lahko nastopijo ovire, ki ovirajo napredek projekta. Podobno kot spremembe morajo biti dokumentirane tudi predvidene ovire. Najpomembnejše pa je, da imamo tudi načrt, s katerim bomo oviro premagali. Seveda pa lahko še vedno nastopijo nepredvidljive ovire.
- **Predhoden dogovor o storitvi** (ang. *Service Level Agreements*). Uporabniki želijo dogovor o storitvi, saj lahko z njim merijo oziroma ocenjujejo orodje, ki so ga dobili. Navadno želijo dogovor o hitrosti izvrševanja poizvedb, kar pa je težko zagotoviti, saj niti ne vemo, kakšne poizvedbe bodo opravljali. Dogovor je treba z razvojem projekta usklajevati z dejanskim stanjem.

### **Določitev tehnološke osnove**

V tem koraku pregledamo in analiziramo predlagane tehnološke osnove, za katere smo se dogovorili v vzpostavitevni dokumentu. Določiti moramo najbolj primerno tehnologijo za podatkovno skladišče, ki lahko vključuje:

- strežnik in njegov operacijski sistem,
- sistem za shranjevanje podatkov,
- sistem za vodenje podatkovnih baz (SUPB),
- orodja za zajem, preoblikovanje in polnjenje podatkov,
- orodja za čiščenje podatkov,
- orodja za poročila in povpraševanje,
- orodja za internet,
- orodja za preizkušanje,
- orodja za računalniško podprto načrtovanje programske opreme,
- pomožna orodja.

Nekatere strežnike in orodja organizacija že lahko ima, nekatere je treba kupiti, nekateri pa včasih niso potrebni. Zahteve iz vzpostavitvenega dokumenta in želena orodja za uresničitev teh zahtev je treba analizirati in primerjati z že obstoječimi orodji v organizaciji in orodji, ki so dostopni na tržišču. Dogovor o merilih in postopku izbire orodja naj bo vključen v vzpostavitveni dokument.

Po izboru je treba orodja namestiti in preizkusiti v okolju, kjer bodo potem uporabljena.

### **Podatkovna baza in zajem/preoblikovanje/polnjenje podatkov**

Sočasno z določanjem tehnološke osnove lahko poteka tudi korak zajema, preoblikovanja in polnjenja. Analizirajo se zahteve uporabnikov iz vzpostavitvenega dokumenta in dokončno se zgradi logični podatkovni model. V tem koraku se konča tudi zbiranje metapodatkov.

Preverijo se podatki v vhodnih datotekah in zgradijo se pravila za njihovo preoblikovanje. Pri tem se lahko uporabljajo orodja za preoblikovanje oziroma se napišejo lastna. Vse izjeme v podatkih se vključijo v dokumentacijo o podatkih.

Ko je logični podatkovni model zaključen in so orodja za zajem, preoblikovanje in polnjenje že definirana, se iz zahtev uporabnikov določi vzorec uporabe podatkov in definirajo podatkovne strukture. Če polnimo operativno podatkovno shrambo, bomo verjetno ostali pri dvodimenzionalnih podatkovnih strukturah, ki bodo zelo podobne tistim iz operativnega sistema. Tudi ko polnimo podatkovno skladišče za vse trenutne in morebitne bodoče uporabnike, ostanemo pri dvodimenzionalni shemi, saj le na ta način lahko zadovoljimo nasprotujoče si zahteve različnih uporabnikov.

Če pa zahtevajo uporabniki predvsem združene podatke, se je bolje odločiti za večdimenzionalno zvezdno shemo. Kljub temu je dvodimenzionalna shema primernejša, če potrebujemo prilagodljiv sistem.

### **Poročila in orodja za poizvedovanje**

Sočasno z zajemom podatkov lahko člani skupine za poročila in orodja za poizvedovanje, oblikujejo poročila, izdelujejo orodja za poizvedovanje in ostala orodja za dostop do podatkov ter metapodatkov. Do tega trenutka mora biti zaključena določitev tehnološke osnove, orodja morajo biti nameščena in preizkušena.

Na podlagi zahtev uporabnikov in pogojev za dostop do podatkov iz koraka gradnje podatkovne baze ter njenega polnjenja se izdelajo specifikacije za dostop do podatkov. Te vključujejo predoglede vnaprej izdelanih poročil, primere pripravljenih poizvedb, primere ad hóc poizvedb, predoglede zaslonov in spletnih strani, če so razvite.

V tem koraku se izdelata tudi množica orodij infrastrukture za podporo podatkovnemu skladišču. To so orodja za dostop do metapodatkov, izdelava priročnikov, izobraževanje prvih uporabnikov, ki delajo s prototipi. Pripravlja se pomoč uporabnikom. Metapodatki so lahko na voljo v obliki standardnih datotek, zaslonov s pomočjo ... Metapodatki morajo vsebovati definicije podatkov, opise postopkov preoblikovanja podatkov, algoritme za združevanje in sumiranje v poizvedbah in poročilih.

Rezultat tega koraka so tudi orodja za nadzor in uglasovanje odzivnosti sistema, s katerimi lahko preprečimo nenadno poslabšanje odzivnega časa.

V tem koraku lahko ugotovimo, da zaradi funkcij za dostop do podatkov, oblike podatkovne baze ali drugih tehničnih razlogov, ni mogoče uresničiti zahtev, ki so zapisane v vzpostavitevnenem dokumentu. Včasih je edina rešitev prilagoditev ali sprememba zahtev. Projektni načrt mora zato vsebovati možnost ponovitve koraka, v katerem smo izdelali vzpostavitevni dokument in vseh naslednjih korakov.

### **Izvedba**

Zadnji razvojni korak je umestitev stabilnega podatkovnega skladišča v produkcijsko okolje, kjer se bodo le še dodajali podatki z vnaprej pripravljenimi paketi. S prenosom iz razvojnega v produkcijsko okolje je priporočljivo počakati nekaj mesecev. V tem času bodo lahko uporabniki opravili tudi poizvedbe, ki jih ne potrebujejo vsak dan. Poleg tega uporabnike zanima, kako se podatki spreminjajo, in v tem obdobju bodo to lahko ugotovili. Če bodo v podatkovnem skladišču že zgodovinski podatki, je to obdobje lahko tudi krajše.

### **Ocenitev projekta**

Po izvedbi podatkovnega skladišča v produkcijskem okolju je treba opraviti še ocenitev projekta. Ker je podatkovno skladišče razvijajoč se sistem, se verjetno že planira novi projekt razvoja, vendar bi bilo nespametno, da ne bi izkušenj iz končanega projekta uporabili v prihodnosti. Na podlagi naučenega bomo lahko spremenili oziroma prilagodili razvojni postopek.

Pravi čas za ocenitev je približno dva meseca po izvedbi, ko so že znani odzivi uporabnikov. Oceniti je treba, kako se je projekt držal terminskega načrta in proračuna. Ali so bile želje uporabnikov izpolnjenе in katere druge vire podatkov še potrebujejo za odločitve? Koliko je bilo napak v podatkih in ali lahko pričakujemo enako stopnjo napak tudi pri drugih izvornih podatkih? So razvijalci pridobili znanja, ki jih bodo lahko znova uporabili pri drugih projektih, kateri svetovalci so sodelovali pri projektu in kako koristni so bili? Kakšna tehnologija je bila uporabljena in kakšno je bilo sodelovanje z dobavitelji?

### **3.5.6 Vzdrževanje podatkovnega skladišča**

Ko je projekt razvoja podatkovnega skladišča zaključen, je treba podatkovno skladišče v uporabi vzdrževati. Očitno je, da je treba podatkovno skladišče polniti s svežimi podatki iz operativnega sistema, vendar vzdrževanje zajema tudi druge dejavnosti.

Sredstva, ki so potrebna za vzdrževanje podatkovnega skladišča, so samo za polovico nižja od sredstev, ki so potrebna za njegov razvoj (Kimball, 1998, str. 716). To velja samo ob predpostavki, da ostaja število uporabnikov nespremenjeno in da se v podatkovno skladišče ne dodajajo podatki iz novih vhodnih virov. Če bomo število uporabnikov povečevali ali dodajali podatke iz novih virov, so stroški vzdrževanja še višji.

Pri stroških vzdrževanja je razlika med podatkovnim skladiščem in operativnimi programi zelo velika. Za vzdrževanje operativnih programov potrebujemo relativno malo virov, za vzdrževanje podatkovnega skladišča pa približno polovico virov, ki so potrebni za razvoj.

Brez dobrega vzdrževanja podatkovno skladišče hitro zastara in ne upravičuje več svojega obstoja.

### **Vzdrževanje delovanja podatkovnega skladišča**

Na začetku dela s podatkovnim skladiščem so uporabniki dopustili, da so bili podatki občasno nedosegljivi in včasih celo napačni. Vendar je postalo podatkovno skladišče orodje, ki ga uporabniki nujno potrebujejo pri svojem delu. Zato želijo in zahtevajo, da so operativni podatki hitro na voljo v podatkovnem skladišču. Skupina, ki skrbi za podatkovno skladišče, mora izpopolnjevati tehnično infrastrukturo, sistem za upravljanje s podatkovnimi bazami in procese, ki upravljajo s podatki in metapodatki. Seveda se mora skupina dodatno izobraževati in spoznavati nova orodja in tehnike (Kimball, 1998, str. 721).

### **Tehnična infrastruktura in sistem za upravljanje s podatkovnimi bazami**

Vzdrževanje tehnične infrastrukture je podobno kakor pri vseh drugih sistemih. Treba je preverjati zasičenost omrežja, zasedenost diska, pomnilnika in procesorja na strežnikih.

Meritve zasedenosti oziroma odzivnosti je treba analizirati in iz trendov ugotoviti, kje je treba sistem nadgraditi, še preden pride do težav.

Tehnična infrastruktura se bo skozi čas zanesljivo spremenila. Pred večjimi spremembami je treba preizkusiti nove dele infrastrukture. To so lahko strojni ali programski deli. Treba je razviti okolje za preizkušanje, v katerem se novosti preverjajo in šele nato se te prenesejo v delovno okolje.

Posebno skrb je treba posvetiti sistemu za upravljanje s podatkovnimi bazami. Spremembe odzivnosti, ki jih lahko povzroči vsakodnevno dodajanje zapisov, je treba ves čas nadzorovati. Usklajevanje delovanje SUPB mora temeljiti na analizi uporabe – kateri podatki in katere agregacije se uporabljajo ... Seveda pa je treba vzdrževati in nadgrajevati tudi strojno opremo, na kateri teče SUPB.

### **Upravljanje podatkov in metapodatkov**

Nenehne spremembe v okolju narekujejo tudi spremembe v operativnih sistemih, ki seveda spremenijo tudi podatke, ki so vhod v podatkovno skladišče. Tudi sam razvoj podatkovnega skladišča ima večkrat za posledico spremembe v operativnem sistemu, ki spremenijo izvorne podatke. Te spremembe je treba nadzirati in ustrezno prilagajati orodja za zajem podatkov. Navadno je problem prilagajanja orodij manjši od problema neobveščeniosti o spremembah! Skupina za podatkovno skladišče mora biti obveščena o spremembah v operativnem sistemu, ravno tako pa se morajo drugi oddelki informacijske tehnologije zavedati posledic, ki jih povzročijo s spremembami v podatkih.

Razvoj podatkovnega skladišča prisili organizacijo k temu, da določi lastnike podatkov. Ker je razvoj vedno časovno omejen in ker je določanje lastnikov težka naloga, je pravi čas za prenos lastništva čas, ko je podatkovno skladišče že v uporabi. Lastnik podatkov mora biti odgovoren za točnost in združljivost podatkov v izvornem sistemu, ker bodo le na ta način podatki v podatkovnem skladišču pravilni.

## **Delo z uporabniki**

Podatkovnega skladišča, ki se ne uporablja, organizacija ne potrebuje. Zato se je treba pri vzdrževanju podatkovnega skladišča usmeriti na njegove uporabnike (Kimball, 1998, str. 718). Uporabnikom je treba pomagati in jih izobraževati.

Pri delu z uporabniki ne smemo pozabiti, da bodo nezadovoljni uporabniki hitro razširili govorice o neuporabnosti podatkovnega skladišča. Zato je treba predvsem na začetku dela posvetiti veliko pozornost podpori uporabnikom. Ni dovolj čakati na zahteve uporabnikov, ki so večkrat negotovi, ker ne znajo uporabljati novih orodij. Osebna pomoč uporabniku je večkrat boljša kot skupinsko delo, ker se v skupini uporabnik navadno ne želi izpostavljati s svojim neznanjem.

Na voljo mora biti služba oziroma oseba, ki bo na voljo za pomoč uporabnikom. Pomoč je treba decentralizirati s pomočjo analitikov, ki dobro poznajo svoje delo. Ti bodo sporočali, kako sistem deluje v praksi, ker bodo z njim reševali svoje vsakodnevne naloge in probleme uporabnikov.

Naprednim uporabnikom je treba omogočiti nadaljevalno izobraževanje. Če to ne bo organizirano, bodo napredni uporabniki neprestano potrebovali pomoč in nasvete, zato je bolje organizirati formalno izobraževanje.

Nekateri uporabniki bodo potrebovali tečaj za osvežitev znanja, ker so določene stvari iz osnovnega izobraževanja pozabili.

Za nove uporabnike podatkovnega skladišča je izobraževanje nujno pred začetkom dela s podatkovnim skladiščem. Tu ne gre toliko za obvladovanje orodij, ampak za pomen podatkov, ki so v podatkovnem skladišču.

## **Ocenjevanje uspešnosti podatkovnega skladišča**

Po ocenitvi uspešnosti samega projekta gradnje podatkovnega skladišča nekaj mesecev po njegovi vzpostavitvi merjenje uspešnosti ni končano. Ves čas je treba preverjati, ali podatkovno skladišče izpolnjuje zahteve, ki so bile določene v vzpostavitvenemu dokumentu. Izpolnjevanje nekaterih zahtev lahko merimo samodejno (npr. odzivnost sistema), za druge moramo povprašati uporabnike. Poročila, ki jih sestavijo uporabniki, navadno ne povedo vsega, zato je treba občasno opraviti osebne intervjuje. Izpolnjevanje zahtev iz dogovora je nujno, da lahko trdimo, da podatkovno skladišče uspešno deluje.

Poleg zahtev iz dogovora, ki jih merimo z vnaprej določenimi postopki, je lahko podatkovno skladišče zelo koristno, a tega ne moremo izmeriti. Zaradi uporabe podatkovnega skladišča so bile sprejete odločitve, ki so koristile organizaciji. Težko je najti primere, v katerih bi bila odločitev sprejeta samo na podlagi podatkov iz podatkovnega skladišča. Je pa veliko odločitev, ki so se sprejele tudi zaradi podatkov iz podatkovnega skladišča. Beleženje teh odločitev naj bo tudi del sistema, s katerim preverjamo uspešnost podatkovnega skladišča (Kimball, 1998, str. 723).

## Rast in razvoj

Uspešna podatkovna skladišča se razvijajo, ker uporabniki zahtevajo nove podatke in aplikacije za delo z njimi. Dober glas o podatkovnem skladišču se bo razširil in nastali bodo novi uporabniki, ki želijo dostop do podatkov in aplikacij.

Skupina za podatkovno skladišče naj ne bi bila razsodnik, čigar želje se bodo uresničile prve. Priporočljivo je ustanoviti vseorganizacijski odbor, ki bo odločal o nadaljnjem razvoju. Poleg postavljanja prioritet mora skrbeti tudi za vire, ki bodo omogočili ta razvoj. Odbor mora imeti dovolj veliko moč, da se bodo njegove odločitve upoštevale pri dodeljevanju sredstev.

Manjše izboljšave se lahko izvedejo brez posredovanja odbora. Seveda pa se je treba najprej dogovoriti, kaj sploh so manjše izboljšave.

Pri zahtevah za večje spremembe, kakršna je na primer dodajanje novega vira podatkov, pa mora odbor odločiti, kaj se bo zgodilo z zahtevo. Lahko jo zavrne, če pa jo sprejme, jo mora umestiti na seznam nalog skupine, ki skrbi za podatkovno skladišče.

Včasih je težko postaviti mejo med tem, ali gre le za evolucijo obstoječega podatkovnega skladišča ali že za nov projekt. Navadno se nadgradnje in novi projekti medsebojno prepletajo. Naj povzamem – uspešno podatkovno skladišče je nikoli dokončana zgodba. Zahteve uporabnikov na eni strani in spremembe v okolju bodo ves čas postavljale nove naloge pred razvijalce podatkovnega skladišča. To pa pomeni tudi, da je bil projekt uspešen!

## 4. Podatkovno skladišče v praksi

Podatkovno skladišče, ki ga predstavljam v nalogi, je nastalo kot rezultat projekta, ki se je začel leta 2000. Osnovni, prvi projekt je trajal dobro leto in njegov rezultat je bilo podatkovno skladišče, ki se je z majhnimi spremembami uporabljalo do leta 2004. Takrat je bilo podatkovno skladišče obširneje spremenjeno z novim projektom.

### 4.1 Planiranje in vodenje projekta gradnje podatkovnega skladišča

Vodenje projekta gradnje podatkovnega skladišča je zahtevna naloga. Največja težava je v tem, da na začetku projekta uporabniki navadno ne želijo tistega, kar si želijo, ko se projekt bliža koncu.

Končni prihodnji uporabniki podatkovnega skladišča imajo za seboj že leta odločitev, ki so temeljila na nekako pridobljenih poročilih. Večina jih težko spremeni svoj način dela, še posebej, če so bili pri delu uspešni. Šele ko vidijo delujoč sistem s pravimi podatki, spoznajo, kako jim lahko to pomaga pri vsakodnevnem delu.

Za razumevanje posebnosti podatkovnega skladišča, ki ga predstavljam, je pomembna tudi zgodovina razvoja družbe, posebej pa še informacijskega sistema v družbi.

#### 4.1.1 Zgodovina razvoja podjetja in njegovega informacijskega sistema

Podjetje, ki ga proučujem, ima dolgo tradicijo, saj je od začetka njegovega delovanja minilo že prek 130 let. Kot leto nastanka podjetja je izbrano leto 1875, ko je bil na območju današnje tovarne postavljen prvi obrat za pridelovanje barve. Kratica JUB se je začela uporabljati leta 1919. Prvi večji tržni uspehi podjetja segajo v sedemdeseta leta prejšnjega stoletja, prodaja pa je hitro naraščala vse do razpada Jugoslavije leta 1991.

Zaradi razpada Jugoslavije je bil padec prodaje leta 1992 kar 55 %. To je bilo eno najbolj kriznih let v zgodovini podjetja. Uspešno izvedena privatizacija in dobre poslovne poteze so do leta 1996 pripeljale do tega, da je prodaja količinsko že preseгла prodajo iz najboljših let pred razpadom Jugoslavije (Batagelj, 2001, str. 30). Količinska prodaja se vidi na Graf 1.

**Graf 1: Količinska prodaja v tonah od leta 1976 dalje**



Vir: Interna dokumentacija podjetja.

Podjetje se je po razpadu Jugoslavije začelo širiti na tuje trge. Širilo se je v dveh smereh – na stare trge bivše Jugoslavije in na nove trge Vzhodne Evrope. Tako so nastala hčerinska podjetja:

- na Hrvaškem leta 1992,
- na Madžarskem leta 1992,
- na Češkem leta 1991,
- v Bosni in Hercegovini leta 1998,
- v Srbiji in Črni Gori leta 2001, od leta 2004 je tam tudi proizvodni obrat,
- v Nemčiji leta 2001,
- v Rusiji leta 2001,
- v Makedoniji leta 2002.



Po razpadu Češkoslovaške republike je JUB imel svojo odvisno družbo na češkem delu države. Da bi obdržal svoj tržni delež, je sklenil dolgoročno sodelovanje s podjetjem JUB Kastaco s.r.o. iz Bratislave, ki je ekskluzivni zastopnik na Slovaškem in prodaja samo izdelke JUB.

Samodejna obdelava podatkov se je v podjetju začela v osemdesetih letih preteklega stoletja. Ročno izpisane dobavnice za izdano blago so posredovali podjetniku, ki je imel računalnik IBM serije 3. Ta je z luknjanimi karticami vnesel v sistem dobavnice in na njihovi podlagi izpisal račune (fakture), ki so jih pozneje poslali kupcem. Vse ostalo je bilo voden ročno.

Leta 1985 so v podjetje prišli prvi štirje računalniki – računalniki Partner, ki jih je izdelovalo podjetje Iskra Delta. Medsebojno niso bili povezani v omrežje, ampak so delovali samostojno. Na njih so delovale aplikacije za vodenje:

- glavne knjige,
- saldakontov,
- materialne bilance in kalkulacije kupljenih surovin in embalaže.

Na četrtem računalniku so se sestavljali dopisi in izdelovale recepture za izdelke. Zanimivo je, da je obračun plač izvajal skupni računski center takratnega SOZD Kemija, ki je deloval v podjetju Belinka. V podjetju so se pripravili podatki za plače, podatki so se v skupnem računskem centru vnesli in obdelali, tam so se plače tudi izpisale in v podjetje vrnilo v ovojnicah. Mimogrede, iz skupnega računskega centra je nastalo današnje podjetje SRC.SI. Vse aplikacije so bile kupljene – v podjetju še ni bilo lastnega razvoja programske opreme.

Partnerje je nadomestil miniračunalnik PDP11/24, ki ga je v Jugoslaviji prodajala Iskra Delta pod imenom Delta 800. Sočasno z njim je v podjetje prišel tudi prvi osebni računalnik (ang. *Personal Computer*, krat. PC), ki je deloval v operacijskem sistemu MS-DOS. Na njem je tekla aplikacija za normiranje proizvodnje, ki je bila tudi izvor podatkov za plače.

Z Delto 800 se je začel tudi lastni razvoj programov. Programi so bili napisani v jeziku cobol, v podjetju sta bila zaposlena dva razvijalca programske opreme. Za dostop do podatkov sta uporabljala sistem Datatrieve. Razvila sta aplikacije, ki pokrivale predvsem področje komerciale:

- Materialno in blagovno knjigovodstvo s planiranjem in tokom denarja.
- Tiskanje računov na podlagi ročnih dobavnic iz skladišča gotovih izdelkov.
- Pozneje sta bila dodana sprejemanje naročil in izdelava dobavnic iz skladišča gotovih izdelkov na podlagi sprejetih naročil.
- Izdelala se je aplikacija za vodenje receptur in spremljanje proizvodnje (produkcije). Na podlagi podatkov o proizvodnji in recepturi se je začel samodejni odpis surovin in embalaže.
- Vodenje osnovnih sredstev in njihova amortizacija.
- Finančni program za menice.

- Za knjiženje v saldakonte se je izdelala temeljnica, ki se je prenesla na osebni računalnik, kjer je bila aplikacija za vodenje saldakontov.

Iz računalnikov Partner so se v osebne računalnike prenesle aplikacije, ki so prej delovale na Partnerjih (saldakonti, glavna knjiga). Osebni računalniki so bili povezani v omrežje, podjetje je dobilo tudi prvi strežnik, ki je deloval z operacijskim sistemom Novell NetWare v3.11.

PDP11 je bil leta 1991 zamenjan z računalnikom Digital microVAX 3900 z operacijskim sistemom VMS. Iz PDP so se na VAX prenesle vse aplikacije, niso pa se mogli prenesti programski zasloni. Te je bilo treba izdelati na novo, tokrat s programom FMS.

Na VAX-u se je vodilo celotno materialno poslovanje skladišča gotovih izdelkov, skladišča surovin in embalaže ter skladišča pomožnega materiala in rezervnih delov. Poleg vodenja materialnega poslovanja so na njem delovale tudi kalkulacije za kupljeno blago.

Dodana je bila aplikacija za planiranje prodaje in proizvodnje – tedenski, mesečni in letni plan.

Proizvodni del sistema je obsegal recepture in kosovnice za artikle, ki jih je delala družba, plane proizvodnje (letne, mesečne in tedenske) ter samodejni odpis porabljenih surovin in embalaže proizvedenih artiklov.

Najpomembnejši sklop programov so bili programi, namenjeni prodaji. Celotni proces, od vnosa naročila do izpisa dobavnice in računa, je bil pokrit s prodajnimi aplikacijami.

Tudi VAX je bil povezan v omrežje z osebnimi računalniki, tako da je bil prenos podatkov med sistemi lažji. Prenašale so se temeljnice za knjiženje v saldakontih in računi za odvisne družbe.

Osebni računalniki so bili povezani v tri omrežja Novell – finančno, maloprodajno in omrežje, namenjeno uporabnikom Comshare.

## **Podpora odvisnim podjetjem**

Z nastankom odvisnih podjetij (večkrat se uporablja tudi izraz hčerinsko podjetje) se je pojavila tudi potreba za informacijsko podporo v teh podjetjih. Naj poudarim, da je to čas druge polovice zadnjega desetletja preteklega tisočletja – internet še ni bil razširjen, najeti vodi so bili zelo dragi, programska oprema, ki je delovala v različnih okoljih, je bila namenjena multinacionalkam, ki so imele nekaj deset tisoč zaposlenih ...

Podpora je šla v dve smeri:

- Odvisno podjetje uporablja lastno programsko opremo.  
Ni sodelovanja med informacijskimi sistemi, izmenjava podatkov je ročna (elektronska pošta, preglednica). Odvisno podjetje samo skrbi za informacijsko podporo. V to smer je šel razvoj v družbah, kjer je bil pravni sistem drugačen kakor v matičnem podjetju – na Madžarskem in Češkem. Poleg prilagajanja lokalnim standardom so bile težave tudi s prevajanjem aplikacij v tuje jezike.
- Matično podjetje opremi odvisno družbo z enako programsko opremo, kakor jo uporablja samo.

Izmenjava podatkov poteka po modemski povezavi. Matična družba skrbi za strojno in programsko opremo. Programska oprema se ne prevaja, uporabniki se naučijo slovenskega jezika. Ta podpora je bila mogoča v družbah, kjer je bil pravni red podoben kakor v Sloveniji, se pravi v državah bivše Jugoslavije, kjer tudi jezik ni predstavljal večje ovire.

Zanimiv je bil način izdelave računov, ki so jih odvisna podjetja izdajala svojim kupcem. Naročila kupcev, ki so formalno kupili blago od odvisne družbe, so se vnašala v matični družbi. Ker ni bilo lokalnih skladišč, so se v matični družbi naročila tudi izvedla. Po izvršitvi naročila sta na VAX-u nastali dve fakturi. Prvo je izdala matična družba odvisni družbi, drugo pa odvisna družba končnemu kupcu. V prvi fakturi so bile uporabljene cene iz transfernega cenika, v drugi pa cene iz lokalnega cenika. Obe fakturi sta se prenesli na osebni računalnik v matični družbi. Prva se je samodejno knjižila v saldakonte, druga pa se je po modemski povezavi prenesla v odvisno družbo. Tam se je knjižila v saldakonte odvisne družbe in se izpisala na papir. To fakturo je prejel končni kupec.

Prenos podatkov se je preverjal s kontrolnimi izpisi – Slika 8.

Matična družba je tako izdelala vse fakture za končne kupce družb, ki so bile povezane v skupni informacijski sistem. Komunikacija je bila enosmerna in se podatki o plačilnih faktur in obveznostih do dobaviteljev niso samodejno prenašali v matično družbo.

### Slika 8: Kontrolni izpis prenosa faktur

```

=====
19-02-2003 KONTROLNI IZPIS PRENOSA RAČUNOV ZA ZAGREB ZA KONČNEGA KUPCA (DOKAČI)
=====
KUPEC      ST.RACUNA  DAT.RAC  ROK  TEM          ZNESEK      DAVEK  DEV  VEZA
=====
1175700 ZD2003/   77  18.02.03  90  200302      100,316.01    18,089.77 385  519
1097900 ZD2003/   78  18.02.03  75  200302       77,099.95    13,903.27 385  520
1097900 ZD2003/   79  18.02.03  75  200302       1,200.04      216.40 385  521
1210500 ZD2003/   80  18.02.03  90  200302      108,739.09    19,608.69 385  527
=====
SKUPAJ ZNESEK ZA PRENOS : 4                287,355.09    51,818.13
=====

```

Vir: Interna dokumentacija podjetja.

### Podpora odločanju

Na VAX-u, ki je bil namenjen operativnemu delu, je delovalo tudi nekaj aplikacij, ki so bile prvi zametki sistema za podporo odločanja v podjetju. Na njem je bil razvit in široko uporabljan informacijski sistem za spremljanje prodaje imenovan Prodajni informacijski sistem (krajše PIS). Uporabniki so imeli na voljo naslednje podatke:

- Skupna prodaja lastnih artiklov v izbranem in preteklem letu po mesecih v kilogramih z indeksom prodaje (Slika 9).

**Slika 9: Zaslonska slika informacijskega sistema PIS – skupna prodaja lastnih artiklov**

**Prodaja po mesecih v kilogramih**

(Brez trgovskega blaga in palet)

Mesec	2001	Kumulativa	2002	Kumulativa	Indeks	Kum.indeks
JAN	3,215,786	3,215,786	2,921,227	2,921,227	90.84	90.84
FEB	3,832,271	7,048,058	5,278,712	8,199,940	137.74	116.34
MAR	7,093,923	14,141,981	7,364,927	15,564,868	103.82	110.06
APR	7,609,449	21,751,431	9,062,132	24,627,000	119.09	113.22
MAJ	9,460,004	31,211,435	10,023,577	34,650,578	105.96	111.02
JUN	7,570,780	38,782,215	8,860,346	43,510,925	117.03	112.19
JUL	8,873,250	47,655,466	10,348,506	53,859,431	116.63	113.02
AVG	8,425,740	56,081,206	9,895,994	63,755,426	117.45	113.68
SEP	7,026,026	63,107,233	8,854,939	72,610,365	126.03	115.06
OKT	7,105,159	70,212,392	7,270,011	79,880,377	102.32	113.77
NOV	4,398,488	74,610,881	5,207,366	85,087,743	118.39	114.04
DEC	2,933,127	77,544,008	3,536,417	88,624,161	120.57	114.29
Skupaj	77,544,008	77,544,008	88,624,161	88,624,161	114.29	114.29

Vir: Interna dokumentacija podjetja in avtor.

- Prodaja lastnih artiklov v izbranem in preteklem letu v kilogramih po državah z deležem in indeksom prodaje.
- Prodaja lastnih artiklov v izbranem in preteklem letu v izbrani državi v kilogramih z indeksom prodaje (Slika 10).

**Slika 10: PIS - prodaja lastnih artiklov na izbrani državi v kilogramih**

Sifra drzave :385 Naziv drzave :HRVA[KA]

V kilogramih brez trgovskega blaga, palet in katalogov

Vrsta blaga	Prodano 99	Prodano 00	Indeks
ACRYLCOLOR	418.987	472.997	112,89
AKRILDEKOR	921	1.138	123,62
AKRIL EMULZIJA	432.024	458.596	106,15
ALGICID	2.800	4.288	153,14
AMIKOL	47	737	
AKRILTOP	220	670	304,54
ACRYLCOLOR RELIEF	263	384	145,80
JUB ANTINIKOTINSKA B.	4.675	4.445	95,09
BIO APNENA BARVA	85.204	119.462	140,20
BIO APNENA FAS.B.	830	662	79,78
ARTCOLOR	1.894	2.588	136,64
COLORANT		5.845	
DIPI COLOR	44.529	53.421	119,96
DENIKOL	10.870	12.269	112,87
EPOKSIL	603	2.088	346,26
JUBOCOLOR	12.541	11.548	92,08
JUBOCID	22.170	12.330	55,61

82 Zapisov v izboru

Vir: Interna dokumentacija podjetja.

- Prodaja lastnih artiklov v izbranem in preteklem letu v izbrani državi po mesecih v kilogramih z indeksom prodaje (Slika 11).

**Slika 11: PIS - prodaja v izbrani državi po mesecih v kilogramih.**

Sifra drzave :385 Naziv drzave :HRVA[KA]

V kilogramih brez trgovskega blaga, palet in katalogov

Mesec	1999	Kumulativa	2000	Kumulativa	Indeks	Kum.indeks
JAN	763.315	763.315	576.367	576.367	75,50	75,50
FEB	640.978	1.404.293	833.996	1.410.363	130,11	100,43
MAR	1.734.877	3.139.170	1.414.968	2.825.331	81,56	90,00
APR	1.463.188	4.602.358	1.781.722	4.607.054	121,76	100,10
MAJ	1.893.951	6.496.310	2.232.980	6.840.034	117,90	105,29
JUN	1.949.510	8.445.820	2.092.232	8.932.267	107,32	105,75
JUL	1.929.565	10.375.385	1.785.069	10.717.336	92,51	103,29
AVG	1.953.485	12.328.871	2.090.447	12.807.784	107,01	103,88
SEP	1.537.984	13.866.855	1.463.299	14.271.083	95,14	102,91
OKT	1.238.968	15.105.824	1.356.618	15.627.702	109,49	103,45
NOV	725.688	15.831.513	1.133.199	16.760.901	156,15	105,87
DEC	434.981	16.266.494	558.732	17.319.634	128,44	106,47
Skupaj	16.266.494	16.266.494	17.319.634	17.319.634		106,47

Vir: Interna dokumentacija podjetja.

- Dnevna prodaja lastnih artiklov.

Podatki so se pripravljali iz operativnih baz vsako noč in so bili na voljo izbranim uporabnikom v komercialnem in tehničnem sektorju.

Največja pomanjkljivost zgoraj naštetih pregledov je bila neprilagodljivost. Uporabnik ni mogel na noben način spremeniti pogleda na podatke. Če so želeli druge podatke ali samo enake podatke v drugi obliki, je moral oddelek za IT izdelati oziroma spremeniti aplikacijo za izdelavo in prikaz podatkov.

Poleg tega so bili na voljo samo podatki o količinski prodaji po državah in artiklih, ni pa bilo podatkov o vrednosti prodaje, o prodaji artiklov posameznim kupcem in podobno.

Ostali podatki, ki so jih potrebovali uporabniki, so bili na voljo samo v obliki papirnatih poročil. Večina se je izdelovala enkrat mesečno – poročila o nabavi, zalogah, terjatvah in obveznostih, plačah in podobna, nekatera pa še redkeje.

Zaradi potreb po podatkih je družba leta 1994 začela izdelovati prvi pravi sistem za podporo odločanju. Za programsko opremo je bil izbran programski paket Commander družbe Comshare. V paketu je bilo razvojno orodje za modeliranje Commander Builder in Commander Prism, na strani uporabnikov pa Commander Desktop, s katerim je uporabnik pregledoval podatke. Kot zanimivost naj navedem, da je samo programska oprema za 15 uporabnikov takrat stala 25.500 švicarskih frankov (1.700 CHF na uporabnika).

Že za razvoj prvega sistema za podporo odločanja je bila sestavljena skupina, ki je vsebovala ključne prihodnje uporabnike sistema, člane oddelka za IT in predstavnika zunanjih izvajalcev. Skupina je pripravila zahteve, ki jih je moral uresničiti sistem, določili so se izvori podatkov, načini, termini polnjenja in preverjanja točnosti podatkov.


Sistem je bil sestavljen iz treh modulov:

- PRO – modul prodajnega področja,
- FIN – modul finančnih tokov, terjatev in obveznosti,
- RAČ – modul izkazov uspeha in stanja sredstev.

Podatki, ki jih je vseboval prodajni modul PRO, so se zbirali in izračunavali v sistemu VAX. V Commander so se samo prenesli in ta je skrbel le za prikazovanje podatkov. Prodajni modul je vseboval podatke o količinski in vrednostni prodaji artiklov. Uporabniki so lahko opazovali posameznega kupca, ti pa so bili združeni po državah. Poleg osnovnih podatkov so jim bili na voljo tudi različni indeksi.

Poleg količinskih podatkov o prodaji, ki je bila uporabnikom na voljo že na VAX-u, so lahko tukaj spremljali tudi finančne podatke o prodaji (Slika 12).

Slika 12: Podatki o dnevni prodaji po trgih v kilogramih, SIT in DEM



SKUPAJ IZD. ZA	SKUPAJ JUB.	ZA OBDOBJE, 01. DAN,	POLNJENO DNE 2. 08. 1996		
01. DAN	KOLIČINA	PAK. VRED. SIT	PAK. VRED. DEM	POVPR/SIT	POVPR/DEM
1. SLOVENIJA	119,243	10,934,947	122,536	91.70	1.03
2. HRVAŠKA	18,588	1,665,296	18,661	89.59	1.00
3. ČEŠKA	18,724	1,437,686	16,110	76.78	0.86
4. SLOVAŠKA	0	0	0	0.00	0.00
5. MADŽARSKA	19,496	1,443,767	16,178	74.05	0.83
6. MAKEDONIJA	0	0	0	0.00	0.00
7. BOSNA	61,993	4,387,167	49,161	70.77	0.79
8. RUSIJA	0	0	0	0.00	0.00
9. OSTALO	0	0	0	0.00	0.00
=====	=====	=====	=====	=====	=====
SKUPAJ	238,044	19,868,863	222,646	83.47	0.94

Vir: Interna dokumentacija podjetja.

V finančnem modulu FIN so bili podatki, ki so se prenašali iz saldakontov. Prenos podatkov je potekal samodejno – v saldakontih se je izdelala datoteka s podatki, ki je bila podlaga za vnos v Commander. Prenašali so se podatki o stanju terjatev in obveznosti.

Računovodski modul RAC je vseboval podatke o mesečnih stanjih na kontih glavne knjige. Polnjenje podatkov je bilo ročno, enkrat mesečno so se podatki vnašali v bazo Commander. Poleg bilančnih podatkov matičnega podjetja so se vnašali tudi podatki iz podjetij v tujini.

S povečevanjem obsega poslovanja, predvsem s širjenjem asortimenta artiklov, je obstoječi sistem počasi postajal premalo zmogljiv za potrebe družbe. Sistem Commander ni dopuščal več kot 128 članov dimenzije. Prodajni program družbe pa je takrat že obsegal več artiklov in zato ni bilo mogoče spremljati posameznega artikla, kot so si želeli uporabniki. Delna rešitev je bilo grupiranje artiklov v grupe (skupine). Tako so bili v skupini Jupol zbrani vsi artikli te blagovne znamke ne glede na količino pakiranja, trg, za katerega so bili izdelani, ali barvno nianso.

Čeprav je Commander dopuščal, da so si uporabniki sami pripravljali poglede na podatke, so bili tako dobljeni pogledi zelo počasni in praktično neuporabni. Tako se je praktična uporaba omejila na poglede, ki so jih uporabniki definirali v fazi razvoja sistema oziroma so jih vzdrževalci vgradili v sistem.

Commander niso vzdrževali člani oddelka za IT, ampak zunanji sodelavci. Tudi to je bil razlog, da je za uresničitev sprememb, ki so jih predlagali uporabniki, bilo potrebnega veliko časa in usklajevanja. Sam sistem pa se je s časom izkazal kot dokaj neprilagodljiv.

Sčasoma je Commander zastarel – tekel je v sistemu MS-DOS, na delovnih postajah pa je že tekel okenski operacijski sistem – Microsoft Windows NT 4.0.

Leta 2000 je Uprava sklenila, da je treba obstoječi sistem za podporo odločanju posodobiti ali zamenjati.

#### 4.1.2 Projekt gradnje sistema za podporo odločanju

Pred formalnim začetkom projekta gradnje sistema za podporo odločanja se je oblikovala manjša skupina, sestavljena iz ključnih uporabnikov in članov oddelka IT. Naloga skupine je bila preveriti različne ponudbe ponudnikov za gradnjo sistemov za podporo odločanja. Skupina se je morala odločiti, ali bomo zaupali gradnjo sistema zunanjim sodelavcem, ali pa bo večina gradnje delo notranjega oddelka IT.

Po predstavitvi ponudnikov je skupina predlagala, da razvoj izvede notranji oddelek za IT. Po predstavitvi ponudb so ugotovili, da je večino dela, po ocenah med 80 % in 90 %, povezanega s pripravo izvornih podatkov in s čiščenjem teh podatkov. Tudi v primeru zunanjega razvoja sistema za podporo odločanju bi moral oddelek IT pripraviti podatke in jih v sodelovanju z uporabniki pripraviti za prenos v podatkovno skladišče.

Ker pa v oddelku IT ni bilo znanja s tega področja, smo k sodelovanju povabili tudi zunanje svetovalce.

Zaradi omejenih kadrovskih virov (razvoj je izvajal notranji IT oddelek) in pomanjkanja izkušenj z razvojem na tem področju je bilo sklenjeno, da bo razvoj sistema za podporo odločanja potekal v več korakih. V vsakem koraku naj bi informacijsko podprli določeni sektor oziroma njegov del. Prvi korak naj bi bil nadomestitev starega sistema za podporo odločanja za področje prodaje (PRO) z novim. To področje je za družbo najbolj kritično in tu je vladala največja potreba po podatkih. Nadaljnji opisi se večinoma nanašajo na ta – prvi projekt.

Uprava je predlog sprejela in izvedbo prenove zaupala projektni skupini, ki je bila ustanovljena za ta namen.

#### Projektna skupina

Projektna skupina za gradnjo sistema za podporo odločanju je bila sestavljena iz naslednjih članov:

- član oddelka IT – vodja projekta,
- član oddelka IT – priprava podatkov,
- član oddelka IT – priprava podatkov,
- član oddelka IT – implementacija podatkovnega skladišča,
- član finančnega sektorja – področje računovodstva,
- član finančnega sektorja – področje financ,
- član komercialnega sektorja – področje prodaje,
- član komercialnega sektorja – področje domače nabave,
- član komercialnega sektorja – področje tuje prodaje,
- član tehničnega sektorja – področje proizvodnje,
- član tehničnega sektorja – področje vzdrževanja.



Projektno skupino so tvorili člani, ki so prihajali iz vseh sektorjev družbe. S tem smo želeli doseči, da bomo v sistem za podporo odločanja vključili vse bistvene podatke, ki jih potrebujejo uporabniki v vseh delih družbe. Člani projektne skupine so bili hkrati tudi ključni uporabniki obstoječega sistema za podporo odločanja in so dobro poznali obstoječi sistem, njegove zmožnosti in pomanjkljivosti.

Vsi člani projektne skupine so bili iz matične družbe, tako da odvisne družbe niso imele svojih predstavnikov v skupini. Razlogov za to odločitev je bilo več. Glavni razlog je bil tehnične narave – odvisne družbe, ki so bile povezane z matično družbo, so bile povezane le prek počasne modemske povezave, kar ni omogočalo, da bi bil sistem za podporo odločanja na voljo tudi njim. Z nekaterimi odvisnimi družbami pa elektronska komunikacija sploh ni bila vzpostavljena. Poleg tega se je večina odločitev za delovanje odvisnih družb sprejemala v matični družbi – imele so relativno malo možnosti samostojnega odločanja. Pozneje se je izkazalo, da bi bilo bolje, če bi v projektno skupino vključili tudi člane iz odvisnih družb.

Zaradi že uveljavljenega naziva DIS (Direktorski Informacijski Sistem) smo sistem za podporo odločanja za prodaje uporabili ta naziv, čeprav ni bil namenjen samo direktorjem, ampak širšemu krogu uporabnikov.

### **Planiranje in vodenje projekta DIS**

Za uspeh projekta DIS so bili člani projektne skupine zelo zainteresirani, zato je bila naloga vodje projekta olajšana, saj so mu bili sponzorji in udeleženci naklonjeni. Poglavitna naloga vodje je bila izdelava čim bolj realističnega časovnega in finančnega plana projekta. Časovni del plana je bilo posebej težko izdelati natančno, ker je članom projektne skupine manjkalo izkušenj na tem področju.

Lažje je bilo izdelati finančni plan, ker je potekala večina razvoja v družbi in se delo udeležencev v projektu ni finančno ovrednotilo. Ostale finančne obveznosti pa so bile bolj ali manj jasne (nakup programske in strojne opreme, stroški izobraževanja in zunanjskega svetovanja).

Večji del finančnega načrta je bil narejen že pred začetkom projekta – v fazi, ko se je Uprava odločala med zunanjim izvajanjem ali notranjim razvojem.

Razdelitev nalog med udeleženci je bila preprosta, saj so bili za posamezno področje v projektni skupini vključeni ključni uporabniki s tega področja. Tudi razporeditev dela med člani Oddelka IT je bila preprosta.

Najtežje delo pri vodenju projekta je bilo usklajevanje časovnih terminov sestankov in rokov za pripravo gradiva med udeleženci iz različnih sektorjev. Ker je bil razvoj projekta pri ključnih uporabnikih le dodatna naloga, ki so jo morali izpolniti, nikakor pa zaradi dela pri projektu niso smeli zanemariti svojih vsakdanjih nalog, je bilo časovno usklajevanje zahtevno. Občasno je prihajalo zaradi zamud pri pripravi gradiva tudi do zamud pri časovnih rokih, ki so bili postavljeni v časovnem planu.

Za vodenje in spremljanje projekta smo uporabljali Microsoft Project, večina komunikacij je potekala po elektronski pošti. Enkrat mesečno so potekali sestanki vseh članov projektne skupine, na katerih je bil predstavljen potek projekta in so se določili nadaljnji koraki v razvoju. Med rednimi sestanki so potekali sestanki, ki so se jih udeleževali samo tisti člani skupine, ki so bili nujni za sprejemanje odločitev. Navadno je bil to ključni uporabnik in večina razvijalcev iz Oddelka IT.

Finančni plan razvoja ni vključeval stroškov dela članov projektne skupine, vključeval je samo stroške zunanjega svetovanja (Tabela 1).

Ko smo se odločili za lastni razvoj sistema za podporo odločanja, smo sklenili, da mora sistem delovati na programski in strojni opremi, ki jo v podjetju že uporabljamo in imamo zanjo usposobljene strokovnjake. Posledično v planu niso vključeni stroški za:

- Izobraževanje za programsko opremo SUPB.
- Nakup orodij za zajem in čiščenje podatkov ter za izobraževanje zanje.
- Sistem izdelave rezervnih kopij in ostale infrastrukture (omrežna oprema, neprekinjeno napajanje, varovanje ...).
- Izobraževanja uporabnikov, ker je bilo izobraževanje interno – v prostorih podjetja so ga vodili zaposleni.

**Tabela 1: Finančni plan razvoja DIS**

<b>Strošek</b>	<b>Vrednost [SIT]</b>	<b>Vrednost [€]</b>
Zunanje svetovanje	500.000	2.500
Programska oprema – poizvedovalni sistem	1.000.000	5.000
Izobraževanje za poizvedovalni sistem	1.000.000	5.000
Programska oprema – Microsoft Windows Server in SUPB	500.000	2.500
Microsoft SQL Server		
Strojna oprema – strežnik za podatkovno skladišče	1.500.000	7.500
<b>Skupaj</b>	<b>4.500.000</b>	<b>22.500</b>

Vir: Interna dokumentacija podjetja.

## 4.2 Potek gradnje DIS

V prvem razvojnem koraku (razvoj DIS) smo bili člani projektne skupine prepričani, da je bila sama vsebina sistema že dobro definirana v sistemu PRO. Uporabniki so bili s podatki, ki so jim bili na voljo, zadovoljni, želeli so si le razširitve.

Tako so bili glavni razvojni koraki usmerjeni bolj na tehnologijo, vprašanja, povezana z vsebino podatkov, so se reševala sproti:

- Določitev tehnološke osnove.
- Določitev orodij za poizvedovanje in izdelavo poročil.

- Zajem strukture obstoječega sistema za podporo odločanja, nato osvežitev in dopolnitev sistema.
- Zajem, čiščenje in preoblikovanje podatkov in polnjenje podatkovnega skladišča.
- Izdelava poročil in izobraževanje uporabnikov za uporabo orodij za poizvedovanje.

#### 4.2.1 Glavni razvojni koraki

##### Določitev tehnološke osnove

Tehnološka osnova sistema je bila v veliki meri določena že z odločitvijo, da v podjetje ne bomo uvajali različnih sistemov, če to ni nujno. Za to odločitev ni bila najpomembnejša cena dodatnega sistema, ampak cena, ki jo plačujemo za vzdrževanje različnih sistemov, predvsem pa za povezave med njimi.

Čeprav so v teoriji vsi sistemi medsebojno povezljivi, se v praksi izkaže, da so že sami prenosi med sistemi vir težav. Poleg tega moramo imeti, če ne bomo uporabili zunanjih izvajalcev, izobražene strokovnjake za vse sisteme, ki jih imamo v podjetju. Neprenehoma je treba skrbeti za nameščanje popravkov, novih različic ... Novi popravki in nove različice večkrat povzročijo nezdržljivost med različnimi sistemi, ki jih odpravijo šele novi popravki ...

Tako smo se odločili, da bo naš sistem sestavljen iz:

- Operacijskega sistema strežnika: Microsoft Windows Server NT. Del tega sistema je tudi spletni strežnik (Internet Information Server).
- Sistema za upravljanje podatkovnih baz: Microsoft SQL Server. Ta sistem vsebuje sistem za vodenje podatkovnega skladišča (Microsoft OLAP Server<sup>2</sup>) ter orodja za zajem, čiščenje in preoblikovanje podatkov (DTS).
- Orodja za razvoj programske opreme: Microsoft Visual Studio.

##### Določitev orodij za poizvedovanje in poročanje

Najtežja je bila verjetno izbira novih orodij za poizvedovanje in poročanje. V podjetje smo povabili različne ponudnike sistemov za podporo odločanja, ki so nam prikazali uporabnost svojih sistemov.

Del ponudnikov smo izločili z odločitvijo o enotni tehnološki osnovi. Tako so odpadle rešitve, ki temeljijo na drugih sistemih za upravljanje podatkovnih baz (npr. Oracle).

Naše naslednje zahteve so bile, da orodje omogoča uporabnikom izdelovanje lastnih poročil in vnaprej definiranih poročil. Seveda mora omogočiti tudi takojšne poizvedbe, izvoz podatkov v tabelaričen program in podobno.

---

<sup>2</sup> Microsoft OLAP Server se je v različici Microsoft SQL Server 2000 preimenoval v Microsoft Analysis Server.

Od orodja smo zahtevali tudi podporo za lokalno valuto, pravi prikaz decimalnega ločila (vejica) in možnost uporabe šumnikov.

Z razvojnega stališča smo želeli rešitev, ki omogoča, da tudi v samem orodju lahko programiramo, oziroma da lahko dele orodja vključimo v svoje aplikacije.

Zaradi slabih izkušenj s programsko opremo, ki ni imela zastopnika v Sloveniji, je bila naša zahteva tudi ta, da mora imeti orodje v Sloveniji zaupanja vrednega distributerja.

Odločitev skupine je bila, da izberemo orodje ProClarity podjetja Knosys<sup>3</sup>. Podjetje je neposredni partner Microsofta pri razvoju poslovne inteligence, omogoča programiranje v VBA (ang. *Visual Basic For Applications*, krat. VBA), uporablja standardni jezik za poizvedovanje po podatkovnem skladišču (ang. *Multidimensional Expressions*, krat. MDX) in je sestavljen iz gradnikov, ki jih lahko vključimo v svoje programe in na spletni strežnik.

### **Zajem strukture obstoječega sistema**

V planu projekta smo načrtovali, da bomo logični podatkovni model sistema dobili z analizo obstoječega sistema za podporo odločanja PRO. Tega naj bi samo nadgradili. Vendar se je izkazalo, da je v času od razvoja tega sistema prišlo do takih sprememb v operativnem sistemu, da smo lahko iz starega sistem prenesli samo najbolj grobo strukturo sistema.

Še enkrat je bilo treba definirati meritve, dimenzije in njihovo strukturo. Poiskati je bilo treba nove izvore podatkov, saj so se operativni podatki v sistem PRO prenašali iz aplikacij v operativnem sistemu, ki so živele samo zato, da so posredovale podatke iz novih aplikacij. Zaradi tega »verižnega« prenosu so bili podatki v sistemu PRO okrnjeni, saj so do njega prihajali prek aplikacij, ki se niso več razvijale. Seveda pa so bile s tem obremenjene tudi nove aplikacije, ki so morale skrbeti za združljivost s starimi aplikacijami.

Tako smo poleg logičnega podatkovnega modela dobili tudi seznam aplikacij, ki smo jih lahko umaknili iz operativnega sistema.

Na novo je bilo treba opredeliti pravice uporabnikov za dostop do podatkov v podatkovnem skladišču. Uprava družbe določi, katere podatke lahko vidi posamezni uporabnik.

### **Zajem, čiščenje, preoblikovanje podatkov in polnjenje podatkovnega skladišča**

Sočasno z dogovarjanjem o podatkih, ki bodo na voljo v podatkovnem skladišču, smo določili mesta, s katerih se bodo črpali podatki.

Ker sta bila izvora podatkov na dveh različnih računalniških sistemih – novejša aplikacije so uporabljale Microsoft SQL Server, ki je deloval na Microsoft Windows Server, starejše pa v

---

<sup>3</sup> Podjetje Knosys Inc. se je pozneje preimenovalo v ProClarity Inc. ravno po njihovem najbolj uspelem produktu. Ime produkta pa je ostalo nespremenjeno.

RDB na sistemu VAX, smo sklenili, da bomo podatke shranili v operativno podatkovno shrambo (v nadaljevanju ODS).

Sam zajem podatkov ni bil problematičen, saj je bil izvor podatkov večinoma v aplikacijah, ki smo jih razvili v podjetju. Tako je bilo treba samo napisati procedure za izvoz podatkov in uvoz v ODS.

Ko pa smo podatke zbrali na enem mestu, se je pričelo časovno najbolj zamudno delo celotnega projekta – čiščenje in preoblikovanje podatkov.

Primer iz šifrantov poslovnih partnerjev – posamezni kupec je bil lahko vnesen v sistem večkrat povsem legalno – kot kupec veleprodaje v Sloveniji, kot kupec maloprodaje v Sloveniji in kot kupec v naših odvisnih podjetjih. Seveda pa so bile v šifrantu tudi prave podvojitve, ko je bil posamezen kupec zaradi napake vnesen večkrat. Po prvem čiščenju oziroma poenotenju baze kupcev smo vanjo začeli dodajati še dobavitelje. Tudi pri teh je prihajalo do podvajanj ...

Čiščenju šifrantov je sledilo preverjanje pravilnosti meritev. Tu smo odkrili napake, ki jih operativni sistem ne bi smel dopustiti. Tako smo na primer našli artikle, ki so bili prodani, a nikoli narejeni.

Med čiščenjem podatkov smo tudi ugotovili, da so starejši podatki še manj kakovostni od sedanjih. Tako smo se odločili, da se omejimo samo na podatke zadnjih petih let in ne desetih, kot smo načrtovali na začetku.

Po čiščenju in preoblikovanju podatkov je bil prenos v podatkovno skladišče preprost.

Čeprav smo že pri planiranju predvidevali, da je ta korak časovno najbolj potraten, smo bili neprijetno presenečeni nad napakami, ki so se pojavljale v podatkih, saj smo bili prepričani, da so naši operativni podatki zelo čisti. Tudi zunanji svetovalci so nam potrdili, da se v tem dolgem koraku najdejo napake na mestih, kjer jih nihče ne pričakuje.

Korak čiščenja z navzkrižnimi preverjanji in sodelovanjem uporabnikov, ki dobro poznajo vsebino podatkov, je poglobljen za zaupanje uporabnikov v točnost podatkov. Če podatki v tem koraku niso čisti (točni), je vse nadaljnje delo zaman, uporabniki ne bodo nikoli povsem zaupali podatkom in bodo vodili lastne ločene evidence.

Mogoče velja na tem mestu poudariti, da je za korak čiščenja nujno sodelovanje uporabnikov, ki dobro poznajo pomen podatkov. Nekateri podatki so lahko na videz povsem pravilni, ker uporabnik ne pozna pravega pomena podatkov (npr. račun za 10 mil SIT je lahko pri nekem podjetju nekaj povsem običajnega, pri drugem pa zelo verjetna napaka). Zato je v primeru, da čiščenje podatkov izvajajo zunanji izvajalci, nujno, da z njimi sodelujejo uporabniki, ki poznajo pomen podatkov. Ni dovolj, da z njimi sodelujejo samo člani internega oddelka za IT, ki navadno le površno poznajo pomen podatkov.

## **Izdelava poročil in orodja za poizvedovanja**

Osnova za poročila so bila že obstoječa poročila v sistemu PRO in PIS. Želeli smo, da se vsa poročila izdelajo dinamično, ko jih bo uporabnik želel. Vsa poročila smo izdelali v okviru orodja ProClarity in so na voljo uporabniku kot vnaprej definirani pogledi.

Ker je orodje ProClarity zadostovalo za vse potrebe uporabnikov, ni bilo treba razvijati orodja za poizvedovanja, v program smo samo vključili nekaj funkcij, ki so jih želeli uporabniki (npr. skok za leto dni v preteklost).

Takoj, ko so bili na voljo prvi podatki v podatkovnem skladišču, smo izdelali poročila, ki so jih uporabniki imeli za najbolj pomembna. Ko so bila poročila na voljo, smo izvedli izobraževanje za prvi krog uporabnikov. Ti uporabniki so v nadaljnjem razvoju sodelovali pri oblikovanju poročil in tudi pri logičnem preverjanju podatkov.

Po izdelavi prvih poročil so uporabniki ugotovili, da pogrešajo nekatere meritve, vendar so bile to meritve, ki so se lahko izpeljale iz meritev, ki so že bile v podatkovnem skladišču. Uporabniki pa tudi imajo možnost, da sami izvedejo preproste meritve iz že obstoječih (t.i. *calculated members*).

Pri izdelavi vnaprej pripravljenih poročil za uporabnike je prišla do izraza možnost orodja ProClarity, da uporabniki sami izdelajo poročila, ki jih lahko posredujejo drugim uporabnikom v obliki bližnjic. Naprednejši uporabniki so na ta način sami izdelali poročila, ki so jih razvijalci nato samo vključili na seznam vnaprej pripravljenih poročil.

Uporabniki, ki potrebujejo podatke za nadaljnje obdelave, ki jih sam sistem za podporo odločanja ne omogoča, lahko pridobijo podatke iz sistema z uporabo vrtilnih tabel v programu Microsoft Excel.

Nekaj podatkov o prodaji, ki so zanimivi za vse uporabnike v internem računalniškem omrežju, se objavlja tudi na internem spletnem strežniku.

### **4.2.2 Uspešnosti prvega koraka projekta DIS**

Prvi korak gradnje sistema za podporo odločanja DIS, prodajni del, je bil zelo uspešen.

Projekt ni bistveno presegel prvotnega časovnega roka, zamude so bile objektivne narave in so izvirale iz prezasedenosti članov projektne skupine – ključnih uporabnikov.

Stroški projekta so bili zaradi spremembe licenčnih pogojev pri programski opremi celo nižji od predvidenih. Kot pa sem že omenil, se stroški dela članov projektne skupine niso ovrednotili.

Bolj kot časovni in finančni okvir projekta je pomembno to, da so uporabniki začeli uporabljati in še uporabljajo novi sistem. V času sočasnega delovanja starega (PRO, PIS) in novega sistema (DIS) so uporabniki primerjali vrednosti v obeh sistemih in po odpravi začetnih nepravilnosti je bil novi sistem pravilnejši. Nekaj napak, ki so jih uporabniki pripisali novemu sistemu za podporo odločanju, je izviralo iz napak, ki so se zgodile v operativnem sistemu. Ko so bile tam napake odpravljene, so bili tudi podatki v DIS-u pravilni.

Večina uporabnikov niti ni opazila, da je stari sistem za podporo odločanja prenehal delovati. Tudi to je dokaz, da so uporabniki novi sistem začeli uporabljati.

### 4.2.3 Nadaljnji projekti razvoja sistema za podporo odločanja

Po uspešnem projektu DIS so začele deževati zahteve za pokrivanje novih področij s podatkovnimi skladišči. Uprava je sklenila, da bo naslednji korak gradnja podatkovnega skladišča s podatki o terjatvah in obveznostih, ki smo ga imenovali FIS (Finančni Informacijski Sistem). Vanj v času nastanka nismo vključili izkazov uspeha, ker se je podjetje pripravljalo na uvedbo novega knjigovodskega standarda.

Vzporedno z razvojem FIS naj bi se gradilo še podatkovno skladišče s podatki o nabavnem poslovanju – NAB (NABava).

Po vzpostavitvi FIS in NAB se je razvoj podatkovnih skladišč začasno ustavil. Poleg rednega vzdrževanja – predvsem gre tu za izdelavo novih bližnjic do poročil ob prehodu leta – se nova podatkovna skladišča dve leti niso razvijala (od 2002 do 2004).

#### Tehnične spremembe

Kot sem omenil zgoraj (glej 4.1.1), je bila povezava med matično družbo in odvisnimi družbami izvedena z modemskimi povezavami. Ko se je pojavila finančno sprejemljiva možnost, smo svoje odvisne družbe povezali v navidezno zasebno omrežje (VLAN) z modemi ISDN. Taka povezava je že omogočala vključitev podatkov iz odvisne družbe v matično, vendar smo prenašali res samo najbolj kritične podatke.

Ko pa je bilo možno tudi na področjih odvisnih družb vzpostaviti širokopasovne povezave (ADSL) do svetovnega spleta, je nastala možnost, da iz odvisnih družb pridobimo vse podatke, ki jih želijo upravljavci v matični družbi. Seveda pa smo sedaj imeli tudi možnost, da omogočimo odvisnim družbam pristop do podatkovnega skladišča na matični lokaciji.

Ko so uporabniki postali tudi upravitelji iz odvisnih družb, se je izkazalo, da nekaterih podatkov iz sistema za podporo odločanju ne morejo pridobiti. Primer – koliko so prodali iz svojega skladišča in koliko neposredno iz matičnega skladišča.

#### Spremembe v družbi

Leta 2004 je matična družba odprla prvi proizvodni obrat v odvisni družbi. Do tedaj so imele odvisne družbe samo skladišča ali pa tudi teh ne. Z odprtjem proizvodnje v odvisni družbi je postal sistem za podporo odločanja nezadosten – ni vključeval možnosti, da bi izvedeli, kje je bil proizvod narejen, izvedeti je bilo mogoče samo, kje je bil prodan.

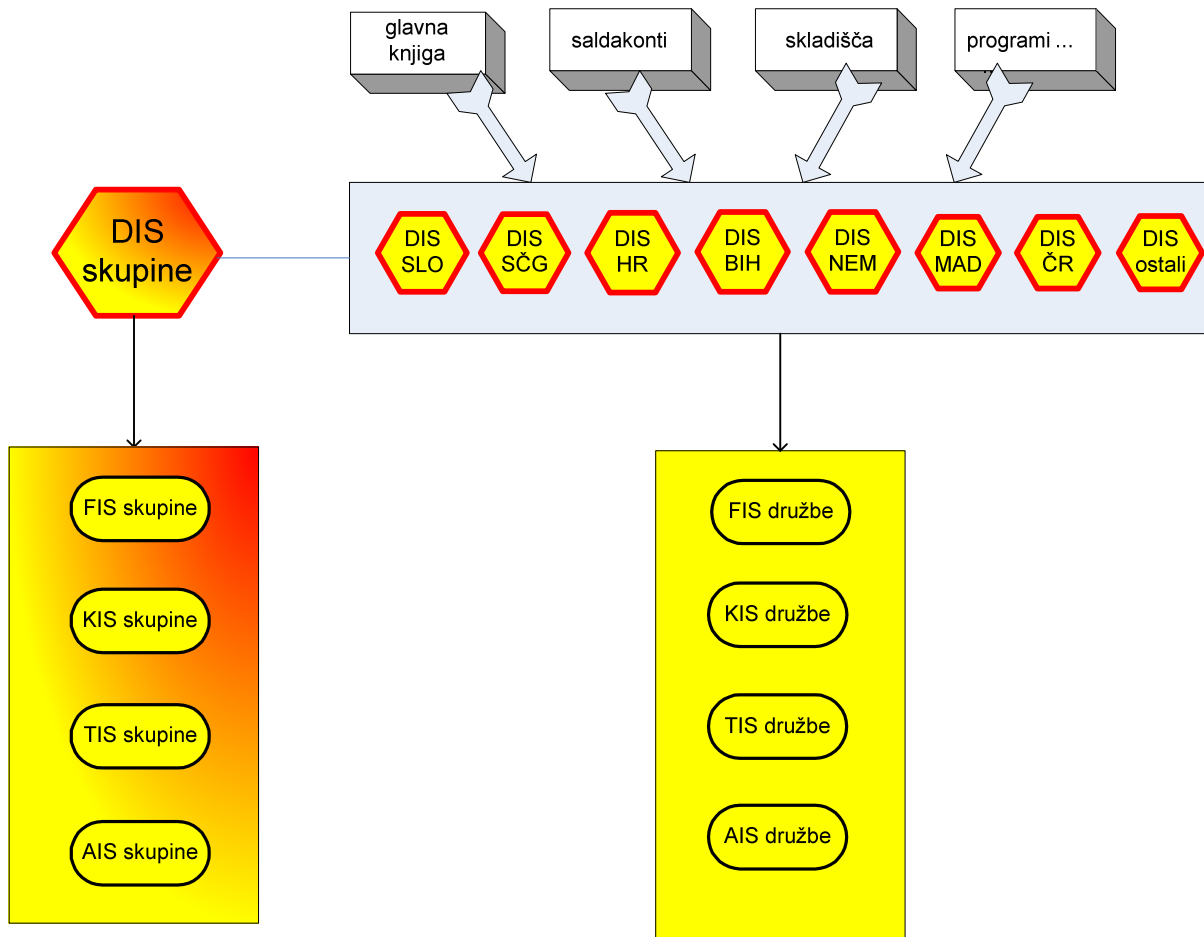
Sistem za podporo odločanja je bil prilagojen matični družbi in tako ni bil primeren za uporabo v odvisnih družbah. Izhajal je iz predpostavke, da se vse proizvode na matični lokaciji in da so skladišča v odvisnih družbah le vmesniki, ki ne vplivajo na skupno prodajo. Slednje je bilo res, dokler je družba prodajala proizvode, ki so imeli relativno majhno vrednost na kilogram. Polno ali prazno skladišče (nekaj sto ton) ni spremenilo prodajne statistike. Ko pa smo začeli prodajati proizvode, katerih kilogram stane nekaj 10.000 SIT, je postalo zelo pomembno, ali je ta proizvod v resnici prodan ali je še v skladišču odvisne družbe.

Če povzamem – sistem za podporo odločanja je po petih letih od gradnje zahteval korenite spremembe. Cilj sprememb je bila gradnja sistema za podporo odločanja **skupine** in ne več

samo matičnega podjetja. Zato se je leta 2005 začelo z razvojem prenovljenega in tudi povsem novega sistema za podporo odločanja, ki je bil tudi tokrat razdeljen na več delov. Zgradba je prikazana na Slika 13.

Matična družba je tudi dobila službo notranje kontrole in ena od prvih nalog je bila priprava strategije razvoja sistema za podporo odločanja v skupini in v posameznih družbah.

**Slika 13: Zgradba sistema za podporo odločanja**



Vir: Interna dokumentacija podjetja.

V trenutku pisanja naloge sta dokončana projekta gradnje naslednjih podatkovnih skladišč:

- KIS (Komercialni Informacijski Sistem),
- TIS (Tehnični Informacijski Sistem).

V izdelavi pa sta še naslednji podatkovni skladišči:

- FIS (Finančni Informacijski Sistem),
- AIS (Aministrativni Informacijski Sistem).



### 4.3 Vsebina podatkovnega skladišča

V nalogi sem se usmeril na podatkovno skladišče za podporo prodajni funkciji, v družbi je to del sistema KIS, ki je naslednik modula PRO. Poleg podatkov o prodaji vsebuje KIS tudi podatke o nabavi (NAB). Rad bi vam predstavil posebnosti, ki so predvsem posledica organizacije družbe, raznolike programske opreme in omejene informacijske vpetosti odvisnih družb v matično organizacijo.

V nadaljevanju bom za podatkovno skladišče s podatki o prodaji uporabljal izraz PRO.

V PRO so zbrani podatki:

- O planu prodaje.
- O sprejetih naročilih.
- O prodaji.

#### Plan prodaje

Plan prodaje je zajet iz letnega plana skupine. Ta je sestavljen iz planov prodaje po posameznih trgih. Seznam trgov, po katerih se je planirala prodaja za leto 2006:

- |                         |                |
|-------------------------|----------------|
| • Slovenija,            | • Češka,       |
| • Hrvaška,              | • Slovaška,    |
| • Bosna in Hercegovina, | • Nemčija,     |
| • Srbija,               | • Avstrija,    |
| • Črna Gora,            | • Rusija,      |
| • Kosovo,               | • Bolgarija in |
| • Makedonija,           | • ostalo.      |
| • Madžarska,            |                |

Planirajo se posamezni artikli v osnovni barvi (baza) in pigmenti, ki so potrebni za spremembo barve (nianse) artiklov. Pri nekaterih artiklih, ki jih v večjih količinah izdelujemo v določeni niansi, se planirajo tudi ti.

Blaga, ki ga ne proizvajamo sami, ne planiramo. Tu gre za trgovsko blago, ki ga kupujemo in prodajamo skupaj s svojimi izdelki (npr. stiropor za fasado). Lahko bi rekli, da družba ne planira prodaje, ampak proizvodnjo. Ker pa je prodaja trgovskega blaga vezana na prodajo naših artiklov, je tak plan povsem primeren.

Osnovni plan je izdelan za celo leto, potem ga na podlagi uteži, ki jih določi planer in so odvisne od pretekle prodaje in pričakovanj, razbijemo po mesecih.

Že v planu ločimo, kateri izdelki se bodo izdelovali v posameznem obratu – trenutno so to trije obrati – dva na matični lokaciji in tretji v odvisni družbi.

Letni plan prodaje lastnih artiklov, razbit po mesecih in trgih, je prenesen v podatkovno skladišče, ko ga potrdi Uprava.

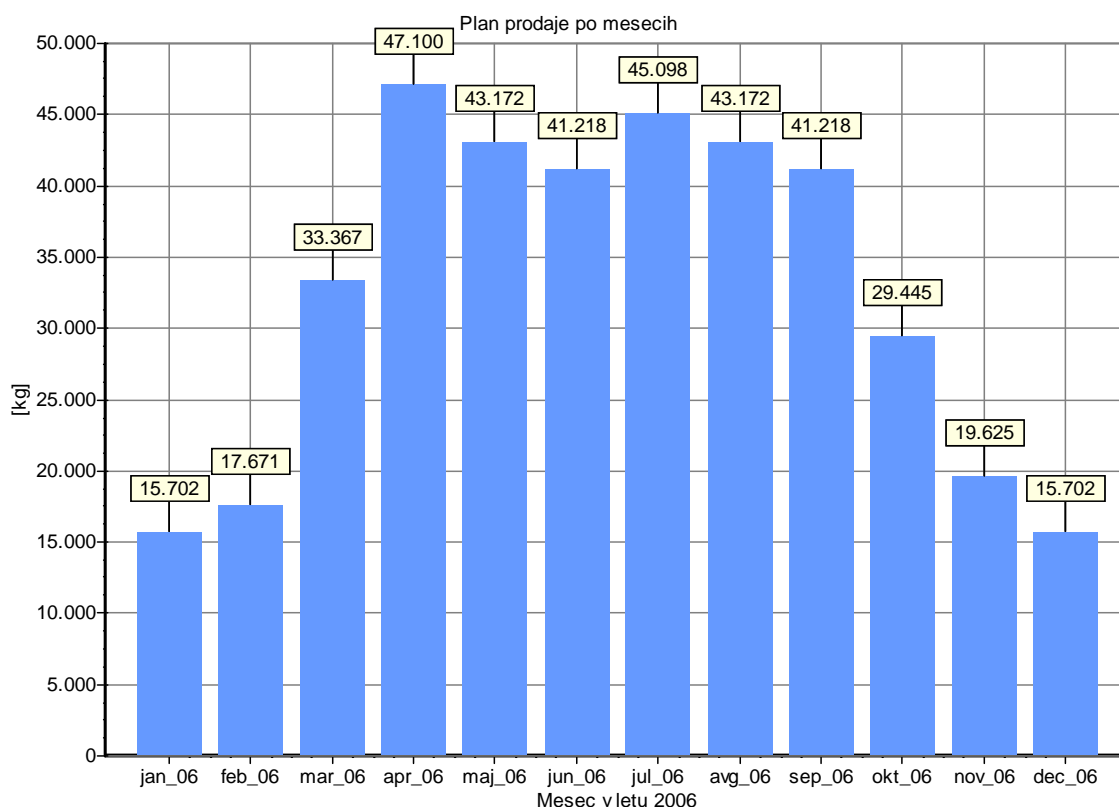
### Prenos iz operativnega sistema v podatkovno skladišče

Plan se izdelava v aplikaciji »Plan«. Prenos iz operativnega sistema v podatkovno skladišče je ročen. Avtomatizacija ni bila smiselna, saj gre za en prenos letno.

Sam postopek je preprost – podatki iz planskih tabel se prekopirajo v ustrezne tabele v operativni podatkovni shrambi. Dopolnijo se z manjkajočimi podatki iz ostalih šifrantov.

Skoraj vsako leto se program za planiranje spreminja – predvsem gre za reorganizacijo trgov, tako, da je treba vsako leto malo spremeniti procedure za prenos.

**Slika 14: Plan prodaje po mesecih v kilogramih (številke so spremenjene!)**



Vir: Avtor.

### Meritve in dimenzije

Osnovna meritev v planu prodaje je samo ena – planirana količina prodaje v kilogramih (kol\_kg\_pla) (Slika 14). Poleg te meritve je tu še planirana količina v enoti mere (kol\_em\_pla). Vse drugo so izračunane meritve.

Dimenzije, ki nastopajo v Planu prodaje, so: Partnerji, Blagovne skupine, Skupine produktov, Čas in Izvor.

Dimenzije bodo podrobneje opisane v nadaljevanju (4.3.2).

## Sprejeta naročila

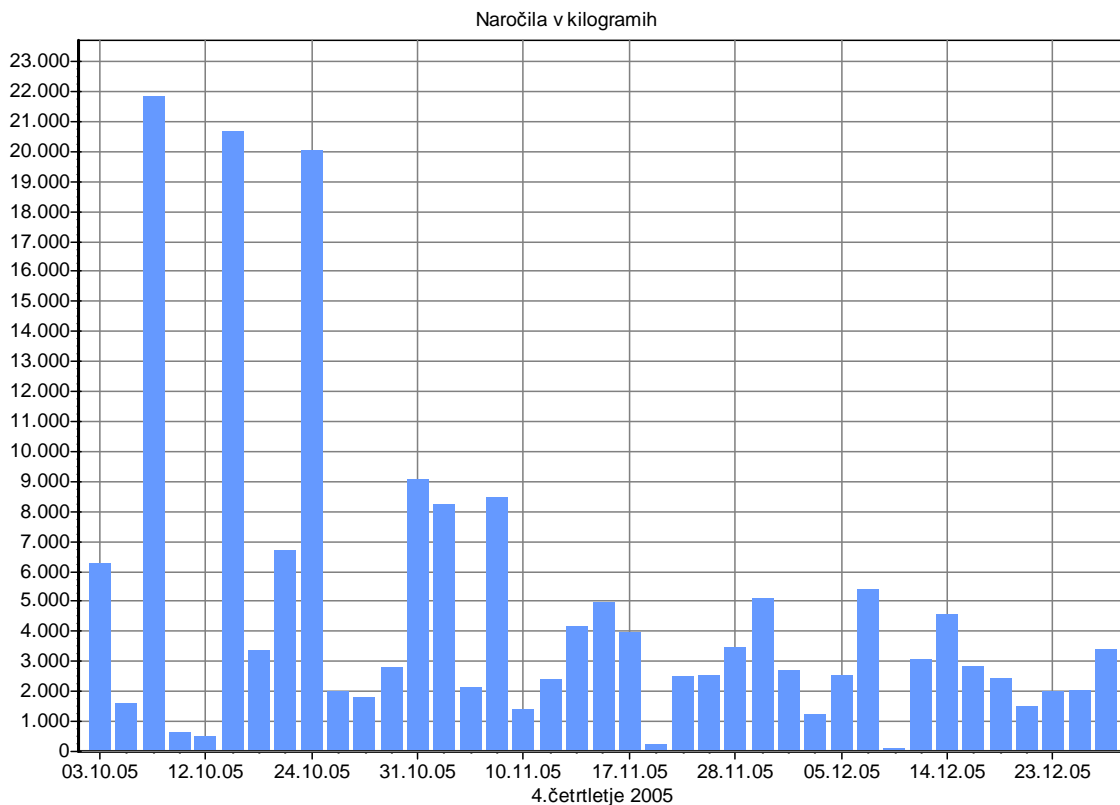
Pri naročilih, ki so vključena v PRO, moramo ločiti dve vrsti naročil – naročila za končnega kupca in za odvisno podjetje. Velja pravilo, da se tam, kjer je odvisna družba v celoti vključena v notranji informacijski sistem, vnašajo naročila končnega kupca. Kjer pa družba ni vključena v notranji informacijski sistem, se vnašajo naročila odvisne družbe.

Naročila se vnašajo v podatkovno bazo samo na mestih, kjer je možna lokalna izdaja blaga – torej tam, kjer so proizvodni obrati ali skladišče. Za ostale lokacije se vnašajo naročila v družbi, ki bo fizično izpolnila naročilo.

## Prenos iz operativnega sistema v podatkovno skladišče

Naročila se enkrat dnevno prenesejo na matično lokacijo, kjer se po avtomatiziranem sistemu prenesejo v operativno podatkovno shrambo.

**Slika 15: Prejeta naročila izbranega kupca v četrtem četrletju**



Vir: Avtor.

### **Meritve in dimenzije**

Pri naročilih sta meritvi, ki nas zanimata, naročena količina v kilogramih (`kol_kg_nar`) (Slika 15) in vrednost naročila brez popustov (`vre_fkr_brt_nar_val`). Vrednosti se vnašajo v operativno podatkovno shrambo v lokalni valuti, tam pa se preračunajo v tolarje (SIT) in evre (€).

Dimenzije, ki nastopajo v Sprejetih naročilih, so: Partnerji, Blagovne skupine, Skupine produktov, Čas, Izvor in Prodajalec.

Dimenzije bodo podrobneje opisane v nadaljevanju (4.3.2).

### **Prodaja**

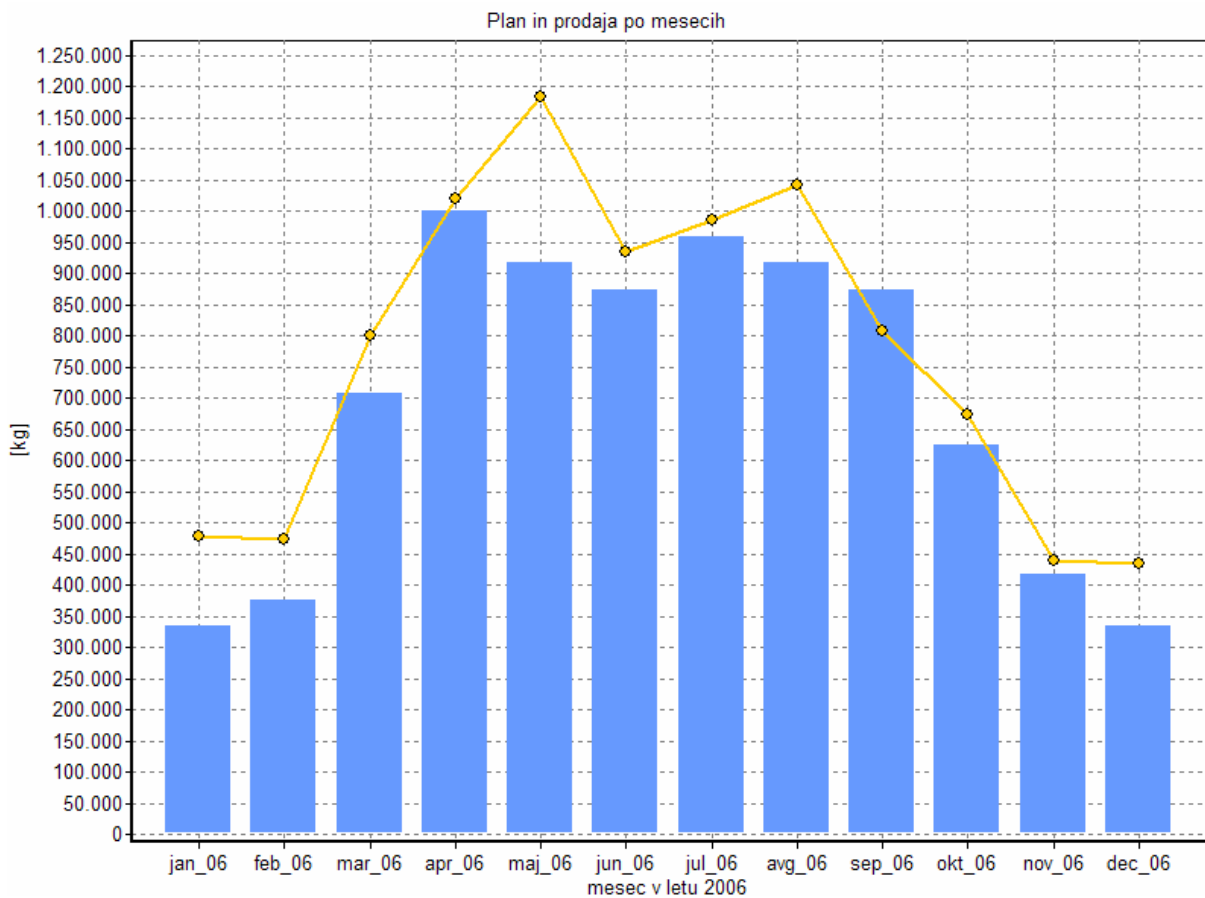
Podatki o prodaji so najpomembnejši del sistema. Vključeni so vsi podatki o prodaji blaga – prodaja med podjetji v skupini, prodaja končnim kupcem v veleprodaji, prodaja v maloprodaji in izdaja blaga za notranjo porabo. Poleg neposredne prodaje so vključeni tudi podatki o dobropisih in izdajnicah.

Sistem ne vključuje računov, ki jih izdaja podjetje za storitve in blago, ki niso del njegovega standardnega prodajnega programa. Tu so računi za prodajo električne energije, osnovnih sredstev, uporabo počitniških zmogljivosti in podobno.

### **Prenos iz operativnega sistema v podatkovno skladišče**

Izvor podatkov o prodaji sta dve prodajni aplikaciji – za veleprodajo in maloprodajo. Enkrat dnevno se prenesejo podatki iz odvisnih družb v matično lokacijo. Tam se samodejno prenesejo v operativno podatkovno shrambo, ki je izvor za prenos v podatkovno skladišče.

Slika 16: Plan in prodaja po mesecih v kilogramih za izbrani trg



Vir: Avtor.

### 4.3.1 Meritve

Pri prodaji sta pomembni tako količina kot vrednost prodaje. Količinsko prodajo merimo z dvema meritvama:

- kol\_em\_pro – prodaja v enoti mere.
- kol\_kg\_pro – prodaja v kilogramih, na Slika 16 je prikazana skupaj s planom prodaje.

Prodaja v kilogramih je pomembna, ker ta meritev pove ne le, koliko je bilo prodano, ampak pove tudi, kakšna logistika je potrebna za to prodajo. Pri prodaji 1.000 kosov ne vemo, kaj potrebujemo za odpremo, pri prodaji 50.000 kg pa takoj vemo, da potrebujemo nekaj tovornjakov.

Pri vrednosti prodaje imamo več osnovnih meritev:

- vre\_lst – lastna vrednost prodanih artiklov.
- vre\_fkr\_brt\_pro – fakturirana vrednost prodanih artiklov brez vseh popustov, to je prodana količina v enoti mere, pomnožena s ceno s cenika.
- vre\_got\_pop – vrednost gotovinskega popusta za vse artikle, ki je že izračunana na osnovnem računu.
- vre\_kol\_pop – vrednost količinskega popusta za vse artikle, ki je že izračunana na osnovnem računu.

- `vre_kol_pop_art` – vrednost dodatnega količinskega popusta na posamezen artikel, ki je že izračunana na osnovnem računu.
- `vre_tra_bon` – vrednost transportne bonifikacije, ki je izračunana na osnovnem računu.
- `vre_fin_dob` – vrednost dodatnih finančnih dobropisov. Ti niso zajeti na osnovnem računu, ampak na dobropisih, ki so izdani na podlagi računa.
- `vre_fin_rab` – vrednost dodatnih finančnih dobropisov, izdanih na podlagi celotnega prometa, ki ga je ustvaril kupec.

Poleg osnovnih meritev so na voljo tudi meritve, ki so izračunane iz osnovnih meritev:

<code>pro/pla_(kg)-grw_I</code>	Indeks doseganja plana – prodana količina glede na planirano količino v kilogramih.
<code>kol_kg_pro-grw_I</code>	Indeks rasti – količina prodanega artikla v kilogramih glede na preteklo obdobje
<code>kol_kg_pro-delez_partnerja</code>	Delež določenega partnerja v prodani količini za dimenzijo partnerji.
<code>kol_kg_pro-delez_artikla</code>	Delež določenega artikla v prodani količini za dimenzijo blagovne skupine.
<code>vre_fkr_net_pro</code>	Neto fakturirana vrednost prodanih artiklov. To je bruto vrednost, ki smo ji odšteli vse popuste, zajete na osnovnem računu. $+ \text{vre\_fkr\_brt\_pro}$ $- (\text{vre\_got\_pop} + \text{vre\_kol\_pop} + \text{vre\_kol\_pop\_art} + \text{vre\_tra\_bon})$ $= \text{vre\_fkr\_net\_pro}$
<code>vre_fkr_net_pro_1</code>	Neto fakturirana vrednost prodanih artiklov 1. Neto fakturirani vrednosti smo odšteli še dodatne finančne dobropise. $+ \text{vre\_fkr\_net\_pro\_1}$ $- \text{vre\_fin\_dob}$ $= \text{vre\_fkr\_net\_pro\_1}$
<code>vre_fkr_net_pro_2</code>	Neto fakturirana vrednost prodanih artiklov 2. Vrednost prodanih artiklov, pri kateri so upoštevani vsi izdani finančni popusti. $+ \text{vre\_fkr\_net\_pro\_1}$ $- \text{vre\_fin\_rab}$ $= \text{vre\_fkr\_net\_pro\_2}$
PVS	Prodajna vrednost minus variabilni stroški. Razlika med neto fakturirano vrednostjo in variabilnimi stroški. $+ \text{vre\_fkr\_net\_pro}$ $- \text{vre\_lst\_pro}$ $= \text{PVS}$

PVS_1	Prodajna vrednost minus variabilni stroški 1. Razlika med neto fakturirano vrednostjo, zmanjšano za finančne dobropise, in lastno vrednostjo prodanih artiklov. +vre_fkr_net_pro_1 <u>-vre_lst_pro</u> =PVS_1
PVS_2	Prodajna vrednost minus variabilni stroški 2. Razlika med neto fakturirano vrednostjo 2 in lastno vrednostjo prodanih artiklov. +vre_fkr_net_pro_2 <u>-vre_lst_pro</u> =PVS_2
PVS_2_na_vre_net_pro_2	PVS_2 na neto fakturirano vrednost artikla. +PVS_2 <u>/vre_fkr_net_pro_2</u> =PVS_2_na_vre_net_pro_2
PVS_2_na_kg	PVS_2 na kilogram prodanega artikla. +PVS_2 <u>/kol_kg_pro</u> =PVS_2_na_kg
Vre_rab	Skupna vrednost rabatov. +vre_fkr_brt_pro <u>-vre_fkr_net_pro_2</u> =vre_rab
Vre_rab_na_vre_brt_pro	Vrednost rabatov na bruto fakturirano vrednost prodanih artiklov. +vre_rab <u>/vre_fkr_brt_pro</u> =vre_rab_na_vre_brt_pro

Vse zgoraj naštetne vrednosti so v osnovni valuti – to je v valuti, ki nastopa na izdanem računu. V podatkovnem skladišču so označene s končnico \_val. Poleg osnovne valute so meritve na voljo tudi v tolarjih (končnica SIT) in evrih (končnica €). Primer meritve vrednosti rabatov:

- vre\_rab\_val – vrednost v denarni enoti računa (tolarji, kune, bosanske marke, dinarji, dolarji ...)
- vre\_rab\_sit – vrednost v tolarjih, iz osnovne valute se preračuna po srednjem tečaju za tolar nacionalne banke države, kjer je bil izdan račun.
- vre\_rab\_€ vrednost v evrih, iz osnovne valute se preračuna po srednjem tečaju za evro nacionalne banke države, kjer je bil izdan račun.

Za meritve, kjer je to smiselno, so v podatkovnem skladišču dodane tudi meritve o vrednosti meritve v celotnem časovnem obdobju od začetka leta do izbranega datuma – Leto do dneva, v uporabi je angleški izraz YearToDate. Te meritve so označene s končnico YTD. Primer:

- vre\_rab\_€ – meritev za izbrano časovno obdobje – npr. 6.2.2006.
- vre\_rab\_€-YTD – vsota meritev za časovno obdobje od začetka leta do izbranega časovnega obdobja – npr. od 1.1.2006 do 6.2.2006.

Za večino meritev je na voljo tudi izračunana meritev o rasti (padcu) meritve glede na isto časovno obdobje v preteklem letu. Te meritve so označene s končnico GRW\_I (iz angleške besede growth). Primer:

- vre\_rab\_€-GRW\_I – indeks rasti (padca) meritve glede na isto obdobje preteklega leta. Če je izbran september 2005, je to glede na september 2004.

Podobno so dodane meritve, ki prikazujejo absolutno rast (padec) meritve glede na isto časovno obdobje v preteklem letu. Te meritve so označene s končnico ABS (absolutno). Primer:

- vre\_rab\_€-ABS – absolutna rast (padec) meritve glede na isto obdobje preteklega leta. Če je izbran september 2005, je to razlika med vrednostjo meritve septembra 2005 in septembra 2004.

Dimenzije, ki nastopajo v Prodaji, so: Partnerji, Blagovne skupine, Skupine produktov, Čas, Izvor, Prodajalec in Računi.

Dimenzije bodo podrobneje opisane v nadaljevanju (4.3.2).

### 4.3.2 Dimenzije

#### Čas

Časovna dimenzija je skoraj vedno navzoča. Brez nje ni mogoče pregledovanje zgodovinskih podatkov oziroma primerjanje sedanjega stanja s preteklim.

Dimenzija je skladna tudi z ostalima podatkovnima skladiščema v družbi – FIS in NAB.

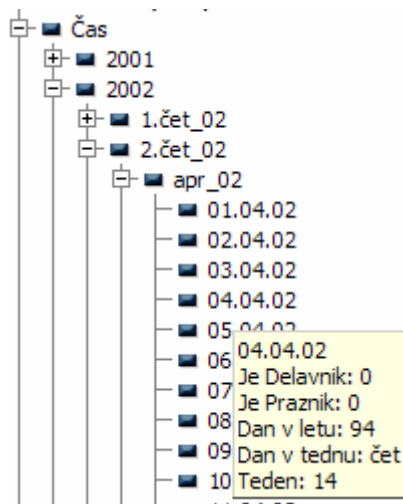
Časovna dimenzija se uporablja v vseh treh delih prodajnega podatkovnega skladišča – v planu, naročilih in prodaji.

Časovna dimenzija ima štiri nivoje (Slika 17):

- Leto.
- Četrtletje.
- Mesec.
- Dan.
  - Datum.
  - Ime dneva.
  - Zaporedni dan v letu.
  - Oznaka o prazniku.
  - Oznaka o delovnem dnevu.
  - Številka tedna, v katerem je dan.

To je edina dimenzija, ki se polni ne glede na podatke v podatkovnem skladišču. Polni se tako, da se za vsako leto v podatkovnem skladišču vnesejo vsi dnevi. Pri polnjenju skladišča z meritvami ni treba preverjati, ali obstaja datum meritve v časovni dimenziji.



**Slika 17: Časovna dimenzija s prikazom lastnosti elementa 04.04.2002**

Vir: Avtor.

## Partnerji

V dimenziji so podatki o poslovnih partnerjih, ki jih združujemo po geografskem ključu – najprej po državah, države pa po trgih.

Dimenzija je skladna z ostalima podatkovnima skladiščema v družbi – FIS in NAB.

Dimenzija Partnerji se uporablja z vsemi nivoji v dveh delih podatkovnega skladišča – prodaja in naročila. V delu plan se uporablja samo najvišji nivo – nivo trga, zato ni mogoče spremljati prodaje po posameznem partnerju ali državi, ampak samo po trgu.

Dimenzija Partnerji ima tri nivoje (Slika 18):

- Trg.
  - o Številčna šifra trga.
- Država.
  - o Številčna šifra države.
  - o ISO oznaka države.
- Naziv partnerja.
  - o Šifra partnerja.
  - o Davčna številka partnerja.
  - o Kraj.

Zgornja nivoja (trg in država) dimenzije se polnita ročno – ob odprtju novega trga ali na začetku prodaje v novo državo. To se dogaja redko, tako da avtomatizem ni potreben.

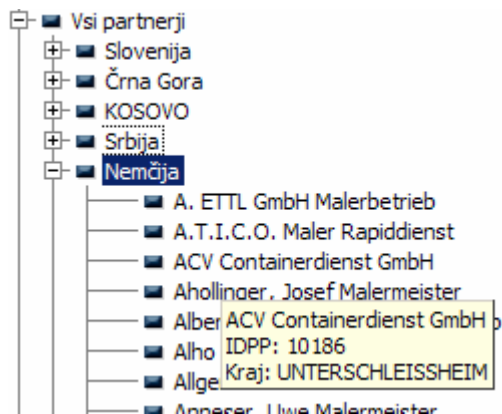
Novi partnerji se v dimenzijo dodajajo samodejno ob nastopu partnerja v meritvi. To je ob prejemu naročila od novega partnerja oziroma pri prodaji novemu partnerju.

Ob spremembi opisnih podatkov partnerja (naziv, kraj, davčna številka) se stari podatki prepisejo z novimi. Ne vodi se zgodovina sprememb na nivoju podatkovnega skladišča.

Če pride do sprememb v šifri partnerja (npr. združevanje partnerjev) ni predviden samodejni postopek, ampak se spremembe izvršijo glede na zahteve uporabnikov. Navadno se podatki v

podatkovnem skladišču ne združujejo, če pa gre za pomembne partnerje, je včasih treba ročno združiti podatke iz dveh različnih šifer na eno oziroma na novo.

**Slika 18: Dimenzija Partner s prikazom lastnosti elementa »ACV Containerdienst GmbH«**



Vir: Avtor.

### Blagovne skupine

V dimenziji so podatki o blagu, ki ga družba oziroma družbe prodajajo. Posamezni artikli so združeni v blagovne grupe, ki se združujejo v blagovne podskupine, te pa v blagovne skupine. Dimenzija ni skladna z ostalimi podatkovnimi skladišči.

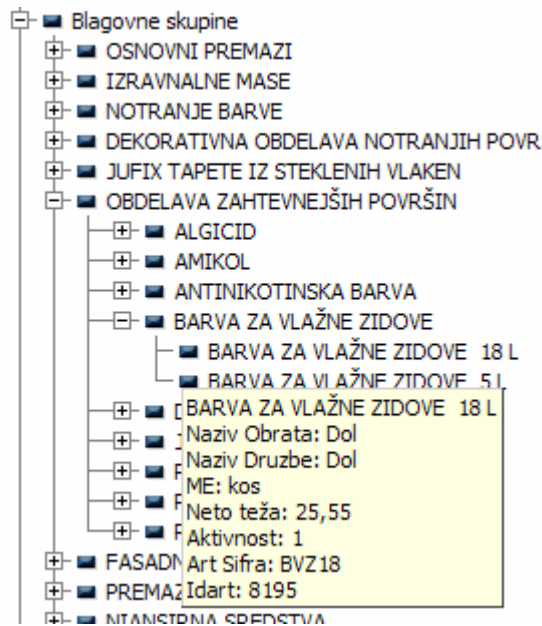
Dimenzija se uporablja v vseh treh delih prodajnega podatkovnega skladišča – v planu, naročilih in prodaji.

Dimenzija Blagovne skupine ima štiri nivoje (Slika 19):

- Blagovna skupina.
  - Šifra blagovne skupine.
- Blagovna podskupina.
  - Šifra blagovne podskupine.
- Blagovna grupa.
  - Šifra blagovne grupe
- Artikel.
  - Naziv obrata, kjer se artikel izdeluje.
  - Naziv družbe, kjer se artikel izdeluje.
  - Merska enota.
  - Neto teža.
  - Oznaka o aktivnosti.
  - Šifra artikla.
  - ID artikla.

Zgornji nivo – blagovna skupina – se polni ročno, ostali nivoji ob prvem nastopu artikla v meritvi. Pri artiklih, ki se izdelujejo na zalogo, je to pri vnosu v letni načrt, za veliko večino artiklov pa je to ob prvi prodaji.

**Slika 19: Dimenzija Blagovne skupine s prikazom lastnosti elementa "Barva za vlažne zidove 18 L"**



Vir: Avtor.

Spremembe v opisnih podatkih se uveljavljajo samodejno, tudi prehajanje artikla iz ene v drugo skupino. Premik artikla iz ene v drugo skupino, podskupino ali grupo je lahko posledica pomote pri vnosu v šifrant, občasno pa grupe artiklov razpadajo oziroma se združujejo. Ne vodi se zgodovina sprememb.

V preteklosti so se pojavljale zahteve, da bi ob zamenjavi artiklov, ki se ne izdelujejo več, z novimi, nadomestnimi, tudi združevali podatke o teh artiklih. Vendar so uporabniki ugotovili, da je dovolj, če primerjajo med seboj blagovne grupe, ne posamezne artikle. Na ta način imajo na voljo še vedno »čisto« zgodovino posameznega artikla.

## Skupine produktov

Medtem ko blagovne skupine sledijo tržni razdelitvi prodajnega asortimenta, z dimenzijo Skupine produktov kategoriziramo asortiment glede na zahteve uporabnikov, ki želijo spremljati prodajo tudi po drugih kriterijih.

Dimenzija ni skladna z drugimi podatkovnimi skladišči.

Dimenzija se uporablja v dveh delih prodajnega podatkovnega skladišča – v naročilih in prodaji. Ker planiramo samo artikle, ki jih izdelujemo sami, uporaba v tem podatkovnem skladišču ni smiselna.

Dimenzija Skupine produktov ima dva nivoja (Slika 20):

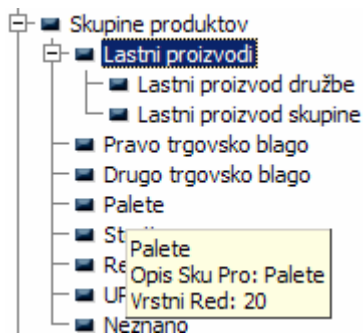
- Skupina produktov.
  - Opis skupine.
- Podskupina produktov.
  - Opis podskupine.

Dimenzija je bila zgrajena ročno in doslej še ni bilo zahtev po njeni spremembi. Ko se artikel vnese v podatkovno skladišče, se na podlagi podatkov iz šifranta artiklov samodejno uvrsti v pravo skupino.

Za uporabnike so najbolj pomembni artikli, ki jih družba proizvaja – Lastni proizvodi. Ker se ti izdelujejo v več družbah, je pomembno tudi to, ali se artikli prodajo neposredno iz družbe, ki je artikel proizvedla (Lastni proizvod družbe), ali iz katere druge družbe (Lastni proizvod skupine).

Tudi pri trgovskem blagu ločimo dve skupini produktov. Trgovsko blago, ki ga družba kupi in proda naprej družbi v skupini (Drugo trgovsko blago), želijo uporabniki spremljati ločeno od trgovskega blaga, ki ga družba kupi in proda končnemu kupcu (Pravo trgovsko blago). Drugo trgovsko blago so predvsem surovine, ki jih centralno nabavlja matična družba in prodaja ostalim družbam v skupini.

### Slika 20: Dimenzija Skupine produktov s prikazom lastnosti elementa "Paleta"



Vir: Avtor.

### Izvor

Z izvorom delimo prodajni asortiment glede na izvor artikla – za lastne artikle, kje je bil artikel proizveden, za kupljene artikle pa, kje je bil kupljen – v kateri družbi oziroma v katerem obratu družbe.

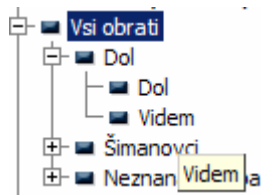
Dimenzija ni skladna z drugimi podatkovnimi skladišči.

Izvor se uporablja v vseh delih prodajnega podatkovnega skladišča – v planu, naročilih in prodaji.

Dimenzija Izvor ima dva nivoja (Slika 21):

- Družba.
- Obrat.

Dimenzija je zgrajena ročno. Ob vnosu meritve v podatkovno skladišče se uvrsti v pravo skupino na podlagi podatkov iz šifranta artiklov.

**Slika 21: Dimenzija Izvor s prikazom lastnosti elementa "Videm"**

Vir: Avtor.

## Prodajalec

Naziv dimenzije Prodajalec rahlo zavaja, ker ne gre za dimenzijo, v kateri bi bili zbrani prodajalci, ampak družbe, ki izdajajo blago. Družbe, ki izdajajo blago, so tiste, ki imajo svoje lastno skladišče.

Dimenzija ni skladna z drugimi podatkovnimi skladišči.

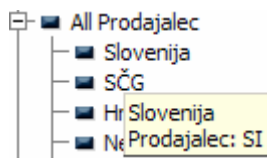
Dimenzija Prodajalec se uporablja v dveh delih prodajnega podatkovnega skladišča – v naročilih in prodaji. Čeprav bi lahko planirali, koliko bo prodano iz posameznega skladišča, se ta dimenzija ne uporablja v planu.

Dimenzija Prodajalec ima samo en nivo (Slika 22):

- Družba.
  - o ISO koda države.

Dimenzija je zgrajena ročno in se spremeni samo, ko skupina začne uporabljati novo skladišče.

Meritev uvrstimo samodejno v ustrezno skupino iz podatka o kraju izdaje artikla.

**Slika 22: Dimenzija Prodajalec s prikazom lastnosti za element "Slovenija"**

Vir: Avtor

## Računi

V podatkovnem skladišču so poleg rednih računov (faktur) tudi stornirani računi, dobropisi, pro forma računi, izdajnice za lastno porabo. Z dimenzijo računi so različne vrste računov ločene po skupinah.

Dimenzija ni skladna z drugimi podatkovnimi skladišči.

Uporablja se samo v prodajnem podatkovnem skladišču.

Dimenzija Računi ima tri nivoje (Slika 23):

- Oznaka vrste računa.
  - o Opis vrste računa.
- Leto izdaje računa.
- Račun

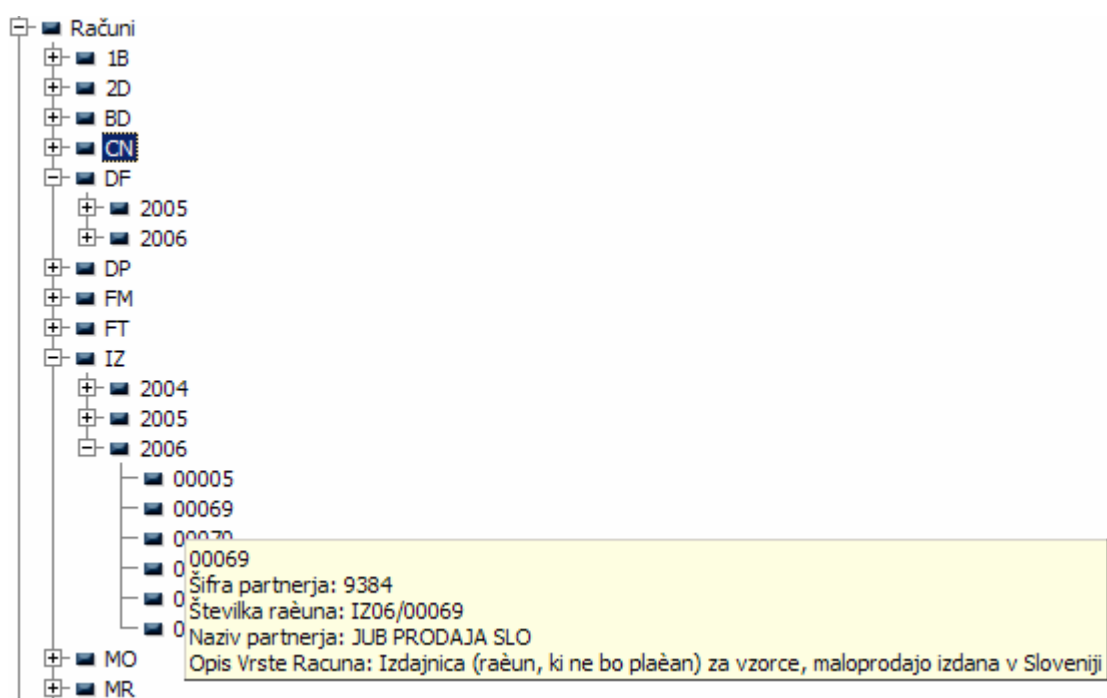
- o Oznaka računa.
- o Šifra partnerja.
- o Naziv partnerja.

Zgornji nivo dimenzije – oznaka vrste računa – se polni ročno ob nastopu nove vrste računa, kar se zgodi redko. Ostali nivoji se polnijo samodejno ob vnosu računov v podatkovno skladišče.

Iz oznake računa uporabnik lahko izve:

- Katera družba je račun izdelala.
- Katera družba je račun izdala.
- Za katero vrsto računa gre (veleprodajni račun, maloprodajni račun, dobropis ...).

**Slika 23: Dimenzija Računi s prikazom lastnosti za element "IZ2006/00069"**



Vir: Avtor.

### 4.3.3 Posebnosti

Sistem za podporo upravljavcem v podjetju je maksimalno prilagojen potrebam uporabnikov. Ker je družba sestavljena iz obvladujočega podjetja in odvisnih podjetij, smo morali s sistemom poskrbeti, da imajo vsi uporabniki podatke, ki jih želijo. Uporabniki pa želijo iste podatke videti na različne načine.

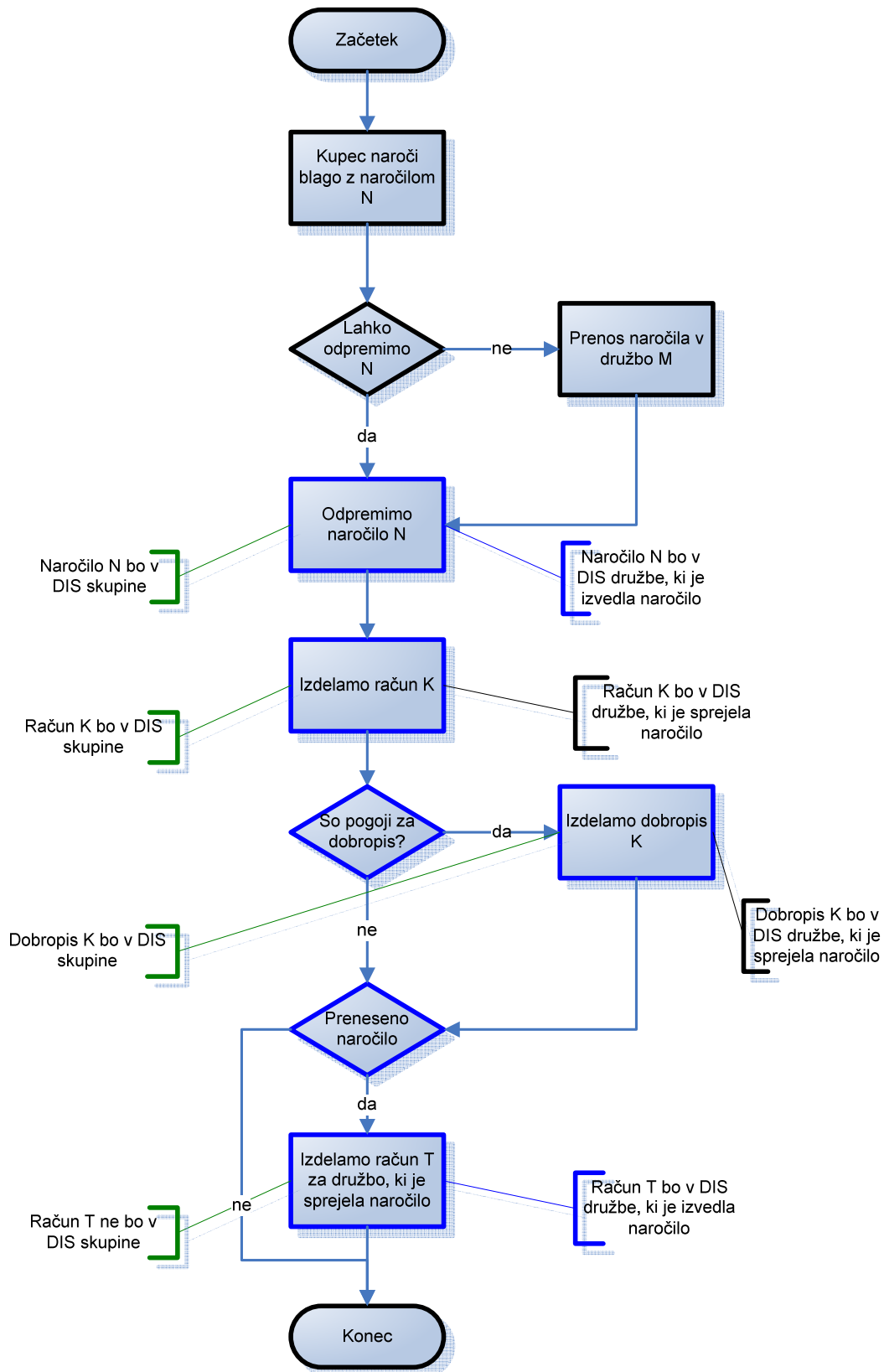
Prodaje blaga iz družbe v skupini v drugo družbo v skupini, uporabnik, ki gleda družbe skupaj kot skupino, vidi kot notranji promet med družbama – zanj to ni »prava« prodaja in je ne želi videti med rezultati prodaje. Uporabnik v prvi družbi, tisti, ki je prodala blago drugi družbi, pa vidi to kot pravo prodajo, saj je bilo blago v resnici prodano. Tako imamo transakcijo, ki jo prvi uporabnik ne želi videti, drugi uporabnik pa.

Prodajo blaga iz katerekoli družbe kupcu, ki ni del skupine, pa želita v svojih podatkih videti oba uporabnika.

Neto vrednost prodaje zmanjšujejo dobropisi – materialni in finančni, ter izdajnice. Dobropisi zmanjšujejo neto vrednost prodaje predvsem pri velikih kupcih, ki so upravičeni do dobropisov zaradi dosežene količine. Ti dobropisi znatno zmanjšajo neto vrednost prodanega blaga in s tem tudi PVS (razlika med prodajno vrednostjo proizvodov (končna tržna cena po lokalnem ceniku) in variabilnimi stroški (variabilni stroški za prodane proizvode glede na izvor proizvodnje – SLO/SČG)). Pri majhnih kupcih, predvsem pri kupcih, s katerimi smo šele začeli poslovati, pa lahko predstavljajo izdajnice velik del prodaje.

Prikaz posla, kjer kupec naroči blago pri družbi, ki nima svojega skladišča, in zato naročilo izpolni druga družba, je na Slika 24.

Slika 24: Prikaz posameznega posla



Vir: Avtor.



Dokumenti, ki nastanejo ob poslu in se vodijo v podatkovnem skladišču (glej Tabela 2):

1. Naročilo N: kupec naroči blago z naročilom N pri družbi D.
2. Račun T: ker družba D nima svojega skladišča, ji blago odpremi družba M in ji za to blago izda račun T.
3. Račun K: družba D izda za naročeno blago račun K kupcu.
4. Dobropis K: ob izpolnjenih pogojih družba D izda dobropis kupcu.

**Tabela 2: Dokumenti, ki nastanejo ob poslu in so v podatkovnem skladišču**

Dok.\pod.skl.	Skupina	Družba D	Družba M
Naročilo N	✓	✗	✓
Račun T	✗	✗	✓
Račun K	✓	✓	✗
Dobropis K	✓	✓	✗

Vir: Avtor.

## 4.4 Učinki gradnje podatkovnega skladišča

### 4.4.1 Združeni podatki

Že sam projekt gradnje podatkovnega skladišča je prinesel nekatere spremembe v organizacijo. Največ jih je bilo seveda v informacijski tehnologiji, nekatere spremembe pa so bile tudi v organizaciji dela.

Največja sprememba v organizaciji dela, je bila zagotovo uvajanje enotnih šifrantov. Posamezni oddelki so za »svoje« potrebe uporabljali lastne šifrante (glej 2.4), za katere so skrbeli sami. Sedaj so šifranti enotni za celotno organizacijo, določene so osebe (skrbniki), ki edine lahko polnijo te šifrante.

Pri šifrantu poslovnih partnerjev je bilo potrebno največje poenotenje. Do uvedbe podatkovnega skladišča so bili poslovni partnerji v različnih oddelkih v različnih šifrantih. Tako smo imeli šifrante za:

- kupce v veleprodaji, ki so se uporabljali tudi za materialno poslovanje,
- kupce v maloprodaji,
- dobavitelje surovin v veleprodaji, ki so se uporabljali tudi za materialno poslovanje,
- dobavitelje pomožnega materiala in rezervnih delov,
- kupce v finančnih aplikacijah,
- dobavitelje v finančnih aplikacijah.

Tako je lahko bil posamezen poslovni partner v šestih različnih šifrantih, ki so bili med seboj povsem nezdržljivi. Veleprodajni šifranti so bili v sistemu VAX, maloprodajni v okolju DOS v bazi dBase, finančne aplikacije so tekle v okolju DOS in Windows in so uporabljali svoj podatkovni format.

Poleg tega so bili nekateri partnerji poslovni partnerji v več različnih družbah (npr. kupci iz Bosne so kupovali neposredno iz matične družbe, iz družbe v Zagrebu in v Sarajevu, lahko pa so bili še dobavitelj za matično družbo).

V novem šifrantu so vsi poslovni partnerji združeni in praviloma nastopajo samo enkrat.

Tudi ostali šifranti so bili polni podvojenih podatkov in uvedba enotnih šifrantov je to močno zmanjšala. Enotni šifranti so tudi zmanjšali možnost napačnih podatkov, ker sedaj vsi uporabljajo iste podatke in se tako napake lažje odkrijejo in hitreje odpravijo.

Seveda pa zahtevajo enotni šifranti večjo odgovornost vnašalcev podatkov. Pred vnosom morajo preveriti, ali niso partnerja že vnesli v drugem oddelku ali družbi. Poleg tega morajo vnašati tudi podatke, ki jih sami ne potrebujejo za svoje delo – npr. v prodaji podatke o transakcijskem računu kupca, ki jih potrebuje nabavni oddelek.

#### **4.4.2 Pridobivanje podatkov**

Do nekaterih osnovnih podatkov, ki so uporabnikom pomagali pri odločanju, so uporabniki lahko prišli zelo hitro tudi pred uvedbo podatkovnega skladišča. Osnovni podatki o prodaji so bili na voljo zelo hitro obliki poročil, ki so se izdelala vsako noč. Tudi v podatkovnem skladišču so podatki od včeraj, tako da tu ni bistvene spremembe. Seveda pa je sedaj možno pridobiti podrobnejše podatke – pri prodaji do nivoja posameznega artikla.

Ko pa je uporabnik želel podatke, ki niso bili del rednih poročil, je stvar povsem drugačna. Na zahtevo uporabnika je moral oddelek IT izdelati poročilo. Če so bili podatki že primerni za poročilo, je bilo treba »samo« izdelati poročilo. Velikokrat pa je bilo treba podatke najprej pripraviti z aplikacijo v cobolu, ki je izračunala zahtevane podatke, in šele potem izdelati poročilo. Če je šlo za poročilo, ki se je večkrat ponavljalo, so ga izdelovali na zahtevo uporabnika ali ob rednih intervalih.

Tudi v programu Commander Desktop so bili uporabniki omejeni v svojem delovanju. Izdelava novih poročil je zahtevala podrobno poznavanje sistema in je bilo za uporabnika prezahtevno. V tem sistemu so nova poročila izdelovali zunanji sodelavci, kar pomeni, da je bilo za izdelavo poročila potrebnega še več časa, seveda pa so ta poročila izvajalci tudi zaračunali.

Čas, ki je bil potreben za izdelavo takega poročila, se je meril v dnevih, za zahtevna poročila pa tudi v tednih.

Večkrat je prihajalo do nesporazumov med uporabnikom in izvajalcem. Poročilo je vsebovalo podatke, ki jih uporabnik ni želel ali pa so bili v taki obliki, da so bili za uporabnika nerazumljivi. Včasih so bile razlike le lepote narave, včasih pa je bilo poročilo za uporabnika povsem neuporabno.

Če povzamem: za izdelavo poročila izdelanega na zahtevo, je moral uporabnik najprej izdelati zahtevo za poročilo (1-3 ure), ki ga je nato oddelek IT realiziral. Za poročila, kjer so bili

podatki že v obliki, ki je omogočala izdelavo poročila, je razvijalec potreboval od 4 do 16 delovnih ur (Vrečar, 2005). Če pa je poročilo vsebovalo podatke, ki niso bili v primerni obliki, se je izdelava poročila podaljšala na delovni teden. Nekaterih podatkov pa sploh ni bilo mogoče pridobiti v želeni obliki. Seveda je bila hitrost rešitve uporabnikovega problema odvisna od trenutne zasedenosti razvijalcev.

Z uvedbo podatkovnega skladišča, ki vsebuje »vse« podatke, ki jih uporabniki potrebujejo ali bi jih lahko potrebovali, je izdelava novih poročil stvar uporabnikov. Uporabniki sami delajo poizvedbe in pridobivajo podatke, ki jih potrebujejo. Seveda je občasno potrebna tudi pomoč oddelka za IT, vendar je to prej izjema kot pravilo.

Hitrost pridobivanja podatkov se meri v minutah. Ker je uporabnik sam tudi izvajalec, ne prihaja do nesporazumov med zahtevami in izvedbo. Napake pa seveda niso izključene in so lahko posledica nepoznavanja pomena podatkov ali napačne uporabe orodij.

Občasno uporabniki še vedno potrebujejo tudi podatke, ki niso zajeti v podatkovnem skladišču. V tem primeru poročilo še vedno pripravi oddelek za IT. Prednost pa je v tem, da so skoraj vsi podatki zajeti v operativni podatkovni shrambi in jih ni treba iskati po različnih virih. V skrajnem primeru pa je treba pridobiti podatke iz operativne baze, vendar so vsaj šifranti enotni, tako da so tudi taka poročila na voljo hitreje, predvsem pa je manj možnosti za napake v poročilih.

Za poročila, ki jih uporabnik sam ne more pridobiti iz podatkovnega skladišča, mora še vedno napisati zahtevo za poročilo (1-3 ure). Razvijalci velikokrat lahko rešijo uporabnikov problem že v okviru podatkov, ki so na voljo v podatkovnem skladišču (1 ura). Če podatki niso na voljo v podatkovnem skladišču, je treba izdelati posebno poročilo. Za preprosta poročila, kjer so podatki na voljo v operativni podatkovni shrambi, to zahteva 1 do 2 uri dela. Izdelava poročila, za katero so podatki na voljo samo v operativnem sistemu, pa zahteva do 3 delovne dneve.

Primerjava časa, potrebnega za izdelavo novega poročila, je v Tabela 3.

**Tabela 3: Čas, potreben za izdelavo novega poročila**

vir podatkov	brez podatkovnega skladišča	s podatkovnim skladiščem
podatki v sistemu za podporo odločanja	5–17 ur	1–4 ure
podatki v primerni obliki	5–17 ur	1–4 ure
podatki v zunanjih virih	teden dni	3 dnevi

Vir: Avtor.

Oddelek za IT sedaj manj sodeluje pri izdelavi poročil in tako prihranjene človeške vire lahko uporabimo za druge projekte. Ocenjujem, da je na mesečni ravni prihranek 80 človek-ur.

Lahko sklenem, da je **pridobivanje podatkov iz podatkovnega skladišča hitrejše** kakor brez njega. Poročila, ki jih izdelajo uporabniki, so njim razumljivejša.

### 4.4.3 Uporaba podatkovnega skladišča

Nenazadnje podatkovno skladišče omogoča dostop do podatkov tudi uporabnikom iz odvisnih družb. Seveda se je dostopnost povečala tudi zaradi razvoja komunikacijskih povezav, vendar to ne bi bilo dovolj.

Za enotna poročila, ki jih bodo uporabljale vse družbe, je potreben centraliziran sistem. Ko gledajo določeno poročilo, vsi uporabniki vedo, kaj pomenijo določeni podatki. To je nujno že na nivoju ene same družbe, ko morajo medsebojno komunicirati posamezni oddelki, še toliko bolj pa je to pomembno pri različnih družbah.

Upravljalci v matični družbi morajo upravljati, predvsem pa nadzorovati tudi delovanje odvisnih družb in pri tem jim je v veliko pomoč podatkovno skladišče. Če bi bili odvisni samo od poročil, ki bi jih pošiljale odvisne družbe, bi bil nadzor mnogo manj kakovosten. Izdelovalci poročil v odvisnih družbah bi lahko pripravili taka poročila, kot bi jih želeli sami. Pri tem niti ni nujno, da bi šlo za prirejanje ali potvarjanje podatkov, dovolj je že, da se slabe informacije skrijejo v kup papirja. Vedno je tudi na voljo izgovor, da je prišlo do nesporazuma pri razumevanju zahtev matične družbe. Pri mednarodnih družbah je primerjanje še težje, ker nastopajo različni jeziki, različne denarne enote, različni knjigovodski standardi ...

Te probleme lahko deloma rešujejo pravilniki o poročanju, vendar pridejo na ta način do matične družbe samo podatki, ki jih odvisna družba želi poslati.

Ko pa so v podatkovnem skladišču podatki, ki so zajeti neposredno iz operativnega sistema odvisne družbe, ne more priti do prirejanja poročil. Razen seveda, če ne pride do prirejanja osnovnih transakcij, kjer gre seveda za zlonamerno dejanje, ki se ga ne da opravičevati z nerazumevanjem.

Upravljalci imajo tako v primeru podatkovnega skladišča ves čas na voljo podatke, ki jih lahko uporabljajo pri upravljanju tako matične družbe kot odvisnih družb.

## 5. Zaključek

Upravljalci lahko danes z informacijsko tehnologijo pridejo do ogromne količine podatkov. Če so bili včasih upravljalci lačni podatkov, so danes dobesedno zasuti z njimi. Ker je podatkov preveč, je potreben sistem, ki omogoča upravljavcu, da uporablja samo tiste podatke, ki jih potrebuje pri upravljanju.

Okolje, v katerem danes delujejo družbe, se ves čas spreminja. Nove okoliščine zahtevajo od upravljavcev, da iščejo in uporabljajo nove podatke. Če bi bili odvisni od klasičnega modela izdelave poročil – izdelava naloga za izdelavo poročila, izdelava poročila v oddelku za IT, preverjanje ustreznosti poročila, uporaba poročila – potem bi prišli do podatkov prepozno.

Rešitev, ki je danes že široko uveljavljena, je uporaba sistema za podporo odločanja, ki uporablja podatkovno skladišče. V podatkovnem skladišču so shranjeni vsi podatki, ki jih upravljavci potrebujejo ali pa bi jih lahko potrebovali.

Uporabniki lahko sami, največkrat brez pomoči strokovnjaka, prihajajo do podatkov, ki jih potrebujejo. Če so pred uvedbo podatkovnega skladišča potrebovali nova poročila, so nanje lahko čakali tudi teden dni. S podatkovnim skladiščem se je čas čakanja skrajšal približno za polovico.

Oddelek za IT je sedaj manj obremenjen z izpolnjevanjem zahtev po novih poročilih. Tako sproščene človeške vire uporabljamo za druge storitve. Ocenjujem, da je na mesečni ravni prihranek 80 človek-ur.

S prihrankom časa uporabnika in s prihrankom virov oddelka za IT lahko sklenem, da so poročila, pridobljena iz sistema za podporo odločanju, cenejša.

Z nalogo sem želel prek teoretičnega spoznavanja podatkovnih skladišč izpeljati projekt gradnje podatkovnega skladišča, ki omogoča kakovostnejše pridobivanje podatkov, pomembnih za sprejemanje odločitev. Uspelo nam je zgraditi sistem, ki omogoča hitrejšo pridobivanje podatkov, prilagojenih potrebam posameznega uporabnika.

V prihodnosti bo treba integrirati v skladišče še podatke iz družb, ki niso vključene v navidezno zasebno omrežje, in omogočiti naprednejše analize podatkov (npr. kaj-če analizo).

Gradnjo podatkovnega skladišča sem vodil kot vodja projekta. Samostojno sem se odločil za arhitekturo podatkovnega skladišča. Izdelal sem rešitev za operativno podatkovno shrambo, pri kateri sem uspel z enkratnim vpisom podatkov zagotoviti podatke za različna podatkovna skladišča (npr. za matično in odvisno družbo). Z enkratnim vpisom podatkov sem dosegel, da se ne morejo pojaviti neskladja med poročili, seveda ob pravilni uporabi podatkov.

S sodelovanjem uporabnikov sem izdelal večino izračunanih meritev in pripravil predloge vnaprej pripravljenih poročil.

Družba, ki jo obravnavam v delu, je gradila in še gradi svoj sistem za podporo odločanja postopoma. Izkušnje, ki smo jih pridobili pri gradnji informacijskega sistema, so pokazale, da so uporabniki najpomembnejši del sistema. Z naglimi, korenitimi in premalo premišljenimi spremembami lahko uporabnike odvrnemo od novih rešitev, ki so sicer tehnično povsem sprejemljive.

Uporabniki potrebujejo čas, da spoznajo prednosti, ki jih prinaša podatkovno skladišče. Ko te prednosti spoznajo, se radi učijo in sodelujejo pri razvoju sistema. In ta sistem bodo uporabniki tudi uporabljali.

Sistem za podporo odločanja, ki se uporablja v družbi, izdeluje poročila, ki jih enako razumejo vsi uporabniki iz različnih oddelkov in iz različnih družb, lociranih v več državah. Poročila so v primerjavi s starejšimi poročili točnejša, saj temeljijo na enotnih šifrantih in podatkih, ki so zajeti neposredno iz operativnih sistemov.

## 6. Slovar kratic, pojmov in izrazov

- Cleansing tools – orodja za čiščenje podatkov
- Common Business Model – skupni poslovni model
- Computer-aided Software Engineering tool (krat. CASE) – orodja za računalniško podprto načrtovanje programske opreme
- Conformed Dimension – skladna dimenzija
- Consistent – dosleden, skladen, združljiv
- Cost/Benefit Analysis – analiza stroškov in koristi
- Cross-functional analysis – medpodročne analize
- Cross-organizational development – vseorganizacijski razvoj
- Data Cleaning – čiščenje podatkov
- Data integrity – neoporečnost podatkov
- Data Mart (krat. DM) – področno podatkovno skladišče
- Data Transformation Services (krat. DTS) – orodje za zajem, čiščenje in preoblikovanje podatkov, vključeno v Analysis server
- Data Warehouse (krat. DW) – podatkovno skladišče
- Dirty Data – nepravilnosti, napake v podatkih
- Extract Transform Load tools (krat. ETL) – orodja za zajem, preoblikovanje in polnjenje podatkov
- Federated Data Warehouse – federativno podatkovno skladišče
- Information overload – informacijska zasičenost
- Integrated – enoten, strnjen; data is integrated – podatki so združeni
- Inteligentni sistemi (ang. Intelligent Systems) – inteligentni sistemi
- Management Support Systems (krat. MSS) – sistemi za podporo upravljanja
- Metadata – opisni podatki
- Multidimensional Expressions (krat. MDX) – jezik za večdimenzijske poizvedbe
- On-Line Transactional Processing Systems (krat. OLTP) – transakcijski ali operativni sistem
- Operational Data Storage (krat. ODS) – operativna podatkovna shramba
- Project Agreement Document – vzpostavitevni dokument
- Relational Database Management System (krat. RDBMS) – sistem za vodenje podatkovnih baz (krat. SUPB)
- Return of Investment (krat. ROI) – donosnost naložbe
- Service Level Agreement – dogovor o storitvi

## 7. Literatura

1. Adelman Sid, Moss Larissa Terpeluk: *Data Warehouse Project Management*. Boston: Addison-Wesley, 2000. 404 str.
2. Anahory Sam, Murray Denis: *Data Warehousing in the Real World*. Harlow: Addison Wesley Longman Limited, 1997. 352 str.
3. Batagelj Katarina (ur.), Hoyer, Štefan (ur.): Spominski zbornik tovarne JUB ob 125 letnici delovanja. Dol pri Ljubljani: JUB d.d., 2001. 146 str.
4. Cabena Peter et al.: *Discovering Data Mining From Concept to Implementation*. Upper Saddle River (New Jearsy): Prentice Hall PTR, 1997. 195 str.
5. Codd Edgar F., Codd Sharon B., Salley C.T.: *Providing OLAP (Online Analytical Processing) to User-Analysts: An IT Mandate*. [URL: [http://dev.hyperion.com/download\\_files/resource\\_library/white\\_papers/providing\\_olap\\_to\\_user\\_analysts.pdf](http://dev.hyperion.com/download_files/resource_library/white_papers/providing_olap_to_user_analysts.pdf)]. 22.2.2004.
6. Cummings Steve: *VBA for Dummies*. Foster City: IDG Books Worldwide, Inc., 1999. 428 str.
7. *Data Warehouse – a searchDatabase definition*. [URL: [http://searchdatabase.techtarget.com/sDefinition/0,,sid13\\_gci211904,00.html](http://searchdatabase.techtarget.com/sDefinition/0,,sid13_gci211904,00.html)]. 21.2.2004.
8. English Larry P.: *Improving Data Warehouse and Business Information Quality*. New York: John Willey & Sons, Inc., 1999. 518. str.
9. Golob Izidor; Welzer, Tatjana: *Arhitekture podatkovnih skladišč*. [URL: [http://www.drustvo-informatika.si/dogodki/arhiv/dsi2001/sekcija\\_a/golob\\_welzer.doc](http://www.drustvo-informatika.si/dogodki/arhiv/dsi2001/sekcija_a/golob_welzer.doc)]. 27.1.2004.
10. Hoyer Matjaž: *Primer razvoja internacionalizacije podjetja v industriji premaznih sredstev*. Diplomsko delo. Ljubljana: Ekonomska fakulteta. 2003. 46 str.
11. Inmon William H.; Imhoff, Claudia; Sousa, Ryan: *Corporate Information Factory*. New York: John Willey & Sons, Inc., 1998. 274 str.
12. Inmon William H.: *Building the Data Warehouse*. New York: John Willey & Sons, Inc., 1993. 298 str.
13. Jarc Darja: *Razvoj podatkovnega skladišča poslovne banke*. Magistrska naloga. Ljubljana: Ekonomska fakulteta, 2002. 83 str.
14. Jindal Rajan, Acharya Abhishek: *Federated Data Warehouse Architecture*. [URL: [http://www.wipro.com/dwlphp/savetragetas.php3?pdf=Federated\\_Data\\_Warehouse\\_Architecture.pdf](http://www.wipro.com/dwlphp/savetragetas.php3?pdf=Federated_Data_Warehouse_Architecture.pdf)]. 19.4.2004.
15. Kimball Ralph et al.: *The Data Warehouse Lifecycle Toolkit*. New York: John Willey & Sons, Inc., 1998. 771 str.
16. Kimball Ralph, Merz Richard: *The Data Webhouse Toolkit Building Web-Enabled Data Warehouse*. New York: John Willey & Sons, Inc., 2000. 401 str.
17. Krsnik Jurij: *Uporaba podatkovnega skladišča na področju prometne varnosti v Ministrstvu za notranje zadeve*. Magistrska naloga. Ljubljana: Ekonomska fakulteta, 2000. 75 str.



- 
18. Lahajnar Sebastian, Rožanec, Alenka: *Izgradnja večdimenzionalnih podatkovnih baz za programske rešitve OLAP*. Zbornik posvetovanja Dnevi slovenske informatike 2000. Ljubljana: Slovensko društvo Informatika, 2000. str. 100-105.
  19. Martinčič Sanda: *Modeliranje sistema za nadzor porabe zdravil*. Magistrska naloga. Ljubljana: Ekonomska fakulteta, 1999. 85 str.
  20. Pendse Nigel: *The OLAP Report: Market share analysis*. [URL: <http://www.olapreport.com/Market.htm>]. 22.2.2004.
  21. Pendse Nigel: *The OLAP Report: What is OLAP?*. [URL: <http://www.olapreport.com/fasmi.htm>]. 21.2.2004.
  22. *ProClarity Analytics Platform 5 Professional*. Boise: ProClarity Corporation, 2002. 172. str
  23. Rensin K. David, Fedorchek Andrew M., Amo William C.: *Microsoft® SQL Server 7 Secrets®*. Foster City: IDG Books Worldwide, Inc., 1999. 714 str.
  24. Rozman Rudi, Kovač Jure; Koletnik, Franc: *Management*. Ljubljana: Gospodarski vestnik, 1993. 312 str.
  25. Simon Alan: *90 Days to the Data Mart*. New York: John Willey & Sons, Inc., 1998. 338 str.
  26. Spofford George: *MDX Solutions With Microsoft® SQL Server Analysis Services*. New York: John Willey & Sons, Inc., 2001. 432 str.
  27. Stanek William R.: *SQL Server™ 7.0 Administrator's Pocket Consultant*. Redmond: Microsoft Press. 1999. 365 str.
  28. Stanek William R.: *Windows NT Server® 4.0 Administrator's Pocket Consultant*. Redmond: Microsoft Press. 1999. 328 str.
  29. Šinigoj Aleksander, Jaklič Jurij: *Direktorski informacijski sistemi vedno bližje direktorjem*. Zbornik posvetovanja Dnevi slovenske informatike 2000. Ljubljana: Slovensko društvo Informatika, 2000 . str. 469-477.
  30. Turban Efraim et al.: *Information Technology for Management, Transforming business in the digital economy, 3<sup>rd</sup> edition*. New York: John Willey & Sons, Inc., 2002. 771 str.
  31. Turban Efraim, Rainer Kelly R.; Potter, Richard E.: *Intruduction to Information Technology*. New York: John Willey & Sons, Inc., 2000. 550 str.
  32. Vaughn William R.: *Hitchhiker's Guide to Visual Basic & SQL Server Sixth Edition*. Redmond: Microsoft Press. 1998. 989 str.
  33. White Colin: *The Federated Data Warehouse*. [URL: [http://dmewview.com/editorial/dmreview/print\\_action.cfm?articleId=1953](http://dmewview.com/editorial/dmreview/print_action.cfm?articleId=1953)]. 19.4.2004.

## 8. Viri

1. 10. Letna konferenca skupine JUB. Dol pri Ljubljani: JUB d.d., 2003.
2. 11. Letna konferenca skupine JUB. Dol pri Ljubljani: JUB d.d., 2004.
3. 12. Letna konferenca skupine JUB. Dol pri Ljubljani: JUB d.d., 2005.
4. 13. Letna konferenca skupine JUB. Dol pri Ljubljani: JUB d.d., 2006.
5. Arh Ervin (član oddelka IT). Osebni intervju: 5.12.2006.
6. Interna statistika podjetja JUB d.d.. Dol pri Ljubljani: JUB d.d., 2006.
7. Microsoftova NT konferenca 2004. [URL: <http://www.ntk2004.microsoft.si/slovenija/ntk2004/default.aspx>]. 14.5.2004
8. Microsoftova NT konferenca 2005. [URL: <http://www.ntk2005.microsoft.si/slovenija/ntk2005/default.aspx>], 20.5.2005.
9. Novak Anton (član oddelka IT). Osebni intervju: 4.10.2005.
10. Slovar informatike. [URL: <http://www.islovar.org/>]. 3.11.2006.
11. Vrečar Milan (član oddelka IT). Osebni intervju: 4.10.2005.
12. Wikipedia [URL: <http://www.wikipedia.org/>]. 3.11.2006.