

# Informacijski sustavi : Planiranje učinaka i analiza zahtjeva

---

Hell, Marko; Mijač, Tea

**Authored book / Autorska knjiga**

*Publication status / Verzija rada:* **Published version / Objavljena verzija rada (izdavačev PDF)**

*Publication year / Godina izdavanja:* **2023**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:124:513060>

*Download date / Datum preuzimanja:* **2025-02-28**

*Repository / Repozitorij:*

[REFST - Repository of Economics faculty in Split](#)





# INFORMACIJSKI SUSTAVI

## Planiranje učinaka i analiza zahtjeva

Marko Hell  
Tea Mijač

Sveučilište u Splitu, Ekonomski fakultet.

**Izdavač:**

Sveučilište u Splitu  
Ekonomski fakultet

**Recenzenti:**

prof. dr. sc. Mario Jadrić  
prof. dr. sc. Mario Spremić

**Lektori:**

Josip Luša  
Katarina Bilonić

**Tehnička urednica i dizajn:**

doc. dr. sc. Tea Mijač

**Autori:**

izv. prof. dr. sc. Marko Hell  
doc. dr. sc. Tea Mijač

**ISBN: 987-953-281-091-2**

©2023. Sveučilište u Splitu, Ekonomski fakultet.

Sveučilište u Splitu  
Ekonomski fakultet  
Cvite Fiskovića 5  
21000 Split, Hrvatska  
<https://www.efst.unist.hr/>

Marko Hell i Tea Mijač

**INFORMACIJSKI SUSTAVI**  
**Planiranje učinaka i analiza zahtjeva**

Znanstvena monografija

Split, 2023

CIP - Katalogizacija u publikaciji  
SVEUČILIŠNA KNJIŽNICA  
USPLITU

UDK 004:65.012(075.8)

HELL, Marko  
Informacijski sustavi : planiranje  
učinaka i analiza zahtjeva / Marko Hell i  
Tea Mijač. - Split : Sveučilište u Splitu,  
Ekonomski fakultet, 2023.

Bibliografija. - Kazalo.

ISBN 987-953-281-091-2

1. Mijač, Tea  
I. Informacijski sustavi -- Planiranje

190720073

## Sadržaj

<b>PREGOVOR</b> .....	<b>III</b>
<b>SADRŽAJ</b> .....	<b>I</b>
<b>1 INFORMACIJSKI SUSTAV KAO PODSUSTAV ORGANIZACIJSKOG SUSTAVA</b> .....	<b>1</b>
1.1 INFORMACIJSKI SUSTAV I INFORMATIČKE TEHNOLOGIJE .....	1
1.2 INFORMACIJSKI TIJEKOVI U ORGANIZACIJSKIM SUSTAVIMA.....	4
1.3 GENETIČKA DEFINICIJA INFORMACIJSKOG SUSTAVA .....	7
1.4 GENETIČKA TAKSONOMIJA INFORMACIJSKIH SUSTAVA .....	8
1.5 ZNAČENJE IS-A ZA ORGANIZACIJSKI SUSTAV .....	11
1.6 DIONICI INFORMATIČKOG RAZVOJA ORGANIZACIJE .....	24
1.7 LITERATURA .....	30
<b>2 PLANIRANJE UČINAKA INFORMATIČKIH TEHNOLOGIJA</b> .....	<b>33</b>
2.1 TEMELJNE POSTAVKE STRATEŠKOG PLANIRANJA .....	33
2.2 ODNOS POSLOVNE, IS I IT STRATEGIJE .....	36
2.3 PREGLED METODA I TEHNIKA STRATEŠKOG PLANIRANJA INFORMACIJSKIH SUSTAVA .....	38
2.4 MJERENJE UČINAKA IS-A/IT-JA PREMA METODI <i>BALANCED-SCORECARD</i> .....	51
2.5 LITERATURA .....	58
<b>3 UPRAVLJANJE PROVEDBOM STRATEGIJE RAZVOJA INFORMACIJSKIH SUSTAVA.....</b>	<b>61</b>
3.1 PLANIRANJE UČINAKA RAZVOJA IS-A NA POSLOVNI SUSTAV .....	61
3.2 MJERENJE UČINAKA STRATEGIJE IS-A/IT-JA NA ZADANE STRATEŠKE CILJEVE.....	72
3.3 OPTIMIZACIJA UČINAKA IS-A/IT-JA PRIMJENOM METODE LINEARNOG PROGRAMIRANJA .....	81
3.4 UPRAVLJANJE KORIŠTENJEM MODELA .....	86
3.5 OPTIMIZACIJA DINAMIKE PROVEDBE STRATEGIJE IS-A/IT-JA PRIMJENOM SUSTAVSKE DINAMIKE	89
3.6 FORMALIZACIJA OPERATIVNE UPOTREBE MATEMATIČKIH MODELA .....	94
3.7 LITERATURA .....	107
<b>4 PRISTUP I METODE RAZVOJA INFORMACIJSKIH SUSTAVA</b> .....	<b>109</b>
4.1 OSNOVNI POJMOVI RAZVOJA IS-A.....	109
4.2 ŽIVOTNI CIKLUS INFORMACIJSKIH SUSTAVA .....	117
4.3 KLASIČNI PRISTUPI RAZVOJU IS-A.....	121
4.4 RACIONALNI UNIFICIRANI PROCES .....	125
4.5 AGILNE METODE .....	127
4.6 METODOLOGIJA DASPIS.....	129
4.7 LITERATURA .....	137
<b>5 UTVRĐIVANJE I MODELIRANJE ZAHTJEVA</b> .....	<b>141</b>
5.1 PRIKUPLJANJE ZAHTJEVA .....	141
5.2 OSNOVE OBJEKTOG PRISTUPA .....	148

5.3	UNIFICIRANI JEZIK MODELIRANJA .....	154
5.4.	IZABRANI ALATI ZA UML MODELIRANJE .....	177
5.5.	LITERATURA .....	182
<b>6</b>	<b>MODELIRANJE POSLOVNE TEHNOLOGIJE .....</b>	<b>183</b>
6.1	PROCESNI POGLED I POSLOVNA TEHNOLOGIJA .....	183
6.2	OSNOVNI KONCEPT MODELIRANJA PRIMJENOM BPMN-A .....	187
6.3	FORMALIZACIJA POSLOVNE TEHNOLOGIJE U SKLADU S NORMOM BPMN 2.0 .....	190
6.4	DETALJNO MODELIRANJE POSLOVNOG PROCESA .....	192
6.5	BPMN MODEL ZRELOSTI ORGANIZACIJE.....	207
6.6	OSNOVNI POJMOVI AUTOMATIZACIJE POSLOVNOG PROCESA .....	211
6.7	LITERATURA .....	217
<b>7</b>	<b>MODELIRANJE PODATAKA.....</b>	<b>219</b>
7.1	MODEL PODATAKA – TEMELJ IS-A .....	219
7.2	MODEL PODATAKA EVA: PRIRODAN POGLED NA STVARNI SVIJET .....	222
7.3	RELACIJSKI MODEL PODATAKA.....	228
7.4	OBJEKTNO ORIJENTIRANI MODEL PODATAKA.....	235
7.5	DIJAGRAM MODELA PODATAKA.....	240
7.6	EVOLUCIJA MODELIRANJA PODATAKA .....	245
7.7	LITERATURA .....	250
	<b>POPIS SLIKA .....</b>	<b>252</b>
	<b>POPIS TABLICA.....</b>	<b>255</b>
	<b>INDEKS POJMOVA.....</b>	<b>256</b>



## Predgovor

Posljednjih dvadesetak godina informacijske znanosti doživjele su veliku ekspanziju primjene u svim područjima poslovanja i svakodnevnog života. Od vremena kad bismo nakon čekanja u redu pred šalterom u banci dobili odgovor: „Znate, računalo mi to ne može napraviti...“, do ponude kredita na mobilnom uređaju koja je na temelju izračuna platežne mogućnosti pojedinca usmjerena samo njemu. Prije bismo za korištenje javnog gradskog prijevoza morali znati red vožnje koji bi bio istaknut kao oglas na svakoj stanici. Koncept otvorenih podataka, tj. u ovom slučaju podataka o pozicijama prijevoznih sredstava u javnom gradskom prijevozu, olakšava pristup informacijama u svrhu planiranja prijevoza bez poznavanja voznog reda. Potrebno je investirati u informacijski sustav koji će javno objavljivati podatke o trenutnim pozicijama gradskog prijevoza, a informatičkim poduzećima prepustiti razvoj aplikacije koju će krajnji korisnik instalirati na svoj (pametni) mobilni uređaj. Na taj način informacija o gradskom prijevozu dobiva na pravovremenosti, aktualnosti, točnosti, preciznosti i pouzdanosti te povećava relevantnost za krajnjeg korisnika koji planira svoj prijevoz. Tako poduzeća javnog gradskog prijevoza mogu poboljšati svoju kvalitetu usluge u cilju podizanja zadovoljstva krajnjih korisnika. Opći podatak o redu vožnje postaje u potpunosti individualiziran jer nam više nije bitno znati dnevni red vožnje za sve nego točno jedan segment u trenutku kada nama je potreban. Pojam *individualizacije* u kontekstu procesa izdvajanja iz nečeg općeg postaje temeljni objekt pojma *standardizacije* kojim se opisuje sistemski opći pristup nekoj posebnoj djelatnosti. Osim opisanog primjera unapređenja u kontekstu efikasnosti, informatičke tehnologije mogu iz korijena promijeniti poslovne modele. Projektiranjem pametnih vozila Ford mijenja svoju misiju od svjetskog proizvođača prijevoznih sredstava u svjetsko prijevozničko poduzeće. Sagledavajući razvoj pametnih vozila, postavilo se dugoročno pitanje kome proizvoditi vozila kad će biti autonomna. U tom kontekstu, bila je potrebna promjena ne samo poslovnog modela nego i smisla postojanja organizacije.

Postoji još niz primjera u kojima informatičke tehnologije mijenjaju ne samo poslovne procese i poslovne modele nego i načine razmišljanja, koji u većini slučajeva moraju biti zasnovani na primjeni informatičkih tehnologija. Međutim, kada, kako i u kojem opsegu uključiti informatičke tehnologije u postojeće poslovne modele? Na koji način, metodološki što konkretnije i dugoročnije, osmisliti učinke informatičkih tehnologija u poslovnim sustavima i podići djelotvornost poslovnih procesa kako bismo preživjeli i uspjeli u najezi novih tehnologija? Upravo ova knjiga istražuje i

analizira osnovne metode i tehnike te prezentira potrebna znanja i vještine koje je potrebno razumjeti kako bi se optimalno iskoristio tehnološki napredak u svrhu ostvarenja željenih poslovnih ciljeva. Vodeći se isključivo poslovnim ciljevima, ne ulazeći u tehnologiju izvođenja informacijskih sustava, u ovoj se knjizi informacijski sustavi promatraju s poslovnog stajališta. Tako se čitatelj upoznaje s mogućnostima procjene doprinosa informacijskih sustava u kontekstu poslovnog sustava i potrebnim analizama zahtjeva koji se pred njega postavljaju.

Premda postoji širok spektar zanimanja koje su područja poslovne informatike i informatike iznjedrila, u posljednje vrijeme su najtraženija zanimanja koja pokrivaju i poslovne i IT vještine. Osvrćući se na prošlost, prevladavalo je svega nekoliko zanimanja i to su uglavnom bili programeri i sistemski inženjeri ili poslovni stručnjaci, odnosno dvije krajnosti. IT poduzeća koja rade na razvoju informacijskih sustava (IS), ne vodeći računa o zahtjevima krajnjih korisnika, često nailaze na probleme pri plasiranju tih proizvoda. Upravo zbog navedenih problema proizašla je potreba za stručnjacima koji predstavljaju sponu IT-ja i poslovne ekonomije, pokrivajući pritom optimalnu kombinaciju znanja i vještina iz IT-ja, ali i poslovnog područja. Sadržaj ove knjige namijenjen je upravo takvom profilu čitatelja. Dodatno, jedno novije istraživanje potvrdilo je kako osnovne digitalne vještine predstavljaju četvrtinu temeljnih znanja za sva zanimanja budućnosti. Naime, prolazimo kroz informatičku revoluciju u kojoj su dostupna različita tehnološka rješenja i potrebno se koncentrirati na njihov doprinos poslovnom sustavu. Kako bi se to procijenilo, prvo je potrebno utvrditi granice informacijskog sustava.

Jesu li *Netflix*, *Facebook*, *Google* i sl. informacijski sustavi? Je li naš mobilni uređaj dio nekog informacijskog sustava? Gdje su granice? Kako bismo odgovorili na to pitanje, prvo je potrebno analizirati pojam IS-a organizacije. Osnovne postavke odgovora na navedena pitanja obrađuju se u **prvom poglavlju ove knjige**. Koristeći sustavski pristup organizaciji, izvedena je uloga IS-a. Tako dolazimo do genetičke definicije IS-a, koja omogućuje definiranje genetičkog taksonomskog prostora. Genetički pristup izvodi ulogu IS-a u organizaciji. Upravo su zato u nastavku poglavlja analizirani modeli procjene i razvoj značenja IS-a za poslovni sustav. Kako bi se uopće moglo govoriti o praktičnoj implementaciji IS-a, u zadnjem odjeljku analizirani su dionici razvoja informacijskog sustava. Time su zaokružene temeljne postavke pristupa razvoju i oblikovanju IS-a.

Kada razmišljamo u učincima IS-a/IT-ja na poslovanje, nameću se pitanja: Kako najbolje iskoristiti IS/IT? Što je to poslovna, a što informatička tehnologija i u kakvoj su vezi? Kako izmjeriti povrat investicije u IS/IT? Postoji još mnogo pitanja koja se

javljaju kad se spomene informatizacija poslovanja. Odgovor na postavljena pitanja možemo dobiti jedino ako planiramo učinke i upravljamo provedbom utvrđenog plana. Međutim, kako dugoročno planirati učinke IS-a/IT-ja ako se tehnologija sve brže mijenja? Zato je potrebno razdvojiti poslovnu, IS/IT i IT strategiju. Upravo o tome govori **drugo poglavlje**. Sistematiziranim pregledom i analizom autori su klasične metode strateškog planiranja stavili u kontekst IS-a/IT-ja.

Ključna postavka za upravljanje provedbom je uspostavljanje mjernog instrumenta koji će omogućiti mjerenje pojedinačnih učinaka IS-a/IT-ja na poslovni sustav. Iz tog je razloga u **trećem poglavlju** prikazan i matematički dokazan algoritam za planiranje, optimizaciju i provedbu strategije razvoja IS-a/IT-ja. Prikaz provedenog istraživanja ukazuje čitatelju na važnost primjene matematike u poslovnom planiranju i integriranju različitih poslovnih aspekata, uključujući i IS/IT. U zadnjem odjeljku ovog poglavlja nalazi se formaliziran prikaz operativne upotrebe razvijenog modela kako bi analizirana materija imala ne samo znanstvenu nego i operativnu vrijednost.

**Četvrto poglavlje** sažima glavne pristupe i metode razvoja informacijskih sustava. Na taj se način predloženi pristup stavlja čitatelju u kontekst šireg pogleda na razvoj IS-a. Metodologijom DASPIS, koja je objašnjena na kraju ovog poglavlja, prikazana je povezanost svih faza životnog ciklusa razvoja IS-a. Jasno su prikazani problemski koraci DASPIS-a te metodâ i tehnikâ za njihovo rješavanje. Prikazom metodologije DASPIS jasno su objedinjene i pozicionirane metode i tehnike dugoročnog planiranja učinaka IS-a/IT-ja obrađene u prvom dijelu knjige te navedene metode i tehnike analize zahtjeva koje će se obrađivati u drugom dijelu knjige. Možemo zaključiti da ovim poglavljem uvodimo čitatelja u područje istraživanja metoda i tehnika analize zahtjeva.

Osnovni je problem kod informatizacije to što je organizacija „živi organizam“ i bilo kakve promjene u odvijanju njezinih procesa ne smiju biti ugrožene tijekom informatizacije. Iz tog je razloga potrebno, nakon što je postavljen plan i mjerni instrument za njegovu provedbu, dublje zaroniti u procese i provesti unapređenja. U drugom dijelu knjige istražuje se i prezentira na koji način to napraviti. Polazeći od klasičnih metoda prikupljanja zahtjeva, autori u **petom poglavlju** uvode čitatelja u logiku objektnog pristupa i grafičkih tehnika za modeliranje zahtjeva. Kako bi se dobila operativna vrijednost, u nastavku poglavlja ukratko su navedeni izabrani alati za objektno modeliranje. Na kraju petog poglavlja objašnjen je primjer korištenja nekih metoda i tehnika modeliranja.

U **šestom poglavlju** obrađuje se i analizira primjenjivost modela BPMN 2.0 koji pruža standardizirani vizualni mehanizam za poslovne procese. Prema prikazanim istraživanjima, notacija BPMN stvorena je kako bi bila razumljiva i poslovnim ljudima i samim programerima, zbog čega joj je posvećeno cijelo poglavlje. Čitatelja se upoznaje s potpunim modelom poslovnih procesa prema normi BPMN 2.0, obuhvaćajući time različite poglede na poslovne procese. Posebno je analizirana i prikazana simbolika za detaljno modeliranje poslovnih procesa. Na taj se način čitatelju omogućuje da krene u istraživanje širokog područja modeliranja poslovnih procesa. Kako bi se shvatila vrijednost procesnog pristupa, u nastavku poglavlja istražen je i prikazan BPMN model zrelosti te osnovni pojmovi automatizacije poslovnih procesa.

Tako smo došli i do posljednjeg, **sedmog poglavlja** knjige, u kojem se obrađuje područje modeliranja podataka. Naime, bez obzira na mogućnosti koje su istražene i prikazane u prethodnom poglavlju, logički ispravno i formalno korektno modeliranje podataka predstavlja nužan preduvjet za razvoj bilo kakvog IS-a. U tom kontekstu, autori su pošli od prirodnog pogleda na sustav analizirajući model podataka EVA. Daljnjim razmatranjima čitatelja se upoznaje s osnovama složenijeg relacijskog i na kraju objektnog modela podataka te načinima prebacivanja jednog modela u drugi. Poglavlje završava praktičnim prikazom cjelovitog pogleda na podatke, koristeći pritom dijagram modela podataka.

Ovom su knjigom obuhvaćena sva najvažnija područja bitna za planiranje i analizu IS-a – od cjelokupnih poslovnih učinaka IS-a do analize i specificiranja detaljnih zahtjeva. Svako od ovih poglavlja moglo bi biti raspisano u nekoliko knjiga, stoga je pravi izazov prilikom pisanja bio, između ostaloga, kako dovoljno duboko, ali ne previše detaljno, istražiti i analizirati informacijske sustave u kontekstu njihovih poslovnih vrijednosti. Zbog toga posebno zahvaljujemo prof. dr. sc. Josipu Brumecu, koji je ponovljenim čitanjima i razgovorima izuzetno doprinio našem načinu razmišljanja, cjelovitosti ovog istraživanja i ukupnoj kvaliteti knjige. Zahvaljujemo i svim kolegama koji su svojim sugestijama i komentarima doprinijeli našem istraživanju i razvoju ovog materijala.

Autori

# 1 Informacijski sustav kao podsustav organizacijskog sustava

Još prije dva desetljeća dva veterana IBM-a izjavila su: „U današnje vrijeme stvari se mijenjaju sve brže. Svakoga dana ljudi se sve više razlikuju jedan od drugog. Potrebe i želje kupaca mijenjaju se brzo. Vrijeme skokovitih promjena kraće je od ciklusa ljudskog vijeka.“

To znači da je stabilnost poslovanja, kao i stabilnost zaposlenja, moguća jedino uz kontinuirano prilagođavanje okolini. Zbog toga je potrebno prikupiti podatke, pravilno odvagnuti informacije te, koristeći stečena znanja i iskustvo, postaviti ciljeve i usmjeriti svoje aktivnosti u svrhu njihove realizacije. Jedino je tako moguće biti pola koraka ispred ostatka konkurencije.

Kako bi se postigla stabilnost poslovanja u složenim sustavima kao što je organizacijski sustav, potrebna je potpuna usklađenost svakog dijela organizacije. Upravo zato informacijski sustavi (IS) zasnovani na modernim informatičkim tehnologijama (IT) danas predstavljaju strateški element uglavnom svake organizacije.

Polazeći od osnovnih svojstava informacije, u prvom odjeljku ovog poglavlja informacijski je sustav prikazan kao neizostavni element modernih organizacija. Koristeći sustavski pristup organizaciji, u drugom odjeljku izvedena je uloga informacijskih sustava. Na toj osnovi, u trećem odjeljku dana je genetička definicija informacijskog sustava. Genetička taksonomija informacijskih sustava opisana je u četvrtom odjeljku. U petom odjeljku izvedene su uloge dionika razvoja informacijskog sustava, dok su u šestom prikazani rizici razvoja informacijskih sustava.

## 1.1 Informacijski sustav i informatičke tehnologije

Definicija informacijskog sustava (Laudon & Laudon, 2020) glasi: „Informacijski sustav predstavlja međusobno povezane komponente koje zajedno prikupljaju, obrađuju, pohranjuju i dijele informacije radi pružanja podrške donošenju odluka, koordinaciji, kontroli, analizi i vizualizaciji informacija u organizaciji.“

Ključna osobina informacije je njezina vrijednost. Iako tu vrijednost nije moguće izravno mjeriti, složiti ćemo se da postoje određeni kriteriji, odnosno osobine informacije na temelju kojih je odabiremo. U skladu s tim kriterijima određujemo njezinu vrijednost, odnosno njezino značenje.

Posebno su zanimljive osobine informacije ovisne o vremenu: pravovremenost i aktualnost. Pravovremenost informacije ovisna je o njezinoj dostupnosti u trenutku donošenja odluke. Aktualnost informacije određuje se s obzirom na trenutak kada je podatak nastao, odnosno vrijeme događaja na koji se odnosi.

S organizacijskog stajališta, veća aktualnost informacije daje više vremena za prilagodbu organizacije u svrhu ostvarivanja njezine misije i postizanja željene vizije. Jednako tako, manja aktualnost informacije skraćuje vrijeme za prilagodbu organizacije novonastalom stanju u području njezina djelovanja. Zbog toga bi svaka organizacija, bilo profitna ili neprofitna, trebala imati mogućnost prikupljanja, pohranjivanja, pretraživanja, obrade i distribucije potrebnih informacija.

Upravo bi informacijski sustav<sup>1</sup>, koji se temelji na načelima novih informatičkih tehnologija, trebao omogućiti pravovremenu opskrbu informacijama – kako iz samog organizacijskog sustava, tako i iz njegove okoline. Automatizacija materijalnih i informacijskih tijekova u skladu s modelom poslovne tehnologije organizacije (engl. *business technology*)<sup>2</sup> trebala bi doprinijeti povećanju fleksibilnosti organizacije u svrhu prilagođavanja potrebnim promjenama.

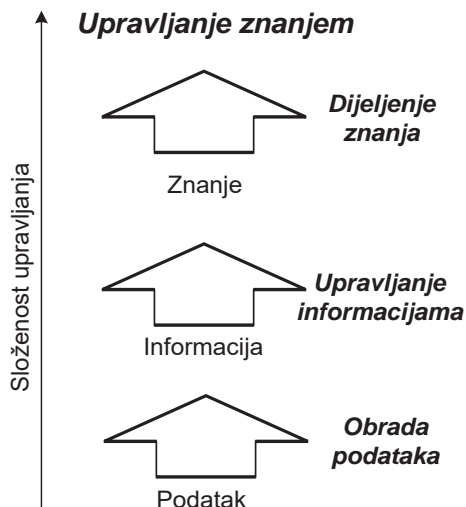
Jedan od osnovnih koraka koji je potrebno poduzeti u modernom poslovanju je kompletna transformacija sadašnjeg, postojećeg stanja u organizaciju koja uči, tj. "učecu organizaciju"<sup>3</sup>. Jednostavan razlog jest činjenica da je danas znanje moć. Učinkovitost učeće organizacije zasniva se na okomitoj sinergiji razina prikazanih na slici 1.1.

---

<sup>1</sup> Prema deskriptivnoj definiciji „sustav koji omogućuje prikupljanje, pohranjivanje, pretraživanje, obradu i distribuciju informacija.“

<sup>2</sup> Poslovna tehnologija podrazumijeva skup svih procesa koji su međusobno povezani materijalnim i informacijskim tijekovima bilo koje vrste, a izvode se radi postizanja misije i ciljeva zbog kojih organizacijski sustav (tvrtka ili ustanova) postoji i djeluje.

<sup>3</sup> Osnovna definicija: „Organizacija koja uči je organizacija koja pojednostavljuje proces učenja svim svojim članovima, i konstantno transformira sebe samu u cilju ostvarenja superiornih performansi.“ (Galić, 2010)



*Slika 1.1 Sinergijske razine učeće organizacije*

U skladu s tumačenjem informacije i podatka<sup>1</sup>, obradom podataka generiraju se informacije. Tako je, primjerice, enormno velika skladišta registratora i drugih oblika papirnate dokumentacije moguće smjestiti na minijaturne memorijske jedinice. Uz pravilnu strukturu arhiviranja, takvi podatci postaju korisni. Obradivanje tako organiziranih podataka otvara mogućnost generiranja informacija velikih vrijednosti. Na taj način IS postaje bitan resurs za generiranje informacija.

Tako generiranim informacijama potrebno je upravljati kako bi se postigla željena kvaliteta odvijanja procesa unutar organizacije. Upravljanjem informacijama postiže se povećanje njihove vrijednosti. Upravljanje zalihama, primjerice, pod uvjetom da nije uspostavljen informacijski sustav upravljanja, najčešće predstavlja generator gubitaka, prostornih te financijskih i drugih resursa. Bez korištenja informatičkih tehnologija njihova optimizacija praktično nije izvediva. Naime, za izračunavanje mnoštava operacija i obradu enormne količine podataka vjerojatno bi trebalo toliko vremena da bi dobivena informacija o potrebnim zalihama postala bezvrijedna.

Istovremeno upravljanje informacijama i ljudskim potencijalom u organizaciji rezultira upravljanjem znanjem, koje je jedna od osnovnih značajki učećih organizacija. Tada ljudski potencijal podrazumijeva sve djelatnike u organizaciji, zajedno s njihovim znanjem, iskustvom i inteligencijom. Ljudski potencijal nužan je za

<sup>1</sup> Informaciju treba tumačiti kao čovjekov spoznajni doživljaj poruke. Za razliku od informacije, podatak je iskaz u nekom jeziku koji odražava stanje nečega. Obradom podataka generiraju se informacije.

pripremu kvalitetne odluke koja se temelji na upravljanju informacijama dobivenih iz informacijskog sustava.

U tom kontekstu informacijski sustav (IS) možemo smatrati podsustavom bilo kojeg dinamičkog sustava, a informatičke tehnologije (IT) jednom od mogućih tehnologija za izvedbu IS-a. Sintagma IS/IT koristi se kako bi se opisalo svojstvo IS-a, tj. da se radi o IS-u koji se temelji na informacijskim i komunikacijskim tehnologijama. Pitanje je samo do koje mjere informacijski sustav i nove informatičke tehnologije mogu doprinijeti povećanju njihove učinkovitosti i efikasnosti.

### 1.2 Informacijski tijekovi u organizacijskim sustavima

Razlog postojanja razvoja i istraživanja IS-a, kao i njegova funkcioniranja, mora se ponajprije utvrditi promatranjem realnog sustava u kojem se on koristi (Brumec, 1997). U ovoj knjizi to su organizacijski sustavi. Prema teoriji sustava i kibernetici, organizacijski je sustav vrlo složen, dinamičan, stohastički, samoorganizirajući, učeći te otvoren s višerazinskom hijerarhijom i povratnim vezama. Svaki organizacijski sustav uključuje ljudske i tehničke resurse, poslovne procese i interakciju s okolinom u svrhu ostvarenja definiranih ciljeva. Upravo na putu ostvarenja ciljeva, a zbog svoje otvorenosti, organizacijski sustavi imaju potrebu prilagođavanja okolini. Reorganizacijom spomenutih čimbenika oni mijenjaju stare i/ili sastavljaju nove vlastite podsustave, tvoreći tako novu organizacijsku strukturu, otporniju na vanjske smetnje pri ostvarivanju željenih ciljeva. Generalno govoreći, cjelokupno funkcioniranje organizacije ostvaruje se izvođenjem velikog broja međusobno povezanih procesa koji se nazivaju poslovni procesi<sup>1</sup> ili poslovna tehnologija (Brumec, 1997). U svakom organizacijskom sustavu sve međusobno povezane aktivnosti i odlučivanja čine skup poslovnih procesa koje je moguće podijeliti u tri razine:

- procesi na razini upravljanja,
- procesi na razini rukovođenja,
- procesi na razini izvođenja.

Ovakva klasifikacija procesa u organizaciji ukazuje i na mogućnost iste raspodjele ciljeva (strateški, taktički i operativni) za čije su ostvarenje potrebni različiti resursi i vrijeme. Izdvajajući zajedničke karakteristike procesa na spomenutim razinama, moguće je uočiti istoimene organizacijske podsustave (upravljanja, rukovođenja i izvođenja) koji se nalaze u međusobnoj interakciji.

---

<sup>1</sup> Poslovni proces je skup povezanih aktivnosti i odluka kojima se postiže parcijalni cilj, a troše neki resursi i vrijeme. (Brumec, 1997)

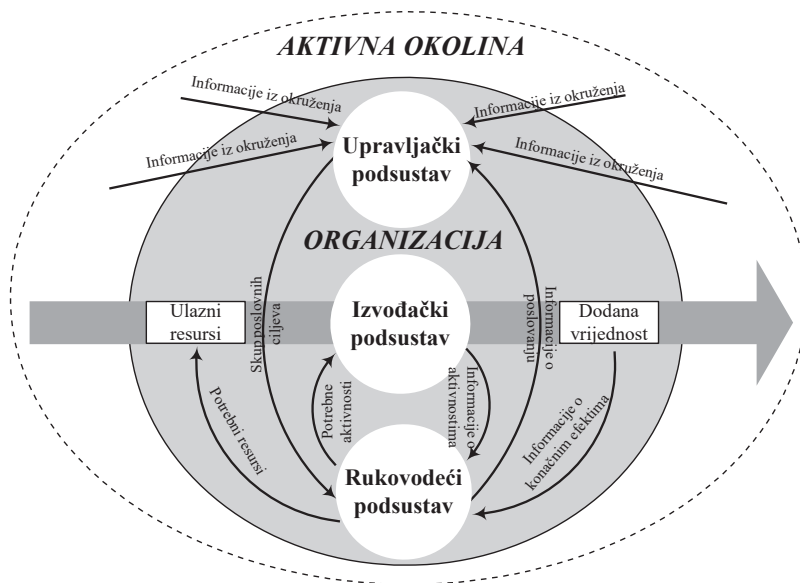


Kroz izvođački podsustav odvijaju se procesi temeljne djelatnosti organizacije. Izvođački podsustav, naime, čini organizacijski sustav prepoznatljivim. Često je to izvana vidljivi tijek cijelog poslovnog sustava, npr. proizvodnja, administriranje, educiranje itd. Kao što sam naziv podsustava upućuje, izvođački podsustav ima funkciju izvođenja ključnih aktivnosti. Sama njegova struktura uvelike će ovisiti o misiji same organizacije. Organiziranim aktivnostima rukovodećeg podsustava, izvođački podsustav proizvodi dodanu vrijednost koristeći ulazne resurse. Struktura njegova IS-a ovisit će o procesima osnovne djelatnosti organizacije. Jedna od uloga IS-a u izvođačkom podsustavu je povećanje produktivnosti rada (Brumec, 1997).

Kako bi izvođački podsustav mogao uspješno ispunjavati svoj cilj, potrebno ga je organizirati, nadzirati i kontrolirati. To znači da sve njegove elemente i potrebne resurse u samom procesu rada treba kombinirati, uskladiti i povezati u jedinstvenu radnu cjelinu. U tu svrhu, rukovodeći podsustav treba raspolagati povratnim informacijama o aktivnostima izvođačke razine, kao i informacijama o njihovim konačnim efektima. S druge strane, rukovođenje se definira kao izvršna aktivnost upravljanja (Brumec & Vrček, 2001; Sikavica, Bebek, Skoko, & Tipurić, 1999). To znači da će rukovodeći podsustav dobiti skup ciljeva od upravljačkog podsustava na osnovi kojih će, uključujući prethodno spomenute povratne informacije, izvršiti organizaciju budućih aktivnosti. Može se reći da je rukovodeći podsustav interpoliran između upravljanja i izvođenja. To znači da on na temelju odluka upravljačkog podsustava i povratnih informacija iz izvođačkog podsustava mora osmisliti način ispunjenja svoje funkcije. Funkcija rukovodećeg podsustava organizacije je organiziranje, kontrola i praćenje uspješnosti procesa u osnovnoj djelatnosti te otklanjanje smetnji uzrokovanih internim ili eksternim čimbenicima. Sukladno funkciji rukovodećeg podsustava, njegov IS mora osigurati pravovremeno opskrbljivanje potrebnim informacijama iz izvođačkog i upravljačkog podsustava, čime se ostvaruje njegova uloga, a to je povećanje učinkovitosti organizacijskog sustava (Brumec & Vrček, 2001; Brumec, 1998).

Svako upravljanje temelji se na donesenim odlukama, a tako se i odlučivanje o poslovnim ciljevima organizacije, koji su potrebni za procese rukovodećeg podsustava, odvija kroz temeljne procese upravljačkog sustava. Donošenjem odluka definiraju se željeni strateški ciljevi koji trebaju dovesti do stabilnosti organizacijskog razvoja. Ispravnost odluka uvelike ovisi o pravodobnim i točnim povratnim informacijama o poslovanju koje se dostavljaju iz rukovodećeg podsustava. Na osnovi njih i odabranih informacija iz aktivne okoline odabiru se najpovoljniji scenariji te se definira skup ciljeva koje rukovodeći podsustav treba provesti u djelo. U skladu s

temeljnom funkcijom upravljačkog podsustava, koja se sastoji od postavljanja poslovnih ciljeva, cilj njegova IS-a je osiguranje stabilnosti rasta i razvoja organizacije (Brumec & Vrček, 2001; Brumec, 1998).



*Slika 1.2 Informacijski tijekovi među organizacijskim podsustavima*

U modelima realnog svijeta materijalne tijekove moguće je opisati informacijama o protoku materije, tj. informacijskim tijekom. Takav pristup složenim sustavima i pojavama (procesu unutar organizacije i njezine interakcije sa okolinom) "virtualizira" elemente i njihove interakcije u sustavu, kao i u njegovoj aktivnoj okolini. Na taj način uzajamno djelovanje elemenata u organizacijskom sustavu i njegovoj aktivnoj okolini svodi se na kretanje informacija među njima. Na slici 1.2 prikazani su osnovni informacijski tijekovi u organizacijskim sustavima. Ne umanjujući odgovornost na određenim razinama, vanjska prepoznatljivost organizacije temelji se na izlaznim rezultatima procesa na razini izvođenja. Zbog toga je na slici 1.2, koja prikazuje informacijske tijekove između organizacijskih podsustava, izvođački podsustav oko kojeg se formiraju ostale razine smješten u centar. Na taj način želi se naglasiti važnost konačnog proizvoda/usluge, putem čega organizacija izvršava svoju misiju. Veliki broj podataka važnih za uspješno djelovanje organizacijskog sustava nalazi se u okolini. Kako bi se u željeno vrijeme mogli iskoristiti, sve te podatke potrebno je objediniti u organizaciji prihvatljivu formu i pohraniti ih. Generiranje informacija iz dobivenih podataka zahtijeva vrijeme i resurse. Nadalje, provođenje donesenih odluka u obliku planiranih akcija, koristeći prethodno opisanu interakciju spomenutih podsustava, također zahtijeva vrijeme i troši resurse. Jednostavno rečeno,

zaprimljene podatke iz okoline ili unutar organizacije potrebno je povezati i u konačnici ih pretvoriti u niz akcija koji bi trebali doprinijeti ostvarenju učinka. Tom prilikom zaprimljeni podatci putuju "informacijskim krvotokom" organizacije, prolazeći pritom kroz spomenute podsustave. Kako bi se optimirala količina resursa i skratilo vrijeme poslovnih procesa, potrebno je za svaki prethodni podsustav izvesti ulogu informatičkih tehnologija.

### 1.3 Genetička definicija informacijskog sustava

Razlog postojanja, razvoja i uloge IS-a moguće je potpuno objasniti jedino ako se promotri realni sustav u kojem on funkcionira. Prema općoj teoriji sustava, svaki skup međusobno povezanih elemenata u svrhu ostvarivanja neke funkcije, tj. skup koji ima strukturu, elemente i funkciju, moguće je promatrati kao sustav. Podsustav nekog sustava definiramo kao skupinu elemenata ili jedan element tog sustava koji može funkcionirati kao sustav nižeg ranga (Radošević, 2001). Na temelju prethodnog tumačenja podsustava izvodi se zaključak da termin *informacijski podsustav* u kontekstu organizacijskog sustava ima isto značenje kao i termin *informacijski sustav*.

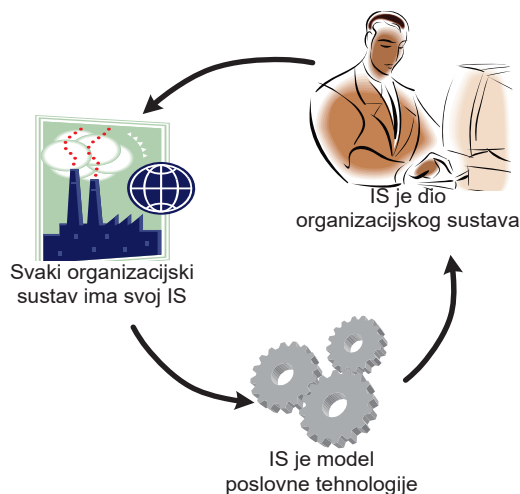
U prethodnom odjeljku analizirane su osnovne vrste procesa u organizaciji. Stavljajući ih u kontekst podsustava organizacije, izvedena je uloga informacijskog sustava.

Iz toga se može zaključiti kako informacijski sustav ne postoji sam za sebe, već je on uvijek podsustav nekog realnog sustava. Svaki realni sustav ima karakteristični informacijski podsustav koji treba omogućiti optimalnu funkcionalnost promatranog realnog sustava. Zbog toga je informacijski sustav uvijek model poslovne tehnologije promatrane organizacije. Jednako tako, svaki organizacijski sustav mora imati svoj informacijski sustav jer u suprotnom nije u mogućnosti funkcionirati, razvijati se i napredovati. To je ujedno razlog zašto razvoj IS-a organizacija najčešće opisuje kao unapređenje postojećeg, a vrlo rijetko kao razvoj novog IS-a.

Na osnovi prethodnog, moguće je istaknuti osnovne značajke odnosa organizacijskog sustava i njegova informacijskog podsustava:

- svaki organizacijski sustav ima svoj informacijski (pod)sustav;
- IS je model poslovne tehnologije nekog organizacijskog sustava;
- IS ne postoji sam za sebe, on je uvijek dio nekog organizacijskog sustava.

Upravo ove tri značajke odnosa organizacijskog sustava i njegova IS-a moraju predstavljati polazište razvojnog procesa IS-a.



*Slika 1.3 Osnovne značajke odnosa organizacijskog sustava i njegova IS-a*

Prema slici 1.3, da bi organizacija bolje djelovala i ispunjavala svrhu svog postojanja, mora je podupirati odgovarajući informacijski sustav. U cilju podupiranja rada organizacijskog sustava na najbolji mogući način, prilikom planiranja i analize IS-a potrebno je izraditi model poslovne tehnologije koji će opisati kako organizacija djeluje. Te su tri značajke u međuovisnosti i nemoguće ih je promatrati odvojeno. Na temelju prethodno iznesenog, moguće je izvesti ulogu IS-a, što nas dovodi do genetičke definicije IS-a.

Na taj način jednoznačno je opisan odnos organizacijskog sustava i njegova informacijskog sustava. Svaki od spomenutih podsustava organizacije zahtijeva različite informacije. Zbog toga funkcije njihovih IS-a sadržane u genetičkoj definiciji ukazuju da se IS može promatrati kao zaseban otvoreni sustav svakog podsustava. Njegovo promatranje kao otvorenog sustava koji komunicira sa svojom okolinom poveznica je s klasičnom definicijom informacijskog sustava organizacije. Razvoj informatičkih tehnologija rezultirao je različitim mogućnostima njihove primjene u širokom spektru organizacijskih i društvenih aktivnosti. Paralelno s različitim primjenom tehnologija razvijali su se i različiti tipovi IS-a. Kao rezultat razvijeni su i različiti tipovi klasifikacija i taksonomija IS-a.

### 1.4 Genetička taksonomija informacijskih sustava

Deskriptivne taksonomije<sup>1</sup> karakterizira fokusiranost na tehnologiju, funkcionalnost i radne karakteristike IS-a. One se zasnivaju na istraživanjima uloge IS-a nakon njihova

---

<sup>1</sup> Taksonomija (grč. *tassein* – „svrstati“, *nomos* – „zakon, znanost“) je znanstvena disciplina koja na

razvoja i korištenja. To znači da polaze od gotovih značajki IS-a. Iako su deskriptivske taksonomije važne za razumijevanje funkcioniranja IS-a u radnom okruženju, nisu dostatne za objašnjenje razloga razvoja i postojanja IS-a te razumijevanje faza životnog ciklusa njihova razvoja (Brumec, 1997).

Genetički pristup razvoju i oblikovanju IS-a izvodi ulogu IS-a u organizaciji. Jedan od kriterija na temelju kojih se može klasificirati IS već je analiziran u prethodnom poglavlju. Taj kriterij polazi od razine na kojoj se proces odvija.

Drugi kriterij je klasificiranje poslovnih procesa s obzirom na predvidljivost slijeda aktivnosti. Sukladno tome, postoje IS-ovi koji podupiru tri osnovna tipa organizacijskih procesa (Brumec, Dušak, & Vrčec, 1998):

- determinirani procesi s predvidivim slijedom su procesi koji su u kontekstu poslovne tehnologije uređenog organizacijskog sustava najčešće jasno definirani i formalizirani;
- determinirani procesi s nepredvidivim slijedom su procesi kod kojih nakon jedne aktivnosti može uslijediti nekoliko drugih aktivnosti;
- procesi inventivnog tipa su procesi koji nisu unaprijed određeni sadržajem, a njihovo pojavljivanje i trajanje nisu predvidljivi.

Ovakva podjela, zajedno s prethodnim razmišljanjem o razinama na kojima se procesi odvijaju, ukazuje na mogućnost dvodimenzionalnog, tj. matričnog prikaza taksonomije IS-a.

Pored razine procesa koju podržava IS i tipova procesa, u obzir je potrebno uzeti i treći kriterij – stupanj zastupljenosti računala u procesima i aktivnostima organizacijskog sustava. To su:

- procesi poduprti klasičnom tehnologijom (tehnologija bez računala);
- procesi poduprti računalnom tehnologijom koja se temelji na podacima (tj. u svrhu sakupljanja, pohranjivanja, obrade i distribucije podataka);
- procesi poduprti računalnom tehnologijom koja se temelji na znanju (tj. računala su nositelji znanja potrebnih za obavljanje vještina koje se zahtijevaju u procesima).

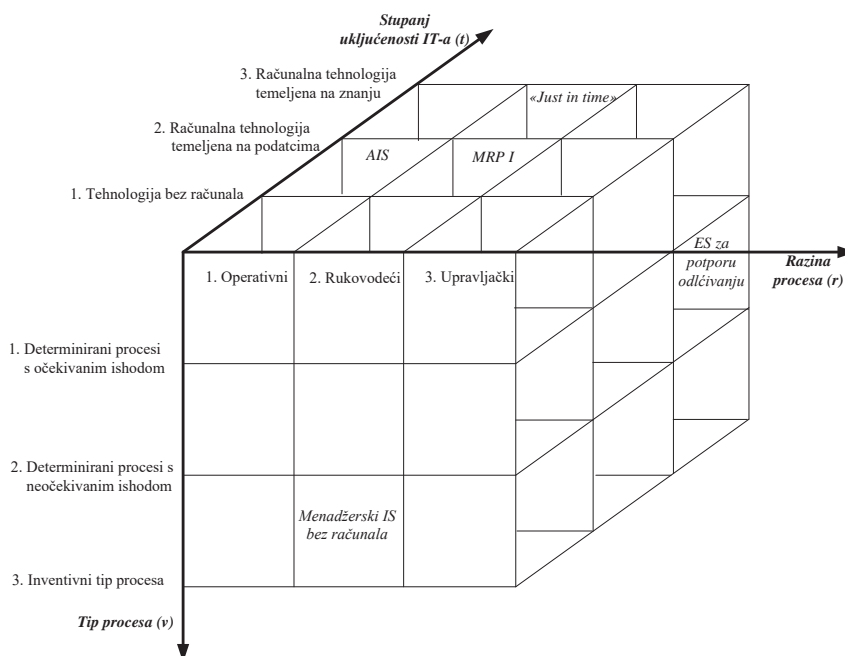
---

temelju sličnosti i razlika taksonomske jedinice kategorizira i razvrstava u skupine (<http://hr.wikipedia.org/wiki/Taksonomija>).

Ovim je kriterijima definiran "genetički taksonomski prostor" informacijskih sustava koji je prikazan na slici 1.4. Svaki od dvadeset i sedam potprostora genetičkog taksonomskog prostora, prikazanog na slici 1.4, predstavlja jednu grupu informacijskih sustava čiji atributi ovise o položaju na osima svake od karakteristika. Iako se sam sustav kategoriziranja (taksonomije) može nadograđivati, ovakva taksonomija primjenjiva je kod strateškog planiranja informacijskih sustava u fazi istraživanja potreba i razvoja novog IS-a. Prostor genetičke taksonomije IS-a trodimenzionalni je prostor s koordinatama:

- vrsta procesa – os  $v$  (tri vrijednosti: determinirani procesi predvidivog slijeda, determinirani procesi nepredvidivog slijeda i procesi inventivnog tipa);
- razina procesa – os  $r$  (tri vrijednosti: izvođenje, upravljanje i odlučivanje);
- stupanj tehnološke opremljenosti – os  $t$  (tri vrijednosti: klasične tehnologije, računala bez baze znanja, računala s bazom znanja).

Svaka od navedene tri varijable (prikazane na koordinatama) ima po tri diskretne vrijednosti na temelju kojih je moguće izračunati *složenost*  $S$  informacijskog sustava koji pripada jednom od dvadeset i sedam potprostora. To znači da je svaki potprostor jednoznačno određen uređenom trojkom  $(v, r, t)$ .



Slika 1.4 Genetički taksonomski prostor (Brumec, Maček & Dušak, 2000)

Složenost informacijskog sustava koji pripada podprostoru  $(v, r, t)$  moguće je izračunati kao mješoviti vektorski produkt u skladu s formulom (1.1)

$$S = (\vec{v} \times \vec{r}) * \vec{t}. \quad (1.1)$$

Formula (1.1) predstavlja volumen kvadra određenog duljinama vektora  $\vec{v}, \vec{r}, \vec{t}$ .

Rezultat složenosti ukazuje na tip IS-a, odnosno genetička taksonomija omogućuje uočavanje i klasifikaciju informacijskog podsustava u kontekstu organizacijskog sustava prije nego što se IS razvio i primijenio, tj. prije nego što su utrošeni raspoloživi resursi organizacije.

Sve prethodno upućuje da oblikovanje i razvoj novog informacijskog sustava organizacije nije moguće prepustiti samo informatičarima. U oblikovanju novog IS-a moraju sudjelovati i stručnjaci koji najbolje poznaju promatrani organizacijski sustav i koji oblikuju njegove buduće funkcije.

## 1.5 Značenje IS-a za organizacijski sustav

U svrhu utvrđivanja mogućih primjena i učinaka tehnologija na razvoj poslovnih sustava danas se provode mnoga teorijska i praktična istraživanja. Pođemo li od pretpostavke da je tehnologija dovoljno razvijena, ostaju dva ključna kriterija:

- isplativost ulaganja u IS/IT i
- sposobnost prihvaćanja tehnologije u organizaciji.

Određeni aspekti prvog pitanja analiziraju se u sljedećim poglavljima. Pojedini elementi koji će utjecati na mogućnost korištenja IS-a/IT-ja te samim tim i ostvarenje njegove primjene nalaze se odgovoru na drugo pitanje. Međutim, ni na ovo pitanje nije moguće dati jednostavan odgovor. U tu su svrhu razvijani različiti teorijski modeli, od kojih će neki biti kratko predstavljeni u ovom odjeljku. Na temelju prikazanih teorijskih modela moguće je uočiti pravilnosti u odnosu organizacijske zrelosti i razvoja IS-a/IT-ja.

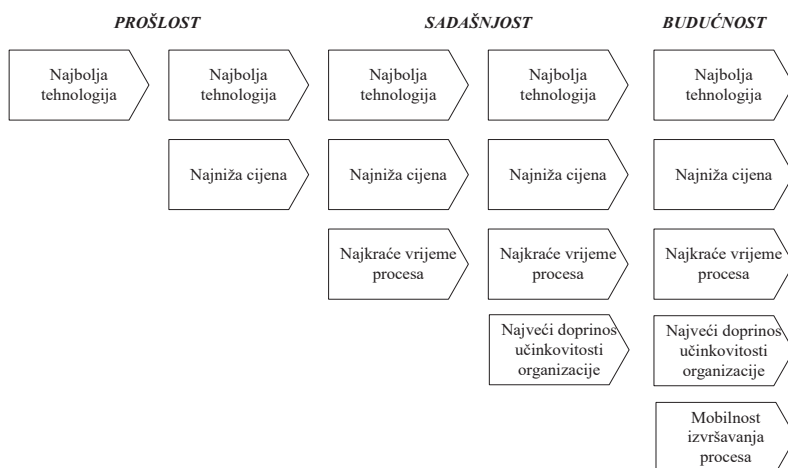
### 1.5.1 Pogled u povijest uloge IS-a/IT-ja

Uloga informacijskih sustava razvijala se postupno, u skladu sa razvojem IS-a/IT-ja, njihovom implementacijom i razinom svijesti o mogućnostima doprinosu djelotvornosti rada i učinkovitosti organizacije. U skladu s tim, razvijali su se i kriteriji za izbor IS-a/IT-ja koje će organizacija koristiti. Tako se u prošlosti zahtijevala najbolja tehnologija kako bi se postigla konkurentna prednost.

Širenje implementacije IS-a/IT-ja u organizacijama doprinijelo je razvoju tržišta IS-a/IT-ja. To je omogućilo kupnju proizvoda IS-a/IT-ja slične namjene od različitih proizvođača, odnosno pojavila se konkurencija. Konkurentna prednost proizvođača IS-a/IT-ja počela se ostvarivati snižavanjem cijena što je rezultiralo uvođenjem cijene kao još jednog kriterija prilikom odabira IS-a/IT-ja. Tako se više nije tražila samo najbolja kvaliteta nego i niska cijena.

Paralelni razvoj ekonomskih teorija te razvoj metoda i tehnika za planiranje IS-a/IT-ja ukazuje na to da je prošlo vrijeme formiranja tržišta IS-a/IT-ja koje se temelji na najnovijoj i najjeftinijoj tehnologiji. Današnja sve šira primjena složenijih ekonomskih metoda i tehnika za planiranje IS-a/IT-ja utječe na razvoj budućih kriterija vezanih za utjecaj IS-a/IT-ja na učinkovitost organizacije. Na taj način organizacija više ne kupuje najnoviji i najjeftiniji IS/IT nego počinje postavljati uvjete prema kojima IS/IT mora biti u potpunosti usklađen s organizacijskim sustavom (ne koristi se pristup *one-size-fits-all* već isključivo individualizirani pristup *business-IT alignment*). Pritom usklađenost podrazumijeva optimiranje strukture organizacije s obzirom na poslovnu strategiju i ciljeve te oblikovanje na način da se anticipiraju potencijali suvremenih IKT-a. Ovakav pristup može se najbolje opisati kao usklađeni razvoj organizacijskog sustava i njegova IS-a.

Vremenski slijed razvoja kriterija IS-a/IT-ja prikazan je na slici 1.5.



**Slika 1.5** Razvoj kriterija za izbor IS-a/IT-ja u okvirima organizacijske strategije

Tržišni zahtjevi već su, između ostaloga, profilirali još jednu zajedničku značajku kojom se IS/IT mora oblikovati kako bi maksimalno skratio organizacijske procese. Ovakav razvoj kriterija za izbor IS-a/IT-ja nije slučajan. Nastao je kao rezultat uzročno-



posljedičnog razvoja IT-ja i postizanja određenog stupnja informatičke zrelosti organizacije. Razvijanjem IT-ja i njegovom implementacijom u organizaciji mijenja se i svijest o mogućnosti njegova utjecaja na učinkovitost organizacije. U tom kontekstu, teško je povući strogu granicu između sadašnjih i budućih kriterija.

Činjenica da će se IS/IT usklađivati s razvojem organizacijske strukture upućuje na to da će budući kriteriji biti definirani smjericama razvoja poslovnih tehnologija. Poslovna tehnologija nije konstanta u vremenu. Zahvaljujući dinamičnom okruženju u kojem promatrani organizacijski sustav djeluje, javlja se i potreba za mijenjanjem poslovnih ciljeva organizacije, što iziskuje promjene poslovne tehnologije. Tako će sadašnja sve raširenija potreba za pokretljivošću čovjeka profiliranjem funkcija organizacije profilirati i nove kriterije izbora IS-a/IT-ja.

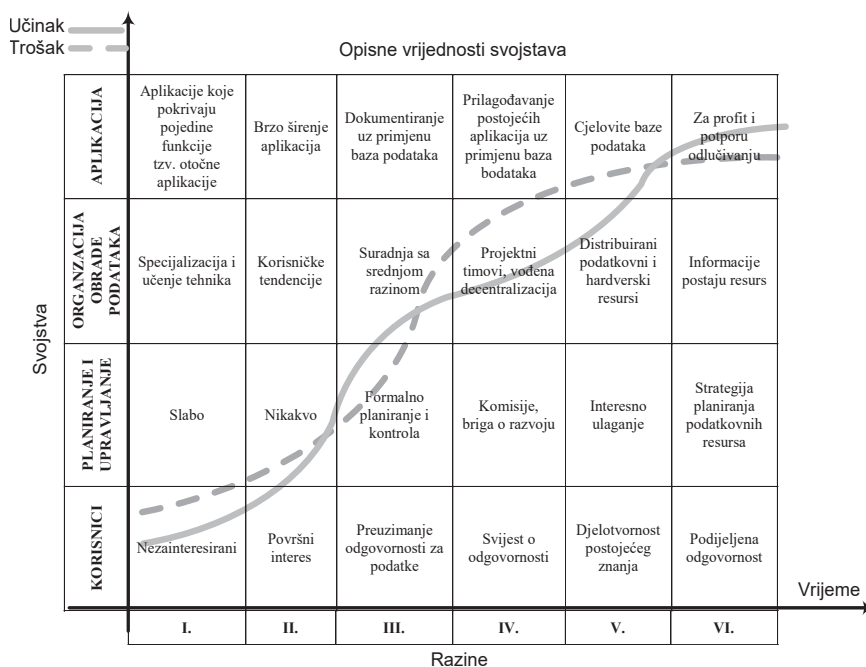
### 1.5.2 Nolanov model zrelosti organizacije

Nolanov model zrelosti organizacije odnosi se na opći pristup korištenju IT-ja u poslovanju. Model je nastao 1974. godine na osnovi empirijskog istraživanja na malom broju organizacija (Wettstein & Kueng, 2002). Rezultati istraživanja pokazali su da sve organizacije koje se koriste informatičkom tehnologijom prolaze kroz slične probleme. Karakteristično je da krivulja učinka i troška u vremenu slijedi S oblik. Utvrđeno je da su sve organizacije prolazile kroz iste faze, koje predstavljaju razine zrelosti organizacije. Zbog toga je promatrana vremenska os podijeljena prvo u četiri razine (razina uvođenja, proširenja, formalizacije i zrelosti), a potom kroz daljnja istraživanja na šest razina zrelosti organizacije:

- I. razina uvođenja,
- II. razina proširenja,
- III. razina upravljanja,
- IV. razina povezivanja,
- V. razina sređivanja,
- VI. razina zrelosti.

Na slici 1.6 prikazan je odnos troška i učinka prema prethodnim fazama.

# 1: Informacijski sustav kao podsustav organizacijskog sustava



**Slika 1.6 Odnos troška i učinka prema Nolanovu modelu razina (Wettstein & Kueng, 2002; Grabavac, 1991)**

Krivulja na slici 1.6 prikazuje troškove ulaganja u IS/IT i učinkovitost organizacije nakon obavljene implementacije prema razinama.

Model pretpostavlja da organizacija na I. razini polako počinje uvođenjem IT-ja u svrhu funkcionalnih aplikacija za reduciranja troškova. Uloženi troškovi održavanja relativno su maleni, a oprema je raspodijeljena u pojedinim odjelima – najčešće u odjelima za računovodstvo i financije. Znanja korisnika su niska, planiranje i kontrola obrade podataka slabi, a znanje o automatiziranoj obradi podataka dosta je specijalizirano.

Razina proširenja prirodni je nastavak prethodne razine u kojoj se značajno proširuje primjena sustava za automatsku obradu podataka. Ovakav razvoj posljedica je uviđanja korisnosti IT-ja, stoga je normalno da dolazi do rasta troškova ulaganja i održavanja. Karakteristično je da se javlja površni interes korisnika, kao i potreba za zapošljavanjem IT stručnjaka te profiliranjem softverskog portfelja.

Na trećoj razini u organizaciji znatno rastu troškovi ulaganja i održavanja IT-ja, stoga se javlja potreba za formalizacijom planiranja i kontrole. Entuzijizam se svodi na normalne okvire, što dovodi do preuzimanja korisničke odgovornosti. Organizacija

obrade podataka diže se na razinu srednjeg menadžmenta. Dokumentiraju se i restrukturiraju postojeći softveri, a menadžment nastoji usmjeriti razvoj u smjeru podrške odlučivanju.

Sam naziv IV. razine ukazuje da je osnovna karakteristika integracija parcijalno razvijenih zasebnih aplikacija. Uspostavljaju se projektni timovi i sustavi za upravljanjem korisnošću informatičkih resursa. Ova razina predstavlja začetak procesa razvoja informacijskog sustava u organizaciji.

Na petoj razini sređivanja dolazi do radikalnih promjena pristupa i pogleda na razvoj IS-a. Podatci se shvaćaju kao jedan od važnih resursa organizacije koji mora stajati na raspolaganju svim korisnicima. Javlja se svijest kako je podacima potrebno odgovorno i svrsishodno upravljati. Zbog toga dolazi do potrebe angažiranja administratora baza podataka. Drugim riječima, raspoložive informatičke tehnologije promatraju se kao integralna cjelina. Djelotvornost postojećeg znanja omogućuje korištenje različitih softverskih platformi s klijentsko-poslužničkom arhitekturom (engl. *client-server architecture*). Paralelno se razvijaju sustavi za potporu odlučivanju, ekspertni sustavi, kao i mnoge druge učinkovite primjene IT-ja.

Prema Nolanu, posljednja razina je razina zrelosti. Informacije postaju resurs, a IS postiže strateško značenje. Razvoj primjene IT-ja okrenut je profitu i potpori odlučivanju.

Neke od kritika kojima je Nolanov model izložen jesu:

- tehnologija se smatra glavnom pokretačkom snagom razvoja u skladu s predloženim razinama;
- krivulja troška svojstvena je svim organizacijama koje implementiraju IT;
- postoje jasno određeni organizacijski ciljevi koji se IT-jem trebaju postići;
- organizacija se promatra u cjelini, tj. prema modelu nije moguće da jedan dio organizacije bude na različitoj razini itd.

Prema Kingu i Kraemeru, Nolanov model razvojni je model koji definira smjer promjena. Nolanov model razina koristan je konceptualan model općeg smjera promjena, stoga je pogodan za bolje razumijevanje trendova utjecaja IS-a/IT-ja na učinkovitost organizacije u cjelini (Grabavac, 1991). Drugim riječima, Nolanovim modelom može se ocijeniti ukupna spremnost organizacije za prihvaćanje IT-ja, dok je za pojedinačno razmatranje funkcija organizacije, tj. pojedinačno promatranje

njezinih procesa potrebno koristiti druge metode i tehnike o kojima će biti više riječi u sljedećim poglavljima.

### 1.5.3 Earlov model

Za razliku od Nolanova modela, Earlov model koncentrira se na faze kroz koje organizacija prolazi prilikom planiranja IS-a. Model je prvi put opisan 1983. Earlov model daje okvir za pravilan izbor učinkovitih metoda i tehnika za oblikovanje IS-a/IT-ja, kao i spoznaju o korisnosti odabrane tehnologije. Earlov model za planiranje IS-a/IT-ja prikazan je u tablici 1.1.

Tablica 1.1 Earlov model planiranja IS-a/IT-ja (Earl, 1989)

	Faza 1.	Faza 2.	Faza 3.	Faza 4.	Faza 5.
<b>Glavni posao</b>	Projektiranje aplikacija	Definiranje poslovnih potreba	Detaljno planiranje IS-a	Procjena strateške prednosti	Veza sa strategijom poslovanja
<b>Ključni ciljevi</b>	Pridobivanje posloводства	Usuglašavanje prioriteta	Usklađivanje aplikacija	Postizanje poslovnih učinaka	Integracija IS-a i poslovnih strategija
<b>Inicijator planiranja</b>	Razvoj informatičke tehnologije	Više posloводства	Korisnici i IT zajedno	Posloводства i korisnici	Usuglašeno: korisnici, posloводства i IT
<b>Pristup planiranju</b>	Razvoj "Bottom-up"	Analiza "Top-down"	Uravnoteženo "Top-down" "Bottom-up"	Poduzetničko-korisničke inovacije	Više metoda istovremeno
<b>Opća značajka</b>	Planiranje vođeno tehnologijom	Planiranje vođeno metodama	Administrativno planiranje	Planiranje vođeno poslovima	Planiranje vođeno organizacijom

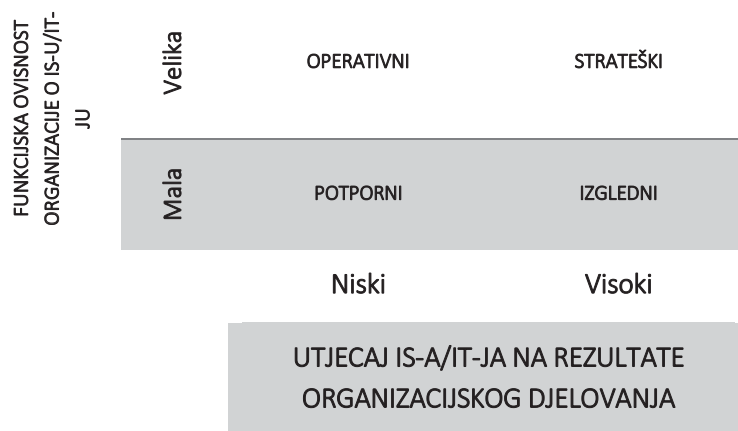
Obratimo li pozornost na inicijatore planiranja i usporedimo ih s ostalim značajkama, moguće je primijetiti da razvoj svijesti o ulozi IS-a/IT-ja, ne zanemarujući ostale čimbenike, doprinosi kvalitetnijem pristupanju problemu. Tako je, primjerice, strateško planiranje IS-a/IT-ja, okarakterizirano petom fazom u tablici 1.1, potrebno provoditi u organizacijama koje imaju jasnu viziju korištenja informatičkih tehnologija u svrhu ostvarivanja strateških ciljeva. Kako bismo povezali faze planiranja sa značenjem, u daljnjem tekstu prikazana je kvalifikacija značenja IS-a za organizaciju. Koristeći misaoni predložak analize BCG (engl. *Boston Consulting Group*), procjenu

važnosti utjecaja informacijskog sustava na učinkovitost organizacije moguće je promatrati kroz dva faktora (Henderson, 1970; Brumec, 1997):

- a) Utjecaj IS-a/IT-ja na rezultate organizacijskog djelovanja, tj. do kojeg stupnja IS i IT kao potpora razvijenom IS-u doprinosi poboljšanju poslovnih rezultata. Mjerenje ovakvih poboljšanja moguće je ostvariti klasičnim ekonomskim indikatorima.
- b) Funkcijska ovisnost organizacije o IS-u/IT-ju, tj. koliki je stupanj ovisnosti operativnih funkcija organizacije.

Na osnovi postavljenih kriterija generalizirane su četiri kategorije značenja IS-a/IT-ja za organizacije, koje su prikazane na slici 1.7.

IS/IT ima potporno značenje u organizacijama u kojima je IS/IT koristan, ali nema kritičnu ulogu na trenutačni i budući poslovni uspjeh organizacije. Potporno značenje IS-a/IT-ja može biti: (1) praćenje radnog vremena, (2) kontrola prihoda i rashoda, (3) financijsko i robno knjigovodstvo, (4) obračun plaća itd. Klasični primjer ovakve organizacije je klasično razvijeno trgovačko poduzeće. Kod organizacija u kojima IS/IT ima potporno značenje preporučuje se ulaganje u nove IT-jeve tek onda kada budu ispunjeni svi ostali zahtjevi za investiranjem.



Slika 1.7 Procjena važnosti IS-a/IT-ja za organizaciju (Brumec, Maček & Dušak, 2000)

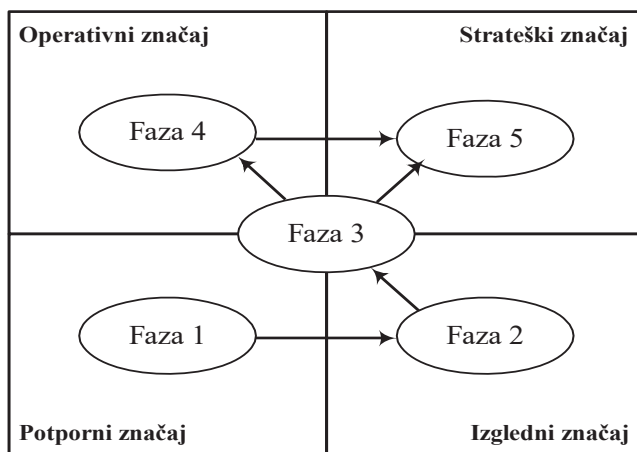
IS/IT ima operativno značenje kod organizacija u kojima je nužan za uspjeh tekućeg poslovanja na izvršnoj razini te gdje se pod izvršnom razinom podrazumijevaju procesi temeljne djelatnosti organizacije. Primjeri operativnih poslova kroz koje se manifestira operativno značenje IS-a/IT-ja jesu: upravljanje zalihama i radnim nalogima, CAD (engl. *computer-aided design*) za proizvodnju, baze

podataka o radnicima, kupcima i dobavljačima, svi trenutačni oblici bankovnog poslovanja itd. Klasičan primjer ovakve organizacije je moderno razvijeno poduzeće, gdje bez primjene računalom poduprtih tehnologija nije moguće odgovoriti na izazove tržišta.

Izgledno značenje IS/IT ima kod organizacija koje bi njegovim uvođenjem mogle poboljšati svoju djelatnost. Pridavanjem većeg značenja IS-u/IT-ju moguće je ostvariti znatniju konkurentsku prednost. Primjeri poslova koje je moguće poboljšati uvođenjem IS-a/IT-ja jesu: (1) planiranje radne snage i njihova usavršavanja, (2) obrada poslovnih dokumenata, (3) internetsko i mobilno bankarstvo, (4) poslovi u kojima je moguće ubrzati i podignuti kvalitetu primjenom sustava za potporu odlučivanju itd. Klasičan su primjer osiguravajuće tvrtke koje prilikom sklapanja osiguravajućih polica koriste ekspertne sustave za određivanja rizika osiguranja i na taj način utječu na postizanje konkurentске prednosti.

Strateško značenje IS/IT ima u organizacijama u kojima strateško planiranje organizacije ovisi o IS-u, a samim tim i o IT-ju koji je njegova potpora. Neinvestiranje u IS/IT u ovom slučaju dovodi u pitanje budućnost cjelokupne organizacije. Primjeri ovakvih poslova jesu: (1) razne tržišne analize, (2) predviđanje strateškog razvoja, (3) komunikacijske veze, (4) investicijski fondovi i proračuni rizika ulaganja itd. Klasični primjer organizacije u kojoj IS/IT ima strateško značenje su aviokompanije. Stavljajući razvoj IS-a/IT-ja prema fazama u kontekst značenja IS-a/IT-ja za poboljšanje organizacije, moguće je dobiti smjernice o načinu planiranja IS-a/IT-ja. U skladu s prethodnim, Earl uočava moguće prelaskе iz niže u više faze, upućujući pritom koje je faze moguće preskočiti, a koje ne. Prema Earlovu modelu, prilikom razvoja IS-a/IT-ja ne bi trebalo biti povratka na niže faze. Ovakav model prikazan je na slici 1.8.

Nolanov i Earlov model razvoja uključuju razinu informatičke svijesti o mogućnostima utjecaja IS-a/IT-ja na učinkovitost organizacije. Poopćavanjem ovih modela moguće je izvoditi zaključke o dinamici razvoja utjecaja IS-a/IT-ja na učinkovitost organizacije.



Slika 1.8 Faze planiranja i značenja IS-a (Earl, 1989)

#### 1.5.4 Čimbenici razvoja informacijskih sustava

Postoji niz razloga koji utječu na organizacijske promjene, pokretanje plana promjena i na razvoj informacijskog sustava. Oni mogu biti unutarnji ili vanjski. Vanjski razlozi tiču se promjena stanja na tržištu, tehnoloških promjena, političkih i pravnih promjena, promjena raspoloživosti različitih faktora itd. Unutarnji čimbenici tiču se promjena organizacijskih ciljeva, promjena organizacijske klime i kulture, promjene organizacijske strukture i sl. Neki od najznačajnijih razloga i čimbenika za poduzimanje plana promjena u razvojnom ciklusu IS-a jesu:

- problemi s postojećim informacijskim sustavom;
- razvoj informatičke tehnologije;
- želja za iskorištavanjem novih prilika;
- rastuća konkurencija;
- promjene faktora eksternog okruženja;
- fuzija, razdvajanje i reorganizacija organizacije;
- rast i razvoj organizacije;
- potreba i zahtjevi da se podaci, informacije i znanje efektivnije i efikasnije koriste.

Neki od faktora koji imaju utjecaj na uspješan razvoj informacijskog sustava jesu:

- podrška uprave;
- jasno definirani ciljevi i funkcije;
- korištenje valjane metodologije za razvoj;

- jednostavan, odgovarajući i suvremen dizajn arhitekture sustava;
- uključivanje korisnika u sve faze razvoja;
- dobro isplaniran i proveden program treninga;
- efikasan plan i realizacija implementacije sustava;
- dobro definiran i primijenjen program održavanja.

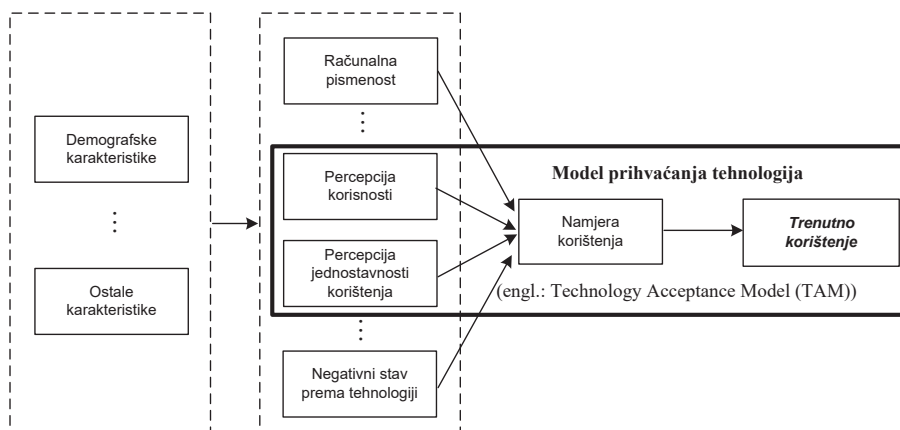
Razvoj informacijskog sustava dugoročan je proces koji se sastoji od više faza, o čemu će više biti prikazano u sljedećim poglavljima.

### 1.5.5 Model prihvaćanja tehnologija

Jedan od osnovnih modela na kojima su zasnovana mnoga istraživanja na temu kako korisnici prihvaćanju tehnologije jest model prihvaćanja tehnologija (engl. *technology acceptance model*) ili skraćeno TAM. TAM je prvi put postavio Davis 1989. (Davis F. D., 1989).

Percepcija jednostavnosti korištenja definirana je kao „stupanj do kojeg korisnik očekuje da ciljani sustav ne predstavlja nikakav napor“, dok je percepcija korisnosti korisnikova „subjektivna vjerojatnost da će služeći se određenim sustavom aplikacija poboljšati svoj rad unutar organizacijskog konteksta" (Khosrow-Pour, 2003).

TAM polazi od hipoteza da namjera korištenja određuje razinu stvarnog korištenja, a percepcija korisnosti i jednostavnosti korištenja određuje namjeru korištenja. Percepcija jednostavnosti korištenja određuje percepciju korištenja jer korisnici smatraju sustav korisnim ako nije potrebno uložiti veliki napor za njegovo korištenje. Prethodne tvrdnje do danas su potvrđene mnogim provedenim istraživanjima. Prošireni model TAM prikazan je na slici 1.9.



Slika 1.9 Prošireni model TAM



Model se također dobro ponaša empirijski. Koristeći analizu regresije mogu su odrediti odnosi varijabli unutar TAM modela te se može predvidjeti korištenje sustava (Davis, Bagozzi, & Warshaw, 1989; Davis F. D., 1989). Osnovni model moguće je nadograđivati i primjenjivati na različita područja primjene IT-ja. Nadogradnja modela odvija se u smjeru određivanja faktora/varijabli koji utječu na percepciju korisnika ili na namjeru njihova ponašanja. Tako je, primjerice, ovaj model korišten za ispitivanje prihvaćanja tehnologija pametnih gradova (Almansour, Beloff, & White 2023), e-obrazovanja (Mustafa & Ali, 2023), telekomunikacijskih tehnologija i mnogim drugim IT područjima (Hsiu-Li & Hsi-Peng, 2008; Chih-Chien, Shao-Kang & Wenchang, 2008). Primjer jednog takvog istraživanja s prikazom provedenog istraživanja moguće je vidjeti u radu S. Brumeca (2008). Tako postavljeni teorijski okviri testiraju se i dokazuju primjenom statističkih metoda. Svaki od njih predstavljat će jedan dio slagalice u širokoj slici modela prihvaćanja informatičkih tehnologija.

Jedna je od karakteristika TAM-a što pretpostavlja da korištenje ovisi o volji, što znači da ne postoje nikakve zapreke da pojedinac koristi IT ako se na to odluči. Međutim, ispitivanja su pokazala kako pristup pojedinca resursima ipak utječe na korištenje (Mathieson, Peacock, & Chin, 2001; Taylor & Todd, 1995). U tom smislu, Mathieson, Peacock, & Chin (2001) proširili su model TAM-a tako da obuhvati utjecaj uočenih resursa na korištenje IT-ja. Element uočenih resursa definiran je kao „stupanj do kojeg pojedinac vjeruje da ima osobne i organizacijske resurse potrebne za korištenje IT-ja“ pa ga treba konceptualizirati ne kao atribut same tehnologije nego i pojedinčeve okoline (Mathieson, Peacock, & Chin, 2001). Uistinu, njihovi su zaključci potvrdili da su uočeni korisnikovi resursi vrijedan doprinos modelu jer je zaključeno da sljedeće kategorije kojima se mjere uočeni resursi bitno utječu na uočenu lakoću korištenja, uočenu korisnost i primjenu IT-ja:

- atributi korisnika, karakteristike korisnika IT-ja koje odražavaju aspekte korisnikove ekspertize sa zadatcima i sustavima, kao i druge aspekte u vezi s korisnikovom situacijom i prirodom zadatka;
- podrška IT osoblja neke organizacije;
- atributi sustava, karakteristike samog sustava, uključujući dostupnost i trošak pristupa.

### 1.5.6 Utjecaj organizacijskih promjena na razvoj IS-a

Bilo je puno pokušaja da se teorijski objasni razlog nastajanja promjena u organizaciji te u tom kontekstu i njezina IS-a, odnosno zašto organizacija poduzima planske promjene u cjelini i/ili u njezinim relevantnim segmentima.

Kurt Lewin postavio je i razjasnio jednu od najstarijih teorija promjena sa stajališta topološke psihologije, izgrađene po uzoru na teoriju polja u fizici, prema kojoj su položaj i kretanje nekog objekta u prostoru određeni silama koje djeluju na taj objekt. Prema ovoj teoriji, promjene nastaju kada su sile koje djeluju u jednom smjeru jače od sila koje djeluju u suprotnom smjeru. Ako su sile koje vode promjenama informacijskog sustava jednako snažne kao i sile koje sprečavaju te promjene, onda je uspostavljeno stanje ravnoteže i ne dolazi do promjena sustava. Promjene i stanje ravnoteže unutar organizacije i njezina informacijskog sustava definirane su kao rezultanta sila koje djeluju u različitim smjerovima.

Drugo objašnjenje promjena IS-a organizacije utemeljeno je na pretpostavci da promjene nastaju kao posljedica dobivanja novih informacija. Smatra se da stjecanje novih informacija o okruženju čovjeka i organizacije ili novih informacija koje čovjek doznaje o tehnološkim promjenama (nove informatičke tehnologije) i o sebi samom potiču promjene u njegovu ponašanju. Pretpostavlja se da je čovjek racionalno biće i da je njegovo ponašanje kao individue i člana organizacije određeno njegovim interesima. Iz tog razloga se očekuje da će on pokretati i prihvaćati predložene organizacijske promjene ako su one racionalno opravdane i ako se može pokazati da će iz predloženih promjena proisteći dobit, npr. prihvaćanje neusporedivo djelotvornijeg i učinkovitijeg IS-a organizacije.

Teorija potkrepljenja također je jedna od teorija kojom se mogu objasniti procesi nastanka organizacijskih promjena. Smatra se da se promjene informacijskog sustava mogu objasniti time da pojedinac ili grupa čine ono za što dobivaju podršku. Ljudi predlažu, prihvaćaju i teže promjenama brže ako: (1) dobivaju više ohrabrenja, (2) dobivaju više poticaja u obliku nagrada i priznanja, (3) imaju osobno zadovoljstvo za pokretanje i prihvaćanje promjena te (4) ako se primjenjuje neka vrsta sankcija za nepoduzimanje i suprotstavljanje promjenama.

Jedna od teorija promjena bazira se na posebnom značenju koje se pridaje utjecaju društvenih normi, najviše onih normi koje vrijede na razini grupe. Promjene nastaju kada članovi grupe nauče nove normativne orijentacije, steknu nova saznanja (primjerice, spoznaju bit i korisnost inovacija o informatičkim tehnologijama), što rezultira novim stavovima, novim vrijednostima i novim odnosima među ljudima. Pritisak koji vrši grupa i utjecaj grupnih normi daju vrlo jak impuls za organizacijske promjene, stoga se smatra da su grupne rasprave najčešće primjenjivane metode u pripremanju i provođenju planova promjena IS-a.

Organizacijske promjene, kao što je razvoj informacijskog sustava, prate otpori jer većina ljudi u organizaciji ne voli promjene, čak i ako su one korisne za organizaciju, grupe i pojedince. Najčešće otpori kod ljudi i grupa nastaju zbog različitih uzroka, izvora i motiva, kao što su:

- nerazumijevanje svrhe, mehanizma i posljedica promjena;
- neuviđanje potrebe za promjenama;
- strah od nepoznatog;
- strah od gubljenja statusa, moći i sigurnosti;
- izbjegavanje napora koje treba uložiti u novo učenje koje se nalaže promjenama;
- nepovjerenje u menadžment;
- narušavanje ustrojenih komunikacija i druženja u radu;
- ustaljene navike;
- postojeće grupne norme;
- remećenje strukture moći;
- loše iskustvo s prethodnim pokušajima promjena.

Kada je otpor promjenama evidentan, spoznat i ozbiljno analiziran, postavlja se pitanje koje se akcije mogu poduzeti da bi otpori bili djelotvorno i učinkovito prevladani. Proces promjena podrazumijeva odmrzavanje *statusa quo*, kretanje k novom promijenjenom stanju i zamrzavanje stanja nastalog promjenom kako bi se ono učinilo permanentnim do sljedeće promjene, odnosno do novog investicijskog ciklusa IS-a. *Status quo* smatra se stanjem ravnoteže. Da bi se promijenilo ovo stanje, odnosno da bi se savladali individualni otpori i grupni konformizam<sup>1</sup>, neophodno je provesti proces nazvan *odmrzavanje*. Udaljavanje od stanja ravnoteže događa se pod djelovanjem pokretačkih snaga koje usmjeravaju ponašanje dalje od *statusa quo* i djelovanja ograničavajućih snaga koje nastoje spriječiti kretanje, odnosno udaljavanje od postojeće ravnoteže. Pokretačke snage treba osnažiti, a ograničavajuće snage sputavati.

---

<sup>1</sup> Konformizam je promjena ponašanja i uvjerenja prema grupi kao posljedica stvarnog ili zamišljenog pritiska grupe. Događa se kad se ljudi nađu u dvosmislenim ili novim situacijama i može se lako objasniti informacijskim utjecajem – drugi ljudi pružaju informacije o prikladnom ili očekivanom ponašanju. Konformizam se sastoji u nekritičnom stavu o zlu, u cilju da budemo društveno prihvaćeni od grupe s kojom se poistovjećujemo, pa čak i kad ona nema ispravne stavove (Aronson, Wilson, & Akert, 2005).

Željeno stanje epilog je procesa promjena, odnosno stanje koje se promjenom želi postići i kao takvo ponovo zamrznuti. Procesi odmrzavanja i kretanja, kao potprocesu procesa promjena, prethode potprocesu implementacije same promjene.

Menadžment organizacije i menadžment IS-a, kao pokretači promjena u životnom ciklusu razvoja informacijskog sustava, mogu koristiti različite taktike za savladavanje otpora u cilju postizanja odmrzavanja i kretanja. U literaturi se predlaže šest taktika za uspješno savladavanje otpora promjenama, i to:

1. taktika obrazovanja i komunikacija,
2. taktika sudjelovanja,
3. taktika olakšavanja i podrške,
4. taktika prilagođavanja,
5. taktika manipulacije i kooptiranja te
6. taktika prisiljavanja.

Kombinacija ovih taktika je poželjna i treba je primijeniti u biti i sadržaju promjena, odnosno u konkretnoj situaciji.

### 1.6 Dionici informatičkog razvoja organizacije

Dionik (engl. *stakeholder*) razvoja informacijskog sustava svaka je osoba koja na bilo koji način ima interes za njegovo postojanje. S obzirom na ulogu, dionike je moguće podijeliti u dvije osnovne interesne skupine. Jedna su skupina dionika zaposlenici koji se bave održavanjem i razvojem informacijskog sustava. Drugu skupinu dionika čine krajnji korisnici, bez obzira radi li se o unutarnjim ili vanjskim korisnicima.

Svaki od dionika, u skladu sa svojim odgovornostima, ima vlastiti pogled na IS, a te poglede je prilikom razvoja potrebno uskladiti. Zbog toga je razvoj informacijskih sustava kompleksan problem koji zahtijeva uključivanje svih dionika.

Prikaz veza među svim dionicima informatičkog razvoja organizacije nalazi se na slici 1.10.



projekte. To znači da se IS promatra kroz investicije i njegove direktne, lako kvantificirane koristi, npr. u obliku uštede automatizacijom poslovanja kojom se mjere klasični financijski pokazatelji. Dugoročni doprinos IS-a/IT-ja generira poslovne vrijednosti na druge načine koji imaju utjecaj na povećanje učinka. Tako se, primjerice, korištenje IS-a/IT-ja u strateškom upravljanju u obliku sustava za potporu strateškom upravljanju može odraziti u smanjenju rizika poslovanja. Dakle, vlasnici sustava snose odgovornost za financiranje razvoja informacijskog sustava, tj. naručuju i financiraju tehnološka rješenja.

### 1.6.2 Krajnji korisnik

Vrijednost IS-a za krajnje korisnike očituje se u povećavanju djelotvornosti i učinkovitosti izvršenja njihovih poslova. To znači da krajnji korisnici promatraju IS kroz funkcionalnost za izvršenje njihovih poslova i jednostavnost njegova korištenja. Postotak IT-jem poduprtih procesa orijentiranih prema krajnjim korisnicima, bez obzira radi li se o unutarnjim ili vanjskim, specifičan je i ovisi o djelatnosti organizacije.

Upravo zadovoljstvo vanjskih korisnika (kupci, dobavljači...) proizvodima i uslugama organizacije često predstavlja ključni element za ostvarenje zadanih poslovnih ciljeva. S druge strane, interni korisnici (operateri, administratori, vlasnici...) najbolje poznaju prednosti i nedostatke procesa u kojem sudjeluju. Zbog toga se prilikom utvrđivanja zahtjeva potrebno usredotočiti na zahtjeve krajnjih korisnika. Na taj način utvrđuju se područja postojećih usluga u kojima se primjenom IT-ja može poboljšati operacionalizacija te uspostaviti i održati zadovoljstva krajnjih korisnika. Na temelju njihovih poslovnih zahtjeva izvršava se:

- planiranje i prioritet IS/IT projekata,
- razvoj novih IT aplikacija,
- rad i održavanje postojećih IT-ja (npr. broj obrađenih upita krajnjih korisnika, prosječna brzina otklanjanja korisničkih problema...).

### 1.6.3 Analitičar sustava

Osoba koja mora sagledati problem u cjelini te uskladiti sve prethodne poglede na IS je analitičar sustava (engl. *system analyst*). Na osnovi utvrđenih poslovnih problema i potrebnih unapređenja organizacije, konzultirajući se s ostalim informatičkim ekspertima iz različitih područja, analitičar sustava treba utvrditi i specificirati zahtjeve korisnika koje novi IS treba zadovoljiti.

Čest je slučaj da analitičar sustava mora obavljati više poslova. Međutim, bez obzira na opseg dodijeljenog posla analitičar sustava mora imati široki spektar znanja o poslovnoj i informatičkoj tehnologiji.

### 1.6.4 Projektanti

Nakon što je analitičar sustava u suradnji s menadžmentom utvrdio poslovne zahtjeve, njihovo prevođenje u izvedive tehničke mogućnosti provodi projektant informacijskog sustava. Dakle, projektant promatra IS u izvođenju projekta do konačnog sustava. Projektant informacijskog sustava priprema detaljnu dokumentaciju za razvoj i integraciju sustava. Njegov je zadatak također da strukturiranje podataka obavlja u skladu s prethodno definiranim zahtjevima odabranog IS-a. U suradnji s analitičarom sustava izrađuje koncept uzimajući u obzir tehničke specifikacije. Uz to, priprema upute za implementaciju.

### 1.6.5 Programer

Promatranje informacijskog sustava kroz raspoložive hardverske i programske implementacije u sustav provode programeri. Oni konstruiraju, prilagođavaju i održavaju informacijski sustav. Posao programera je da piše, testira i otklanja neispravnosti te osmišljava i testira logičku strukturu za rješenje problema. Programeri konvertiraju zahtjeve korisnika u programski kod.

### 1.6.6 Menadžer informacijskog sustava

Menadžer informacijskog sustava je osoba koja nadzire rad i razvoj IS-a. Menadžer ujedno snosi i odgovornost za njegovo ispravno funkcioniranje. U fazi razvoja menadžer informacijskog sustava u suradnji s menadžerom poslovnog sustava planira učinke IS-a kojima se može poboljšati funkcionalnost procesa poslovnog sustava. Na taj način definira se i utjecaj IS-a na ostvarenje zadanih poslovnih ciljeva organizacije. S druge strane, u fazi planiranja i analize utjecaja paralelno surađuje i s analitičarom sustava, koji mora prepoznati te potom, u skladu s planom razvoja, specificirati zahtjeve krajnjih korisnika. Rezultat su ugovoreni poslovi informatizacije organizacije.

Razvoj IS-a od ideje do realizacije zahtijeva vrijeme i ne mora biti realiziran u kontinuitetu. Postoje mnogi ograničavajući faktori koji mogu utjecati na odgodu razvoja IS-a, stoga menadžer informacijskog sustava vodi i upravlja provedbom svih informatičkih projekata. Kada se pokrene određeni informatički projekt, menadžer IS-a uključuje ostale dionike. Formiranje interdisciplinarnog tima za izbor IS-a/IT-ja u promatranoj organizaciji dodatno postaje uvjetovano i razinom zrelosti organizacije za prihvaćanje novih informatičkih tehnologija.

### 1.6.7 IT konzultant

IT konzultant vanjski je ekspert područja koji surađuje s analitičarom sustava, sugerira i predlaže koje tehnologije bi bilo najbolje koristiti kako bi se zadovoljili svi poslovni ciljevi organizacije te kako bi se riješili dotadašnji problemi organizacije. Osnovni je zadatak IT konzultanta da strateški usmjeri klijenta u odabiru tehnologije, nabave tehnologije te da za sve to vrijeme osigura potrebnu podršku. Kao ekspert za to područje, uvelike pomaže organizaciji da odabere optimalno tehnološko rješenje.

### 1.6.8 Sudionik u procesu

Sudionik u procesu svaka je osoba koja direktno sudjeluje u svakodnevnom izvršenju bilo koje od aktivnosti koja pripadaju određenom procesu. To je osoba koja najbolje poznaje sve detalje i upravo zbog toga ima značajnu ulogu kod specifikacije zahtjeva. Analitičar sustava prilikom izrade specifikacije zahtjeva mora konzultirati svakog od sudionika u procesu kako bi rezultat bio relevantan. Jedan proces može imati više sudionika, a isto tako jedan sudionik može sudjelovati u više različitih procesa.

### 1.6.9 Menadžer poslovnog sustava

Menadžer poslovnog sustava je osoba koja nadzire rad i razvoj poslovnog sustava (izvršni direktor, član uprave za operativno poslovanje, upravitelj proizvodnje ili sl.). Menadžer ujedno snosi odgovornost za njegovu funkcioniranje. U fazi razvoja već je navedeno kako zajedno s menadžerom informacijskog sustava planira učinke IS-a kojima se može poboljšati funkcionalnost procesa poslovnog sustava. Na taj način definira se i utjecaj IS-a na ostvarenje zadanih poslovnih ciljeva organizacije. Osim toga, menadžer poslovnog sustava je taj koji komunicira s vlasnikom sustava i koordinira odobrenje i financiranje IS-a.

### 1.6.10 Administrator baze podataka (DB administrator)

Administrator baze podataka (engl. *database administrator*) je osoba odgovorna za performanse, integritet i sigurnost baze podataka. Njegov je zadatak da prema poslovnim modelima podataka, definiranim u zahtjevima odabranog IS-a, specificira i napravi tehničke modele. Rad administratora počinje kada se pokrene određeni informatički projekt. Menadžer informacijskog sustava koordinira radom administratora kao i ostalim sudionicima tima IS-a/IT-ja.

### 1.6.11 Vlasnik procesa

Vlasnik procesa je osoba unutar organizacije koja je u potpunosti odgovorna za određeni proces. Primjerice, bez obzira na zaposlenike koji sudjeluju u nekom od procesa, točno se zna tko je odgovoran, npr. voditelj prodaje za poslovne procese s područja prodaje. Vlasnik procesa kao odgovorna osoba poznaje sve osobine



aktivnosti koje pripadaju tom procesu. Ova je osoba ključna prilikom prikupljanja zahtjeva koju obavlja analitičar sustava jer poznaje sve detalje i relevantna je za iznošenje istih. Vlasnik procesa je taj koji koordinira i upravlja tijekom procesa na svim razinama. Jako je bitno znati tko snosi odgovornost za svako područje u organizaciji.

### 1.6.12 Isporučitelj informatičke opreme

Isporučitelj informatičke opreme u koordinaciji s menadžerom informacijskog sustava isporučuje potrebnu opremu. Nabava opreme unaprijed je definirana specifikacija onoga što je potrebno kako bi se osigurao pravilan rad IS-a u organizaciji. Proces nabave mora se odvijati u skladu sa zakonima o javnoj nabavi ako je to primjenjivo u organizaciji.

## 1.7 Literatura

1. Almansour, H., Beloff, N., & White, M. (2023). *IAI-CGM: A framework for intention to adopt IoT-enabled continuous glucose monitors* doi:10.1007/978-3-031-16072-1, 46
2. Brumec, J. (1997). Strategic Planning of Information Systems, *Journal of Information and Organizational Sciences*, 21(2), pp. 11–26.
3. Brumec, J. (1998). Strategic Planing of IS. *Journal of Information and Organisational Science*, 23(2), str. 11–26.
4. Brumec, J., & Vrček, N. (2001). Procjena učinaka korištenja IT metodom Balanced Scorecard. *Zbornik radova CASE 13*, pp. 175-183, CASE d.o.o. Rijeka 2001.
5. Brumec, J., Maček, V. & Dušak, V. (2000). Boston Consulting Group (Bcg) Matrix Analysis During the Strategic Planning of Government Administration Information Systems (Gais), *Journal of Information and Organizational Sciences*, 24(2), pp. 93–107.
6. Brumec, S. (2008). Predicting the Intention to Use Internet – a Comparative Study. *Journal of Information and Organizational Sciences*, 30(1).
7. Chih-Chien, W., Shao-Kang, L., & Wenchang, F. (2008). Extending the technology acceptance model to mobile telecommunication innovation: The existence of network externalities. *Journal of Consumer Behaviour Volume 7 Issue 2*, str. 101–110.
8. Davis, Bagozzi, & Warshaw. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management science*.
9. Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly* (13:3), str. 319–339.
10. Earl, M. J. (1989). *Management Strategies for Information Technology*. USA: Prentice-Hall, Inc.
11. Galić, M. (2010). Učeće organizacije. *MediAnali: međunarodni znanstveni časopis za pitanja medija, novinarstva, masovnog komuniciranja i odnosa s javnostima*, 4(7), pp. 179–194.
12. Grabavac, V. (1991). *Analiza i implementacija informatičkih sustava*. Zagreb: Školska knjiga.
13. Henderson, B. (1970). *The Product Portfolio*. The Boston Consulting Group, Boston.
14. Hsiu-Li, L. & Hsi-Peng, L. (2008). The role of experience and innovation characteristics in the adoption and continued use of e-learning websites. *Computers & Education* 51, str. 1405–1416.

15. Khosrow-Pour, M. (2003). *Information Technology and Organizations: Trends, Issues, Challenges & Solutions*. London: Idea group.
16. Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems-managing the digital firm*. Pearson.
17. Mathieson, K., Peacock, E., & Chin, W. (2001). Extending the Technology Acceptance Model: The Influence of Perceived User Resources. *Database For Advances in Information Systems*, Vol. 32.
18. Mustafa, A. S. & Ali, N. (2023). The adoption and use of moodle in online learning: A systematic review. *Information Sciences Letters*, 12(1), 341-351. doi:10.18576/isl/120129
19. Radošević, D. (2001). *Osnove teorije sustava*. Zagreb: Nakladni zavod Matice hrvatske.
20. Sikavica, P., Bebek, B., Skoko, H., & Tipurić, D. (1999). *Poslovno odlučivanje*. Zagreb: Informator.
21. Taylor, S., & Todd, P. A. (1995). Understanding Information Tchnoogy Usage: A Test of Competing Models. *Information System Research*.
22. Wettstein, T., & Kueng, P. (2002). A maturity model for performance measurement systems. *Management Information Systems*.



## 2 Planiranje učinaka informatičkih tehnologija

Sveukupni napredak znanosti i njezine primjene u svakodnevnom poslovanju ubrzavaju proces skraćivanja vremena aktualnosti informacija. Postoji čitav niz primjera kojima se može potkrijepiti prethodna tvrdnja, primjerice od najjednostavnije primjene informatičkih tehnologija u svrhu poboljšanja protoka informacija pa sve do matematičkih algoritama kojima se, korištenjem informacija u prognostičke svrhe, podiže njihova vrijednost, a samim tim ostvaruje preduvjet za postizanje konkurentske prednosti. Dugoročnim planiranjem razvoja IS-a želi se osigurati buduće zadovoljenje potreba organizacije za informacijama i resursima u trenutku kada je potreba iskazana.

Ulaganje u IT te u informatičku opremu općenito predstavlja velik, ako ne i najveći udio u troškovima organizacija. Često se ističe polemika u istraživanjima pruža li IT vrijednost i konkurentsku prednost, no većina istraživača slaže se s izjavom da u prosjeku implementacija IT-ja poboljšava performanse i povećava vrijednost organizacije (Dahlberg, Kivijärvi, & Saarinen, 2015).

Kratak zaključak prikazan je pod nazivom *paradoks efekt*: „Veće ulaganje u IT, sporiiji rast produktivnosti“. To je razlog zašto se mnoga današnja istraživanja fokusiraju na strateško planiranje informacijskih sustava i isplativost investiranja u IS/IT.

Strateškim planiranjem IS-a otvara se mogućnost za maksimiranjem njegova utjecaja na učinkovitost organizacije, čime se smanjuje tehnološki rizik ulaganja u IS/IT.

### 2.1 Temeljne postavke strateškog planiranja

Kako bi se uopće moglo govoriti o strateškom planiranju informacijskog sustava, potrebno je prvo nekoliko riječi posvetiti strateškom planiranju. Postoje dvije skupine autora koje se razilaze u poimanju bar jednog aspekta strategije (Wehrich & Koontz, 1993)<sup>1</sup>. Jedna skupina autora pod pojmom strategije podrazumijeva krajnje točke (svrha, misija, ciljevi) i sredstva za njihovo ostvarenje (politike i planovi). Druga pak skupina stavlja veći naglasak na sredstva za ostvarivanje krajnjih točaka nego na same krajnje točke.

U ovoj knjizi pod pojmom strategije podrazumijeva se odlučivanje o rasporedu i načinu korištenja sveukupnih resursa organizacije. U tom kontekstu, strateško je planiranje usklađivanje resursa organizacije s čimbenicima iz okoline. Rezultat

---

<sup>1</sup> Izraz strategija izveden je iz grčke riječi *strategos* što znači "vojskovođa"

strateškog planiranja je strateški plan na temelju kojega je moguće mjeriti uspjeh provođenja strategije.

Kao prvo, potrebno je utvrditi koje su trenutačne funkcije organizacije. Odgovarajući na to pitanje, moguće je ustanoviti odstupanje od definirane misije organizacije. Misiju svake organizacije jasno definira vlasnik, tj. osnivač. Može se reći da misija odgovara na pitanje zašto je organizacija osnovana, tj. koja je njezina osnovna funkcija. Preispitivanje misije može ukazati na potrebu cjelokupne reorganizacije djelatnosti organizacije. Daljnji razvoj i sve aktivnosti vezane za djelovanje organizacije odvijaju se u svrhu što kvalitetnijeg ostvarivanja misije.

U skladu s definiranom misijom, tj. razlogom zašto organizacija postoji, potrebno je definirati željeno stanje organizacije, tj. viziju budućeg razvoja organizacije. Mandat za upravljanje organizacijom dodjeljuje se osobi ili skupini osoba na temelju vizije razvoja organizacije. Vizijom razvoja određuju se opće smjernice razvoja organizacije kao ciljevi koji se žele postići u svrhu ostvarenja misije.

Definiranjem strategije razvoja organizacije, tj. odlučivanjem o sveukupnom rasporedu resursa organizacije, odabire se način ostvarenja vizije. U skladu sa strategijom definira se strateški plan. Na osnovi potrebnog usklađivanja resursa organizacije s čimbenicima iz okoline određuju se vremenski određeni ciljevi koje je potrebno ostvariti kako bi se postigle definirane strategije. Drugim riječima, strateški plan treba odgovoriti na četiri pitanja koja je moguće prikazati formulom (2.1).

$$\text{Strateški plan} = f(\text{što, tko, čime, kada})$$

(2.1)

Može se reći da je strateški plan plan akcija potrebnih za ostvarivanje strategije i da on svojim sadržajem mora pružiti odgovor na četiri pitanja, i to:

- Što se želi postići? Odgovarajući na ovo pitanje, opisuje se niz potrebnih akcija u svrhu postizanja zadanog cilja. Za svaku od akcija potrebno je navesti njezinu svrsishodnost i jamstva njezina ostvarenja.
- Tko je odgovoran? Odgovarajući na ovo pitanje, definiraju se ljudski resursi i njihove odgovornosti pri provođenju planiranih aktivnosti.
- Čime se želi postići? Postojeći i novi resursi potrebni za provođenje željenog cilja.
- Kada se želi postići? Okvirni vremenski raspored ostvarivanja akcija, tj. planirano vrijeme ostvarenja.

Upravo ovako akcijski orijentiranom formulacijom strateškog planiranja stvoreni su preduvjeti za kontinuiranu kontrolu i upravljanje ostvarivanjem strategije, što bi trebalo doprinijeti kvalitetnom razvoju organizacije. Međutim, prije svega potrebno je postizanje dogovora o samom procesu strateškog planiranja.

Uključujući razjašnjenje zakonskih okvira u kojima organizacija djeluje, potrebno je preispitati misiju organizacije. Pritom treba imati u vidu sva zakonska i pravna ograničenja u kojima organizacija djeluje. Jedino potpuno razumijevanje i potpora prethodnim analizama svih zainteresiranih dionika omogućit će daljnje razvijanje definiranog procesa strateškog planiranja.

Za odabir strategije potrebno je dobro poznavanje trenutačne situacije u okolini organizacije, kao i u njoj samoj.<sup>1</sup> Donošenje strateških odluka naprečac može biti kobno za organizaciju. Isto tako, česte promjene strategije zbog nekonzistentnosti djelovanja organizacije mogu također dovesti do gubljenja povjerenja - kako suradnika tako i zaposlenika organizacije – te općenito informacijske "zbrke" u informacijskim tijekovima organizacije.

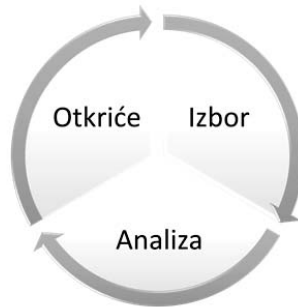
Nakon istraživanja poslovnog djelokruga organizacije, u svrhu ostvarenja vizije potrebno je istražiti trendove razvoja postojeće poslovne tehnologije. Jedino je na taj način moguće uočiti i pretpostaviti probleme te tako stvoriti preduvjet za identifikaciju strateških ciljeva. U tom kontekstu, a u skladu s efektivnom vizijom organizacije, potrebno je preispitivanje postojećeg strateškog plana, kao i njegovo prilagođavanje.

Generalno govoreći, dosadašnje faze procesa strateškog planiranja temeljene su na tri uzročno-posljedična koraka prikazana na slici 2.1.

Korake provođenja strateškog plana, tj. ostvarivanje niza akcija, potrebno je upotpuniti adekvatnim jamstvima jer određene doze nesigurnosti mogu dovesti do nemogućnosti ostvarivanja željenih ciljeva. Čak ni pomoću detaljne analize često nije moguće predvidjeti skorašnje promjene okoline, stoga je nužno upravljanje provedbom strateškog plana.

---

<sup>1</sup> O metodama i tehnikama analize postojećeg stanja bit će više riječi u sljedećim odjeljcima.



*Slika 2.1 Proces strateškog planiranja (Stacey, 2007)*

## 2.2 Odnos poslovne, IS i IT strategije

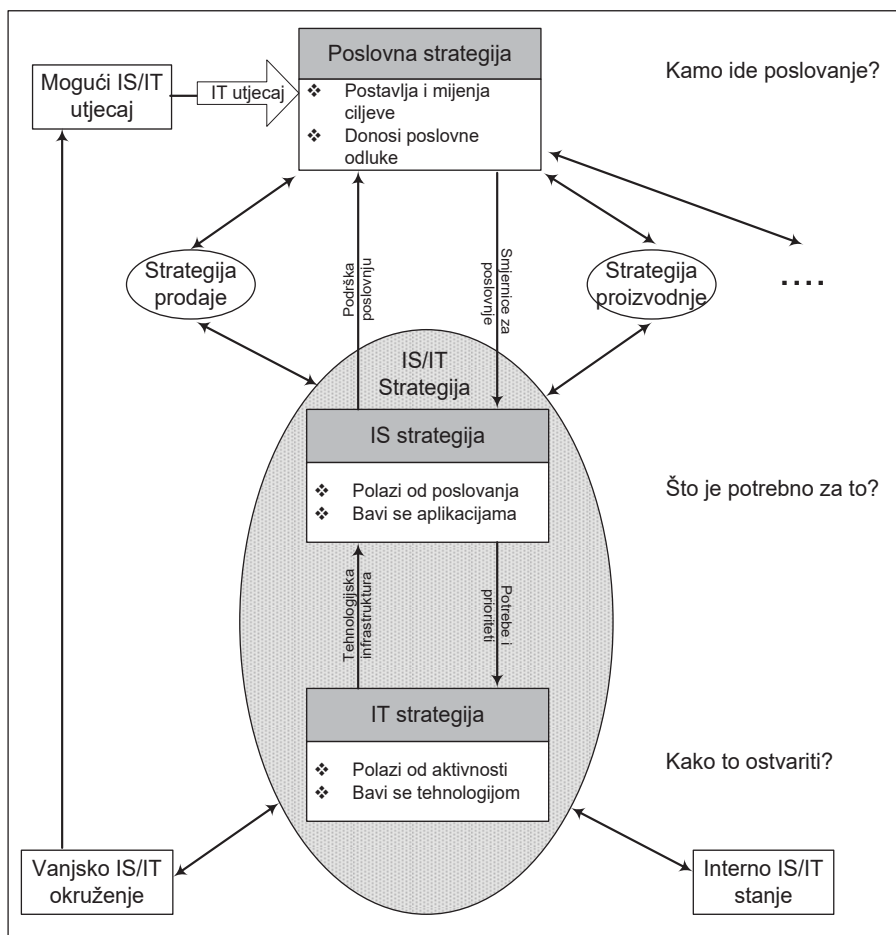
U prvom odjeljku prikazana su promišljanja o pretpostavkama za učinkovitu primjenu IT-ja u organizacijskom sustavu. Razmatranje investiranja u IS/IT trebalo bi promatrati kroz organizacijski sustav u kontekstu ranije navedenih značajki. Promatrajući informacijski sustav kao podsustav organizacijskog sustava, potrebno je uočiti njegovu ulogu u poboljšanju informacijskih i materijalnih tijekova organizacije. Optimalno organizirani procesi i tijekovi organizacije, potpomognuti novim informatičkim tehnologijama, trebali bi omogućiti brže prilagođavanje organizacije novonastalim uvjetima, a samim tim i kvalitetnije ostvarenje strategije, odnosno provođenje strateškog plana organizacije.

Primjena IS-a/IT-ja ovisi i o okruženju: o konkurenciji, o dobavljačima, o kupcima. Brzi razvoj informatičkih tehnologija i njihova implementacija u organizacijama, u kontekstu klasifikacije na slici 1.4, zahtijeva promatranje IS-a/IT-ja kao podsustava organizacijskog sustava. Na taj način postiže se novi način promišljanja o strateškom planiranju koji ima utjecaj na sve razine upravljanja. Odnos IS/IT strategija prikazan je na slici 2.2. Upravo takav odnos strategija korišten je kao polazište jer ukazuje na potrebu holističkog pristupa strateškom planiranju. To znači da je potrebna detaljna analiza značenja IS-a/IT-ja za svaku organizaciju.

Razgraničavanjem strategije razvoja informacijskog sustava (IS strategije) i strategije razvoja informatičko-tehnološkog podsustava organizacije (IT strategije), u kontekstu poslovnog sustava i njegovih ostalih pripadajućih strategija, IS/IT strategija postaje nerazdvojivi dio strateškog planiranja organizacije uopće. Štoviše, sve su strategije uzročno-posljedično povezane kao na slici 2.2. Sukladno navedenom, strateško planiranje informacijskih sustava moguće je definirati kao dugoročno planiranje korisnih učinaka informacijskog sustava i upotrebe informatičkih tehnologija u



poslovanju u sklopu strateškog planiranja razvoja poslovnog sustava kao cjeline (Brumec, Dušak, & Vrček, 2000).



Slika 2.2 Odnos razvojne, IS i IT strategije (Ward & Griffiths, 1996)

Prihvaćajući prethodna polazišta, strateško planiranje informacijskih sustava stavljeno je u kontekst strateškog planiranja razvoja organizacije. Kao i kod metoda strateškog planiranja općenito, tako je i kod strateškog planiranja informacijskih sustava osnovni nedostatak učinkovita kontrola provedbe strateškog plana. U tu svrhu, strateško planiranje informacijskih sustava zahtijeva interdisciplinarni znanstveni pristup. Sintezom određenih matematičkih, informatičkih i ekonomskih metoda trebalo bi se doprinijeti poboljšanju kvalitete strateškog planiranja informacijskih sustava. Jednom kada je strateška perspektiva IS-a/IT-ja uspostavljena, ona treba biti kontinuirano unaprijeđena – odnosno strategije i planovi moraju se konstantno ažurirati (Ward & Peppard, 2002).

## 2.3 Pregled metoda i tehnika strateškog planiranja informacijskih sustava

U prethodnom odjeljku pokazan je pristup prema kojem se informacijski sustav ne može planirati zasebno. On je uvijek model nekog organizacijskog sustava. Ovakav pristup ukazuje na mogućnost korištenja klasičnih metoda strateškog planiranja u svrhu strateškog planiranja informacijskih sustava. Najprimjereniji prikaz važnosti nekih metoda i tehnika sadržanih u ovom poglavlju koje se koriste u strateškom planiranju prikazuje profesor Ralph D. Stacey, svrstavajući iste u poglavlje podnaslova *Tradicionalna mudrost* (Stacey, 2007; Hammer, 1990). Pod tradicionalnom mudrošću on podrazumijeva sve one preporuke i objašnjenja za uspješnu provedbu strateškog menadžmenta koje predlaže većina menadžera i konzultanata te se mogu pronaći u većini knjiga i udžbenika o menadžmentu. Upravo dio tih metoda, prema SPIS-u, sastavni je dio procesa strateškog planiranja informacijskih sustava (Alamri, Almutiri, Ballahmar, & Zafar, 2016). Zajednička pretpostavka tradicionalnog pristupa temelji se na približavanju organizacije stanju stabilne ravnoteže prema obrascima ponašanja koje obilježavaju predvidivost, pravilnost, dosljednost i sklad.

Organizacija u stanju stabilne ravnoteže slijedi predvidljiv put na kojem nema nikakvih iznenađenja. U tom kontekstu, preduvjet je uspjeha prilagođenost organizacije okolini, što znači da bez poznavanja buduće okoline<sup>1</sup> nije moguće planirati svoju buduću poziciju. Poznavanje budućeg stanja okoline otvara mogućnost definiranja alternativnih pozicija, a samim time ciljeva i akcija potrebnih za njihovo ostvarivanje. Moguće alternative postaju strateške opcije organizacije, od kojih se odabire strategija organizacije prema kriterijima prihvatljivosti, ostvarivosti i prikladnosti.

### 2.3.1 Analiza poslovnog portfelja

Bazirajući svoju ideju na fazama životnog ciklusa proizvoda<sup>2</sup> i krivulji iskustva<sup>3</sup>, konzultantska tvrtka *Boston Consulting Group* (BCG) pod svojim je akronimom razvila matricu analize portfelja ili matricu BCG. Metoda je izvorno korištena za analizu poslovnog portfelja. Međutim, u smislu upotrebe strateškog planiranja informacijskih

---

<sup>1</sup> Pojam okoline organizacije analiziran je u okviru definicije okoline sustava kod objašnjavanja sustavskog pristupa.

<sup>2</sup> Kao što je navedeno u početku ovog poglavlja, ovim poglavljem ograničavamo se samo na metodologije i tehnike strateškog planiranja koje se spominju u metodi SPIS. Više o fazama životnog ciklusa proizvoda kao i odnos sa strateškim opcijama prikazao je M. Porter (1985) u svom djelu *Competitive Advantage: Creating and Sustaining Superior Performance*, New York.

<sup>3</sup> Krivulja iskustva bazira se na promatranju organizacije i njezine efikasnosti s obzirom na opseg proizvodnje. Kako se organizacija kreće niz krivulju učenja smanjujući troškove, može se smanjiti i cijenu proizvoda. Tako je krivulje cijene i troškova moguće povezati s idejom životnog ciklusa proizvoda i različitim strategijama.

sustava (SPIS-a), ne analizira portfelj već procjenjuje kakvi se učinci mogu postići ako se „informatizira“ neka skupina poslovnih procesa. Zbog navedenog, vrijednosti na apscisi i ordinati imaju druga značenja nego kod upravljanja portfeljem.

Polazeći od poslovnih procesa, primjena analize BCG kod strateškog planiranja informacijskih sustava modificira prethodno navedene kriterije opisane koordinatnim osima u prethodnom poglavlju. Matrica BCG u ovom slučaju podijeljena je na četiri podmatrice s obzirom na sljedeće kriterije:

- uvjetovanost izvršavanja procesa primjenom informatičkih tehnologija i
- doprinos novih informatičkih tehnologija smanjenju troškova funkcioniranja organizacije.

Svaki od procesa u organizaciji raspoređuje se na temelju navedenih kriterija u jednu od četiri podmatrice, kao na slici 2.3.



*Slika 2.3 Matrica procjene važnosti IS-a za organizaciju (Brumec J., 1997; Hammer, 1990)*

S obzirom na značenje IT-ja za cjelokupni sustav, na temelju analize BCG procese u organizaciji moguće je podijeliti na sljedeće četiri kategorije:

- Potporni procesi na koje IT u svrhu smanjenja troškova imaju nizak doprinos i malu uvjetovanost uspjelosti izvršenja. IT-jevi u takvim procesima nisu nužni jer se potporni procesi mogu odvijati i bez njih.
- Operativni procesi imaju veliku uvjetovanost primjene IT-ja, iako njihova modernizacija nema bitan doprinos smanjenju troškova poslovanja. Ovi procesi, najjednostavnije rečeno, zahtijevaju korištenje IT-ja.
- Izgledni procesi čije izvođenje primjenom IT-ja može imati visoki doprinos u svrhu smanjenja troškova poslovanja organizacije, iako imaju nisku razinu

uvjetovanosti uspjelosti izvršenja primjenom IT-ja. Korištenje IT-ja u ovoj kategoriji procesa ima izgledno značenje za poslovanje organizacije.

- Strateški procesi čija je uvjetovanost izvršenja procesa primjenom modernih IT-jeva velika, a istovremeno primjena IT-ja ima visok doprinos na smanjenje troškova funkcioniranja organizacije. U ovoj je kategoriji IT ključan za poslovanje organizacije.

S obzirom na važnost IT-ja i sukladno prethodnim kategorijama, mogu se izvući zaključci o područjima, strategiji i prioritetima modernizacije IT-ja u cjelokupnoj organizaciji. Nastavno na to, informatizacija potpornih procesa ima najniži prioritet, dok najviši prioritet imaju operativni i strateški procesi. Međutim, rezultati ovakve analize nedovoljni su kao apsolutni pokazatelji redosljeda uvođenja novog informacijskog sustava. Naime, osim dviju diskretnih varijabli pri određivanju prioriteta, potrebno je uzeti u obzir i druge pokazatelje, kao što su, primjerice, spremnost izvršitelja promatranog procesa za prihvaćanje promjena, njihova raspoloživost za edukaciju i sl.

### 2.1.1. Analiza prednosti, nedostataka, prilika i prijetnji

Naziv metode SWOT proizlazi iz početnih slova engleskih riječi: *Strengths* (prednosti), *Weaknesses* (nedostatci), *Opportunities* (prilike) i *Threats* (prijetnje). Svrha je metode da se na temelju analize stanja organizacije i njezina okruženja, u skladu s navedenim pojmovima, procjenjuje izvedivost pothvata i u skladu s tim napravi budući niz akcija za ostvarivanje strateških ciljeva. Kritike SWOT-a najviše se odnose na način na koji menadžeri provode i rabe dobivene rezultate (Hill & Westbrook, 1997). U provedenom istraživanju, od devedeset razmatranih izvješća poduzeća, njih dvadeset koristilo je SWOT, od čega je četrnaest konzultantskih tvrtki. Osnovni problem kod provođenja SWOT analize koji autori vide je nekorištenje nijednog od rezultata SWOT-a u daljnjem procesu strateškog planiranja. Na taj se način dovodi u pitanje potreba provođenja SWOT analize.

Iz tog je razloga vrlo važno naglasiti da sve prosudbe moraju dati stručnjaci iz organizacije u kojoj se provodi projekt unapređenja IS-a/IT-ja, tj. nitko od vanjskih konzultanata, zato što izvršitelji poslova u organizaciji najbolje poznaju sve elemente okruženja u kojem rade (vanjske i unutarnje). Na taj se način SWOT analiza koristi i kod strateškog planiranja IS-a/IT-ja. Sposobnosti organizacije da uspješno provede projekt IS-a/IT-ja i da prihvati nove IT-jeve okarakterizirane su kao prednosti ili nedostatci. Prijetnje ili prilike odnose se na moguću procjenu učinaka novih IT-ja

ovisno o njezinoj zastupljenosti u okruženju. Na slici 2.4 prikazane su četiri strateške smjernice.

		<b>PREDNOSTI</b> (engl. strengths)	<b>NEDOSTATCI</b> (engl. weaknesses)
Procjena učinaka novih IT-jeva	<b>POGODNOSTI</b> (engl. opportunities)	Koristiti nove IT-jeve sa očekivanim uspjehom	Biti oprezan u korištenju novih IT-jeva
	<b>PRIJETNJE</b> (engl. threats)	Istražiti i identificirati mogućnosti korištenja novih IT-jeva	Zaštititi se pomoću korištenja novih IT-jeva

*Sposobnost prihvaćanja novih IT-jeva u organizaciji*

**Slika 2.4 Smjernice korištenja novih IT-jeva procijenjenih na temelju SWOT analize**

Na ovaj način moguće je postići uravnoteženost prosudbe utjecaja IS-a/IT-ja s obzirom na mogućnosti njegove primjene i sposobnosti korisnika da ih prihvate. Samim time, primjenom SWOT analize omogućuje se uočavanje visokorizičnih IS/IT strategija. Donošenje sudova o stanju organizacije s aspekta IS-a/IT-ja također se zasniva na identifikaciji elemenata SWOT-a. U tom kontekstu, mogućnost primjene novih tehnologija smatra se vanjskim čimbenikom, dok se sposobnost organizacije za njihovo prihvaćanje smatra unutarnjim čimbenikom. Tako, primjerice, prednost, odnosno slabost, može biti stav uprave prema korištenju IS-a ili postojanje tehničke podrške za njegovo održavanje ili razvoj. Prijetnja ili prilika, npr., može biti tehnološka promjenjivost okoline ili postojanje odgovarajućih tehnologija. Na taj način provođenje SWOT analize omogućuje otkrivanje visokorizičnih IS/IT strategija.

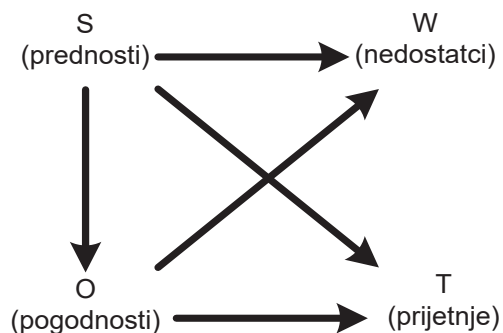
Neka je SWOT analizom utvrđena pogodnost u vidu raspoloživih informatičkih tehnologija i prednost u obliku sposobnosti organizacije za njihovo prihvaćanje. Strategija korištenja IS-a/IT-ja koja je izvedena na osnovi prethodnih sudova prilično sigurno omogućuje organizaciji ostvarenje zadanih strateških ciljeva. Ako pak postoje raspoloživi IT-jevi (pogodnost), a organizacija nema sposobnosti za njihovo prihvaćanje (nedostatak), korištenje IS-a/IT-ja u ostvarenju zadanih strateških ciljeva nosi visoki stupanj rizika. Zbog toga je potrebno biti oprezan pri utvrđivanju IS/IT strategija. Naime, može se dogoditi da implementacija novih IT-jeva, zbog

nesposobnosti organizacije za njihovo prihvaćanje, ne ostvari očekivani učinak na ostvarenje zadanih strateških ciljeva ili ostvari djelomični učinak sa znatnim zakašnjenjem. Iz tog je razloga potrebno utvrditi strategije kojima će se podignuti razina sposobnosti prihvaćanja novih IT-jeva. Jake resurse IS-a/IT-ja u organizaciji (prednost) u situaciji postojanja prijetnji potrebno je koristiti za detaljno istraživanje mogućnosti primjene IS-a/IT-ja i utvrđivanja strategija za otklanjanje mogućih prijetnji. Utvrđivanje primjene ili pak otklanjanja utvrđenog nedostatka sposobnosti prihvaćanja IS-a/IT-ja moguće je promatranjem IS-a/IT-ja u kontekstu elemenata SWOT-a cijelog poslovnog sustava. To znači da je, osim utvrđivanja elementa SWOT promatranjem primjene novih tehnologija (vanjske okolnosti) i sposobnosti njihova prihvaćanja (unutarnje okolnosti), u analizi stanja potrebno proširiti poglede korištenja IS-a/IT-ja u okviru poslovnog sustava.

Buduće razdoblje u kojem se aktivnosti odvijaju moguće je podijeliti u niz vremenskih intervala omeđenih diskretnim trenutcima. Dinamika SWOT analize za praćenje stanja novih okolnosti zasniva se na parcijalnim SWOT analizama u tim diskretnim trenutcima. Na osnovi otklonjenih prijetnji i nedostatka te iskorištenih pogodnosti i prednosti, može se konstatirati promjena strateškog položaja te potom, prema potrebi, korigirati planirani niz akcija. Tako se periodičnom primjenom SWOT analize omogućuje upravljanje razvojem strateške situacije.

### 2.3.2 Proširena SWOT analiza

Istraživanja i primjena SWOT-a u prethodno predloženom smjeru napravljeni su u okviru proširene SWOT analize (pSWOTa) prilikom utvrđivanja strategija razvoja IS-a. Prema pSWOTa, za razliku od klasičnog pristupa, prepoznate su dvije neposredne, dvije posredne i jedna agresivna strateška alternativa, čiji je smjer djelovanja grafički moguće prikazati slikom 2.5.



Slika 2.5 Alternativne strateške smjernice pSWOTa

Prikazani smjerovi djelovanja predstavljaju slijedeće strategije:

- (S,W) – neposredna strategija koja koristi organizacijske prednosti za otklanjanje organizacijskih nedostataka,
- (O,T) – neposredna strategija koja koristi pogodnosti okruženja za otklanjanje prijetnji iz okruženja,
- (S,T) – posredna strategija koja koristi organizacijske prednosti za otklanjanje prijetnji iz okruženja,
- (O,W) – posredna strategija koja koristi pogodnosti okruženja za otklanjanje organizacijskih nedostataka,
- (S,O) – agresivna (*shape-to-future*) strategija koja istovremeno koristi prednosti organizacije i pogodnosti okruženja radi postizanja ciljeva.

U kontekstu strateškog planiranja informacijskih sustava, SWOT analizom potrebno je procijeniti sposobnost organizacije da prihvati nove IT-jeve. U svrhu povezivanja elemenata SWOT-a sa strateškim planom, metoda pSWOTa predviđa konstruiranje zasebne tablice za svaki zadani strateški cilj (ZSC)<sup>1</sup>. Metoda pSWOTa ima četiri faze:

1. Polazni elementi SWOT analize: Za svaki ZSC treba utvrditi elemente SWOT analize i upisati ih u tablicu;
2. Rangiranje elemenata SWOT-a: Za utvrđene elemente pojedinačnog cilja potrebno je odrediti rang, odnosno utvrditi snagu njihova utjecaja na ostvarenje zadanih ciljeva;
3. Nalaženje elemenata strategije za realizaciju zadanog strateškog cilja: Utvrditi strategiju koja će omogućiti ostvarenje svakog zadanog strateškog cilja, tj. aktivnosti upotrebe organizacijskih resursa;
4. Četvrta faza sastoji se od tri koraka:
  - 4.1. Utvrđivanje specifičnih aktivnosti za provedbu prethodno definiranih strategija;
  - 4.2. Određivanje mjernih jedinica za aktivnosti definirane u prethodnom koraku;
  - 4.3. Utvrđivanje drugih strateških ciljeva na koje utječe svaka promatrana aktivnost.

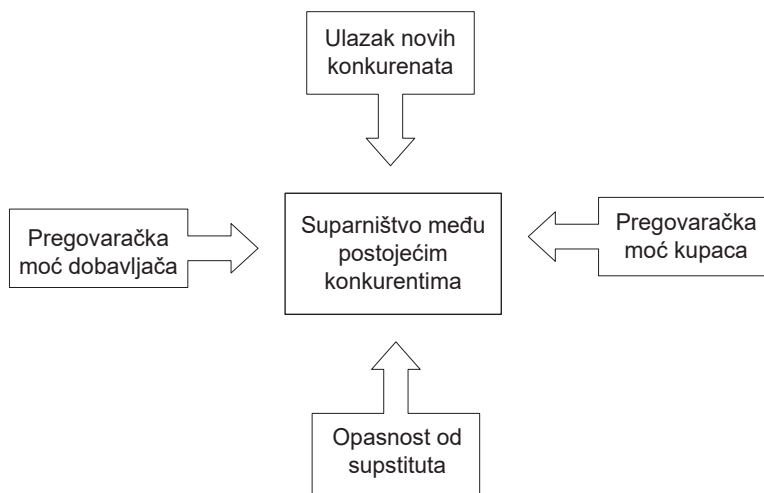
---

<sup>1</sup> Kvantificirani strateški ciljevi izvedeni iz vizije na prijedlog menadžmenta, a potvrđuje ih vlasnik. Kao takvi predočuju glavne ciljeve organizacije zbog čega imaju pridjev "zadani". Svaki zadani strateški cilj mora sadržavati sljedeće: oznaku cilja, smjer promjene, objekt promjene, naziv mjere, oznaku mjere, "As is" i "To be" vrijednosti i mjernu jedinicu.

### 2.3.3 Porterov model pet konkurentskih sila

Michael Porter stavio je klasične ekonomske teorije o tržišnim oblicima u funkciju analiza konkurentskih prednosti tvrtke (Porter, 1985)<sup>1</sup>. Definiranje tržišta temeljeno je na utjecaju skupina potrošača koji imaju slične zahtjeve i strateških skupina konkurenata koji se natječu kako bi zadovoljili te zahtjeve. Ovakve smjernice definiranja tržišta omogućuju identifikaciju njegove strukture, što je moguće prikazati slikom 2.6. Porter je strukturu opisao pomoću pet konkurentnih sila (5F) koje su prikazane na slici 2.6 (Porter, 1985; Bernroider, 2002):

- ulazak novih konkurenata,
- opasnost od supstituta,
- pregovaračka moć kupaca,
- pregovaračka moć dobavljača,
- suparništvo među postojećim konkurentima.



*Slika 2.6 Pet konkurentskih sila koje determiniraju industrijsku profitabilnost (Porter, 1985)*

On polazi od klasičnog modela SCP (engl. *structure-conduct-performance*)<sup>2</sup>, kojim se tvrdi kako tržišna situacija (industrijska struktura) determinira ponašanje tvrtke. Zbog toga je potrebno postojećim "igračima", koristeći zalihe profita, stvoriti barijere ulaska

<sup>1</sup> Više o ovoj problematici koja se bazira na Porterovim razmišljanjima, moguće je pronaći u njegovoj knjizi *Competitive Advantage: Creating and Sustaining Superior Performance*; The Free Press, New York, 1985

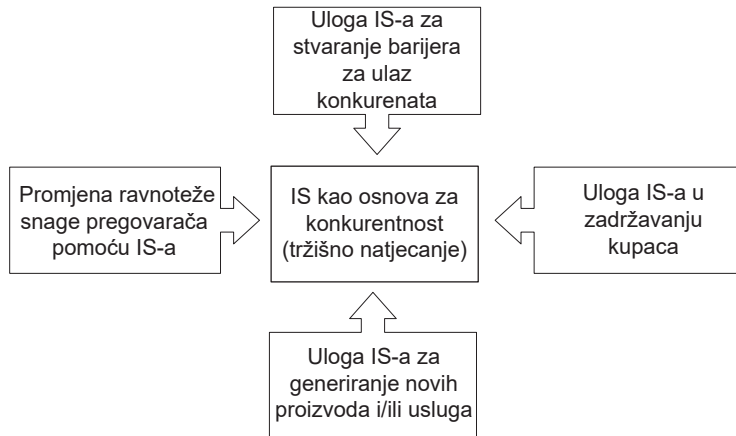
<sup>2</sup> SCP model polazi od tradicionalne klasifikacije tržišnih situacija: savršena konkurencija, monopol, oligopol. Detaljnija obrada ovog modela prikazana je u doktorskoj disertaciji Nikše Alfrevića *Konkurentska prednost velikih poduzeća i upravljanje organizacijskim promjenama*.



drugim „igračima“. Poduzeća koja to mogu priuštiti zauzimaju snažne pozicije na tržištu. Ako su pak barijere niske, to prisiljava postojeće "igrače" na konkuriranje troškovima. Druga prijetnja narušavanju konkurentne pozicije je mogućnost uvođenja supstituta trenutačnom proizvodu. Pregovaračka moć kupaca ovisit će o nekoliko faktora. Ako kupci kupuju veliki broj proizvoda iz ponuđenog asortimana ili ako su kritični element proizvodne kvalitete, imat će veliku pregovaračku moć, što će utjecati na smanjenje cijene proizvoda. Kao i kod kupaca, malobrojnost dobavljača ili prevelika ovisnost o njihovoj kvaliteti povećavat će moć dobavljača, samo što je ovdje, za razliku od kupaca, uvjetovano povećanje cijene izlaznog proizvoda poduzeća u skladu s pregovaračkom moći dobavljača. Peta je sila suparnička sila među konkurentima, koja iziskuje dodatna sredstva za marketing ili pak snižavanje cijena u odnosu na cijene konkurencije.

Jednom kad se definiraju sile, potrebno je donijeti odluku koja je sila najvažnija. Svrha identifikacije tržišne strukture i izvora konkurentskih prednosti je izvođenje strategija koje će biti u skladu s okolinom.

U kontekstu strateškog planiranja informacijskih sustava, ovaj model moguće je koristiti za ocjenjivanje pogodnosti utjecaja IS-a na svaku od pet snaga spomenutih sila (F). Krajnji je cilj procjena učinaka novog IS-a na povećanje tržišne konkurentnosti i razvojne stabilnosti poduzeća. Prvo je potrebno procijeniti važnost pet spomenutih sila te potom, u skladu s procjenom, povezati pogodnosti korištenja IS-a za svaku od njih. Definirajući pokazatelje stupnja povrata investicija, klasičnim 5F modelom određuje se cjelokupna profitabilnost poslovanja organizacije te tako povezuje način određivanja pogodnosti korištenja IS-a. Pogodnosti korištenja potrebno je tražiti u četiri aspekta: (1) pregovaračka moć dobavljača, (2) zadržavanje kupaca, (3) proizvodnja novih proizvoda i (4) ulazak konkurenata. U kontekstu pregovaračke moći kupaca, IS može poduprijeti suradničku snagu za smanjenje troškova. IS u odnosu s kupcima omogućuje kategorizaciju i diferencira skupinu kupaca u cilju promjena cijena dostave za određene skupine korisnika (segmentacija). Pogodnosti korištenja IS-a u kontekstu Porterova modela 5F prikazane su slikom 2.7.

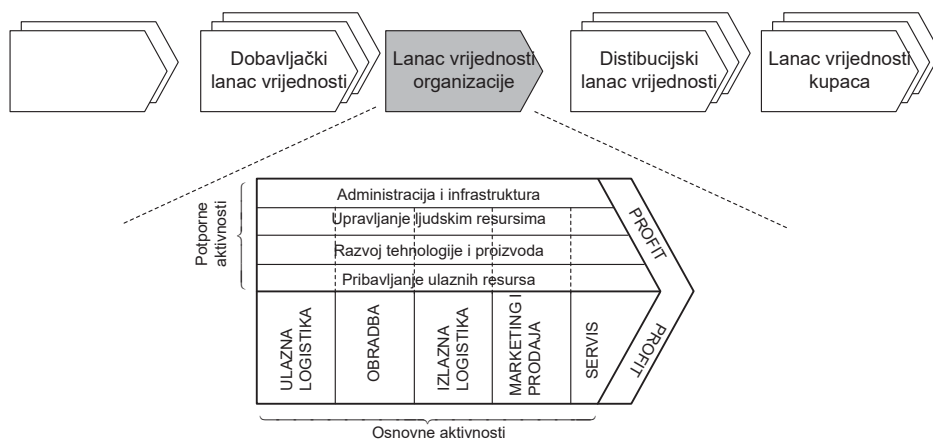


Slika 2.7 Pogodnosti korištenja IS-a u kontekstu Porterova modela 5F

U tom kontekstu potrebno je analizirati učinak IS-a na ispunjenje želja kupaca ili njihovo zadržavanje kroz povećanje izlaznih troškova (engl. *switching costs*) kako ne bi otišli drugim prodavačima. Kako bi se spriječio ulaz novih konkurenata, potrebno je analizirati učinak IS-a na postavljanje teško dostižnih svojstava (cijena, kvaliteta, isporuka itd.) te se zaštititi od zamjenskih proizvoda (supstituta).

### 2.3.4 Analiza lanca vrijednosti

U svrhu određivanja smanjenja troškova, odnosno diferencijacije proizvoda i usluga, Porter je razvio analizu lanca vrijednosti – engl. *value chain model* (slika 2.8). Lanac vrijednosti prikazuje način stvaranja novih vrijednosti proizvoda od sirovina, preko dobavljača, proizvodnje te preko distributera do krajnjeg potrošača. Prva faza analize je identifikacija karika lanca presudnih za ostvarivanje niskih troškova.



Slika 2.8 Vrijednosni sustav i generički lanac vrijednosti (Porter, 1985)

Strategija se usklađuje s položajem organizacije u ukupnom lancu vrijednosti. Svaka organizacija uzima *inpute* te koristi neke procese za obradu, tj. *inputima* dodaje neku vrijednost kako bi proizvela *outpute* koji će biti plasirani sljedećoj karici u lancu. Prilikom analize lanca vrijednosti unutar organizacije, uvedena je podjela na osnovne i potporne aktivnosti (slika 2.8). Pod osnovnim aktivnostima podrazumijevaju se:

- Ulazna logistika, tj. organiziranje ulaznih resursa. Ona sadrži sve aktivnosti vezane uz *inpute* (primanje, skladištenje, rukovanje i distribucija za daljnju obradu te cjelokupno upravljanje zalihama).
- Obrada je u proizvodnim organizacijama proizvodnja, tj. proces transformacije *inputa* u *output*.
- Izlazna logistika, tj. organiziranje distribucije svojih proizvoda i/ili usluga. Njome je obuhvaćena priprema distribucije *outputa* prema idućoj karici u lancu, uključujući i skladištenje u svojoj organizaciji.
- Marketing i prodaja obuhvaćaju aktivnosti vezane za upoznavanje potrošača s proizvodom i osiguravanje njegove raspoloživosti.
- Servisom su obuhvaćene sve aktivnosti koje doprinose održavanju vrijednosti proizvoda, kao što su instalacije, savjeti, popravci itd.

Za obavljanje primarnih aktivnosti nužne su potporne aktivnosti:

- Pribavljanje ulaznih resursa obuhvaća aktivnosti administracije prilikom naručivanja *inputa* za primarne aktivnosti.
- Razvoj tehnologije i proizvoda koji obuhvaća aktivnosti istraživanja i razvijanja procesa kojima se *input* pretvara u *output*. Razvijanje samog procesa može utjecati i na razvoj samog proizvoda.
- Upravljanje ljudskim potencijalima odnosi se na cjelokupne aktivnosti osiguravanja i poboljšavanja produktivnosti svih zaposlenika organizacije.
- Administracija i infrastruktura sadrži aktivnosti kao što su upravljački i rukovodeći sustavi te sustavi kontrole i financiranja primarnih aktivnosti.

Na taj način u svakoj su organizaciji određeni konkurentni procesi koji pripadaju nekom shematskom polju sa slike 2.8. Vrijednost IS-a u vrijednosnom lancu može se odrediti analizirajući vrijednosni lanac aktivnosti i njihova povezivanja. Na taj način ova se metoda koristiti za određivanje pogodnosti korištenja IS-a u svrhu poboljšanja interne vrijednosti lanca. Kao što je prije rečeno, Porter razdvaja aktivnosti kojima se dodaje vrijednost, tj. primarne aktivnosti, i one koje ih podupiru. Korištenje IS-a/IT-ja za primarne aktivnosti je pogodnost za ostvarenje konkurentske uspješnosti i utvrđivanje prioriteta informatizacije. U skladu s granicom (profit = dodana vrijednost

– trošak), IS može biti koristan u svakoj aktivnosti u kojoj može smanjiti trošak. Temeljeći odgovore na principima korištenja IS-a, sljedećim pitanjima moguće je procijeniti njegov doprinos u podizanju dodane vrijednosti:

- Može li se poboljšati tržišna vrijednost proizvoda?
- Postoji li mogućnost za smanjenje troška promatrane aktivnosti?
- Je li moguće eliminirati aktivnost u cijelosti?
- Može li se koristiti aktivnost u svrhu diferencijacije organizacije?

Na ovaj način analizu lanca vrijednosti moguće je koristiti pri strateškom planiranju informacijskih sustava.

### 2.3.5 Preustroj poslovne tehnologije

Preustroj poslovne tehnologije (engl. *business process reengineering*) metodološki je pristup provedbi organizacijskih promjena koji su krajem 90-ih godina 20. stoljeća popularizirali američki konzultanti M. Hammer i J. Champy s timom istraživača okupljenih oko T. Daveporta, profesora s Harvardske poslovne škole. Cilj ovog pristupa je radikalno promijeniti, tj. iz temelja redizajnirati poslovanje organizacije u svrhu znatnijeg poboljšanja njezine učinkovitosti. Promatranjem relevantnih indikatora poslovanja (veličina i struktura troškova, kašnjenje procesa, kvaliteta poslova itd.) odbacuje se mogućnost postupnog poboljšanja procesa i traži njihova radikalna promjena.

Posebna se pozornost pritom pridaje poslovnim procesima koji postaju temeljne jedinice strukturiranja novih procesnih organizacija. Na taj se način grupiranjem aktivnosti raspodijeljenih kroz tradicionalnu organizacijsku strukturu stvara organizacija s autonomnim operativnim poslovima, čime se zahtijeva i reorganizacija cjelokupnog sustava, uključujući i novu ulogu menadžmenta organizacije. Ovakvim pristupom prvo se pokušavaju identificirati potencijalni informacijsko-tehnološki napretci pa se tek onda traži mogućnost njihove implementacije u poslovne procese. Na taj način izbjegava se automatizacija postojećih zastarjelih procesa.

Prema Hammeru i Champyju, stvarna snaga tehnologije ne leži u mogućnostima poboljšavanja postojećih procesa nego u tome što omogućuje prevladavanje starih pravila i otvara mogućnost obavljanja poslovnih procesa na nov način. Hammer izdvaja sedam temeljnih principa transformacije poslovnog procesa pomoću reinženjeringa (Hammer, 1990):

- Posao je potrebno usredotočiti na rezultat, a ne na zadatak, tako da se izvođenje poslovnog procesa povjeri jednoj osobi koja će biti usmjerena prema obavljanju procesa u cjelini, a ne pojedinačnim koracima procesa.

- Proces izvode oni zaposlenici koji koriste rezultate procesa (samostalnost izvođenja), za razliku od tradicionalnih organizacija, u kojima se na temelju principa podjele rada pojedine organizacijske jedinice specijaliziraju za izvršavanje jedne vrste zadatka.
- Obrada informacija dodjeljuje se onima koji je generiraju, što znači da organizacijske jedinice i administrativne strukture specijalizirane za obradu postaju suvišne.
- Zemljopisno raspršene resurse treba tretirati kao da su centralizirani jer upotreba suvremenih informacijskih tehnologija omogućuje središnjici organizacije da raspolaže podacima iz dislociranih podružnica i obratno.
- Umjesto povezivanja parcijalnih rezultata paralelnih aktivnosti, pomoću IT-ja treba povezati njihovo obavljanje.
- Kontrolu i odlučivanje treba ugraditi u izvođenje poslovnog procesa, čime se postiže viši stupanj angažmana zaposlenika i korištenje njihovih potencijala.
- Informacije treba prikupljati samo jednom, i to na njihovu izvoru, što se konkretno provodi centraliziranim prikupljanjem podataka te njihovim pohranjivanjem u zajedničke baze podataka koje su putem informacijskog sustava dostupne svim zaposlenicima.

Na ovaj način postiže se preispitivanje svrhe, načina i mjesta obavljanja uobičajenih poslova u organizaciji, a sve na temelju vrijednosti za kupca. S tog se aspekta svi poslovi, prema Hammeru, dijele na aktivnosti i procese (Hammer, 1990):

- koji stvaraju dodanu vrijednost,
- koji ne dodaju vrijednost za potrošača i
- koji ne dodaju i ne podupiru stvaranje vrijednosti.

Na temelju prethodno iznesenog, moguće je zaključiti da je bit ovakva preustroja temeljito preispitivanje i korjenito preoblikovanje poslovnih procesa kako bi se najbrže i najjeftinije došlo do proizvoda koji zadovoljava potrebe kupaca.

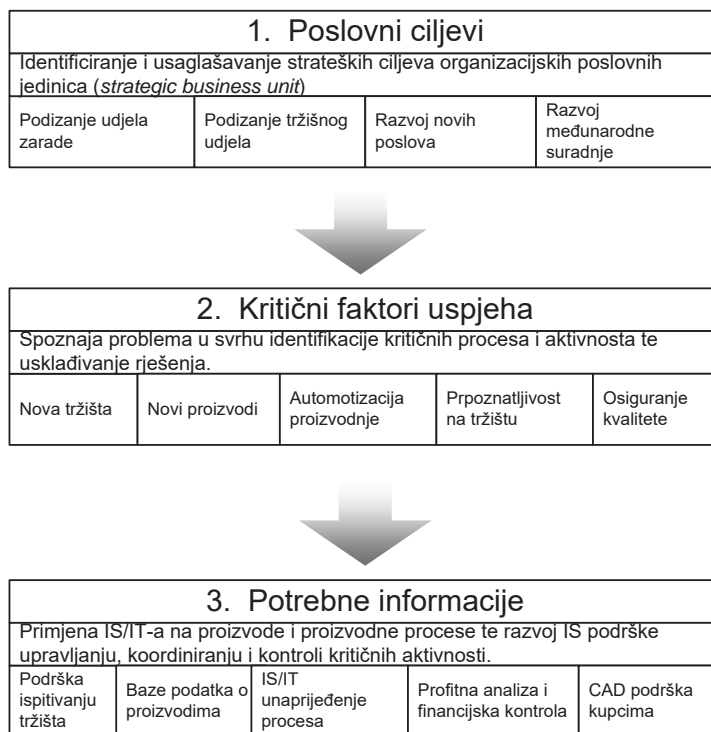
Prednost preustroja poslovnih procesa (PPP) na temelju načela IS-a/IT-ja ima tri glavne značajke:

- sloboda oblikovanja poslovnih procesa u okvirima fizičkih resursa,
- uvođenje procesa pomoću kojih organizacija upravlja promjenama primjenom IS-a,
- prilagođavanje novog IS-a u svrhu učinkovitosti post-PPP organizacije.

Prethodno upućuje na to da uspješna provedba PPP-a nije moguća bez upotrebe IS-a/IT-ja. Upravo u tom konceptu vidljiva je uloga PPP-a u procesu strateškog planiranja informacijskih sustava.

### 2.3.6 Metoda kritičnih faktora uspjeha

Metoda kritičnih faktora uspjeha CSF (engl. *critical success factors*) razvijena je na MIT-ju. Ideja metode je da se primjenom predloženih tehnika pomoću nekoliko intervjua uoče i definiranju potrebne informacije. Na slici 2.9 prikazana je poveznica kritičnih faktora uspjeha i zahtijevanih informacija u kontekstu primjene analize kritičnih faktora u okviru strateškog planiranja informacijskih sustava.

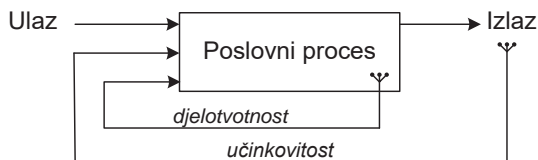


Slika 2.9 Primjena analize kritičnih faktora pri strateškom planiranju IS-a (Kaplan & Norton, 1996)

Upravo su kritični faktori uspjeha (CSF) faktori na koje se treba fokusirati, tj. faktori nužni za ostvarivanje uspjeha poslovne strategije. Na osnovi njih određuju se ključne poslovne odluke i informacije za njihovo donošenje za sve razine upravljanja. Na taj način definira se uloga IS-a, odnosno što IS mora omogućiti.

### 2.3.7 Analiza *End-means*

Analiza *End-means* temelji se na teoriji sustava. Modelom povratnih veza prvog reda moguće je odrediti odstupanja od zadane djelotvornosti rada, dok se modelom povratnih veza drugog reda mogu odrediti odstupanja od zadane učinkovitosti poslovanja, što je strukturno moguće prikazati slikom 2.10.



Slika 2.10 Povratne veze analize *End-means* (Robson, 1997)

Navedeno znači da je za svaki proces potrebno definirati izlazne (*Ends*) i ulazne (*Means*) zahtjeve te informacije za postizanje djelotvornosti rada i učinkovitosti poslovanja. Izlaz jednog procesa je ulaz za sljedeći proces. Osnovni su koraci analize:

- specifikacija ulaza;
- specifikacija izlaza;
- specifikacija potrebnih informacija o djelotvornosti, tj. pokazatelja djelotvornosti organizacije;
- specifikacija potrebnih informacija o učinkovitosti, tj. pokazatelja učinkovitosti organizacije.

Primjenjujući navedene korake, ova se analiza koristi u pristupu reorganizacije poslovnih procesa i određivanja potrebnih informacija. U tom kontekstu može se koristiti i za definiranje važnosti IS-a/IT-ja pri strateškom planiranju informacijskih sustava.

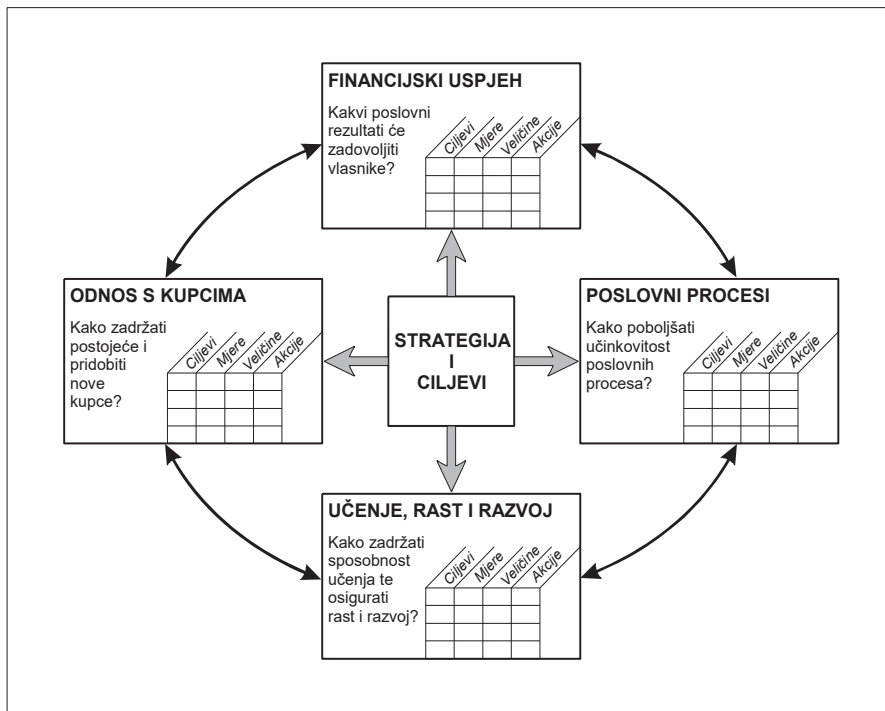
## 2.4 Mjerenje učinaka IS-a/IT-ja prema metodi *balanced scorecard*

Metodu originalnog naziva *balanced scorecard* (BSC) izvorno su 1992. postavili Kaplan i Norton da se poduzećima omogući definiranje strategije razvoja i praćenje uspjeha provedbe te strategije (Kaplan & Norton, 1992; Gibbons & Kaplan, 2015). Razvoj BSC-a temelji se na empirijskom iskustvu velikog broja organizacija kako bi se otklonio nedostatak mjerenja učinkovitosti isključivo financijskim pokazateljima. Koncept metode BSC-a prikazan je na slici 2.11.

Za razliku od prethodnih metoda, BSC-om se predlaže tzv. "uravnoteženi bodovni sustav" kojim se ukupna uspješnost organizacije promatra kroz četiri različite perspektive. To su:

- financijski uspjeh;
- odnos s kupcima;
- poslovni procesi;
- učenje, rast i razvoj.

Svaka od perspektiva predstavlja se skupom pokazatelja promjena (engl. *lead indicators*) koje je potrebno vrijednosno opisati kako bi se mogla mjeriti postignuća aktivnosti kojima se pokreću promjene (engl. *drivers*). Na temelju ustanovljenog uzročno-posljedičnog djelovanja trebali bi se postići zadani, također mjerljivi ciljevi kojima se prati razina postignutih izlaznih vrijednosti pokazatelja stanja (engl. *lag indicators*). Za svaki indikator obično se definira donja i gornja teoretska granična vrijednost te ciljana i najniža prihvatljiva vrijednost. Koncept BSC-a prikazan je na slici 2.11.



Slika 2.11 Koncept metode balanced scorecard (Kaplan & Norton, 2014; Kaplan & Norton, 1996)

Financijski uspjeh mjeri se dosadašnjim klasičnim pristupom, tj. tradicionalnim financijskim pokazateljima kao što su: profit, stopa povrata kapitala, prihod po zaposlenom, porast prodaje itd.



Indikatori iz perspektive odnosa s kupcima utvrđuju se tako da se analizira što učiniti da se zadrže postojeći, odnosno pridobiju novi kupci. Tako se, primjerice, kod roba široke potrošnje najčešće provode ankete. Obradom rezultata ankete ocjenjuje se privrženost i zadovoljstvo kupaca koje se može izraziti bodovnom skalom i omjerom izgubljenih ili pridobivenih kupaca.

Indikatori perspektive poslovnih procesa predstavljaju sveukupnu organiziranost organizacije. Mjerene veličine iz ove grupe ukazuju na sveukupnu organiziranost sustava. Mogu se mjeriti vrijednosti poput transakcijskih troškova, koeficijenta obrtaja zaliha, vremena odaziva na zahtjeve, iskorištenosti kapaciteta itd. Ciljevi koji se trebaju ostvariti u ovom segmentu mjerenja moraju biti u skladu s poboljšanjem učinkovitosti poslovnih procesa organizacije.

Učenje, rast i razvoj omogućit će inteligentno prilagođavanje organizacije promjenama iz okoline te na taj način osigurati daljnji rast i razvoj. U tom kontekstu treba se osigurati zadržavanje sposobnosti učenja te rasta i razvoja organizacije. Ova perspektiva obično se mjeri brojem proizvodnih, tehnoloških i poslovnih inovacija, novih proizvoda i usluga itd.

Prilikom određivanja ciljeva i indikatora treba voditi računa radi li se o profitnim ili neprofitnim organizacijama. Naime, strateški se ciljevi neprofitnih i profitnih organizacija razlikuju pa bi zato trebalo očekivati različite sustave mjerenja.

Autori Martinson et al. sugeriraju menadžerima primjenu BSC-a za pomoć prilikom procjene ulaganja u IT (Martinsons, Davidson, & Tse, 1999). Velika financijska ulaganja u IS traže jasno objašnjenje njihove isplativosti. Šarenilo izvješća o isplativosti ulaganja u IS ukazuje na teškoću procjene doprinosa IS-a u poslovanju organizacije. Razlozi su metodološke, administrativne i metričke prirode. Pođemo li od stajališta da se IS i njegovo planiranje mora promatrati kao podsustav organizacijskog sustava, a ne sustav koji postoji sam za sebe, nameće se zaključak da se i mjerenje učinaka IS-a mora promatrati kroz učinke – ne samo financijskog karaktera nego i ostalih pokazatelja učinkovitosti organizacije u cjelini.

Jedan od pristupa, pod nazivom *information economics*, predlaže da se prilikom procjene ulaganja u razvoj IS-a/IT-ja, osim financijskih pokazatelja, uzima u obzir i stupanj konkurentnosti i potpore krajnjim korisnicima. Također se predlaže podjela promatranja rizika na dvije domene – poslovnu i tehnološku. Međutim, ni te dvije domene ne obuhvaćaju sve učinke koje su posljedica korištenja novih IS-a/IT-ja (Martinsons, Davidson, & Tse, 1999).

Osnovna ideja za proširenje pogleda na organizaciju proizlazi iz perspektive koncepta pod nazivom BSC za IS (Martinsons, Davidson, & Tse, 1999). Naime, BSC za IS preporučuje perspektive i ciljeve kojima se treba ustanoviti i mjeriti učinak IS-a/IT-ja na ostvarenje poslovnih ciljeva organizacije. Upravo ti pogledi na organizaciju, u kombinaciji s prethodnom primjenom SWOT analize za procjenu učinaka IT-ja, primjenjivi su u koracima razvijenog modela za utvrđivanje IS/IT strategija i aktivnosti za njihovo ostvarenje.

Predloženi koncept BSC za IS sličan je klasičnom konceptu BSC. Osnovne ideje za preoblikovanje perspektiva BSC-a proizlaze iz sljedećih tvrdnji:

- IS projektom istovremeno se ostvaruje pogodnost za krajnjeg korisnika i organizaciju u cjelini, a ne samo za pojedinačnog kupca/korisnika.
- IS odjel treba promatrati kao internog, a ne eksternog pružatelja usluga.

Sukladno tome, perspektive za mjerenje performansi IS-a jesu:

- orijentiranost korisnicima (krajnji korisnici),
- poslovne vrijednosti,
- interni procesi,
- spremnost za budućnost.

Primarni ciljevi IS-a dijele se na dva tipa: ciljevi djelotvornosti i ciljevi učinkovitosti. Ciljevi djelotvornosti odnose se na procese, stoga ih je potrebno promatrati kroz perspektivu internih procesa. Ciljevi učinkovitosti odnose se na korisnike, zbog čega se promatraju kroz perspektivu orijentiranosti korisnicima i perspektivu poslovnih vrijednosti. Perspektivom spremnosti za budućnost, prepoznajući potrebu inovacija i učenja, obuhvaćene su nužne tehnologije te poslovne prilike i izazovi za osiguranje stabilnosti rasta i razvoja.

Procjenu stanja u kontekstu poslovne vrijednosti IS-a/IT-ja moguće je podijeliti na dvije kategorije:

- kratkoročni doprinos, koji se obično odnosi na pojedinačne IS/IT projekte i
- dugoročni pogled, koji još uključuje i cjelovitu funkciju IS-a.

To znači da se za upravljanje proračunom IS-a i njegovim direktnim, lako kvantificiranim koristima, npr. u obliku uštede automatizacijom poslovanja, koriste klasični financijski pokazatelji. Međutim, IS/IT može generirati poslovne vrijednosti na druge načine koji imaju utjecaj na ostvarenje različitih ciljeva. Tako se, primjerice, korištenje IS-a/IT-ja u strateškom upravljanju, u obliku sustava za potporu strateškom upravljanju, može odraziti na učinkovitosti organizacije i smanjenje rizika poslovanja.

Iz tog razloga je indikatore i ciljeve, u pogledu poslovnih vrijednosti, potrebno tražiti u kontekstu trenutnog stanja:

- proračuna IS-a/IT-ja (npr. raspodjela prema ostalim proračunskim stavkama, odnos proračuna IS-a i ukupnog prihoda, trošak IS-a prema zaposleniku...);
- ostvarenosti prihoda od proizvoda i usluga podržanih IS-om/IT-jem;
- poslovnih vrijednosti IS/IT projekata i području IS odjela/funkcionalnosti (npr. tradicionalni financijski pokazatelji, vrijednost povezivanja i ubrzavanja, tehnološke inovacije...);
- rizika (npr. uspjelost IS strategija, tehnološki rizik...);
- poslovne vrijednosti IS odjela ili tima (npr. resursi uloženi u strateške projekte, komunikacija osoblja IS-a i menadžmenta, međusobno razumijevanje...).

Iz perspektive orijentiranosti korisnicima općenito se može reći da se vrijednost IS-a očituje u povećanju djelatnosti i učinkovitosti izvršenja njihovih poslova. Postotak procesa poduprtih informatičkim tehnologijama i orijentiranih krajnjim korisnicima, bez obzira radi li se o unutarnjim ili vanjskim korisnicima, specifičan je i ovisi o djelatnosti organizacije. Iz tog razloga se prilikom određivanja elemenata SWOT-a iz perspektive orijentiranosti korisnicima potrebno usredotočiti na područja odabira postojećih usluga za primjenu i operacionalizaciju te uspostavljanje i održavanje odnosa i zadovoljstva krajnjih korisnika.

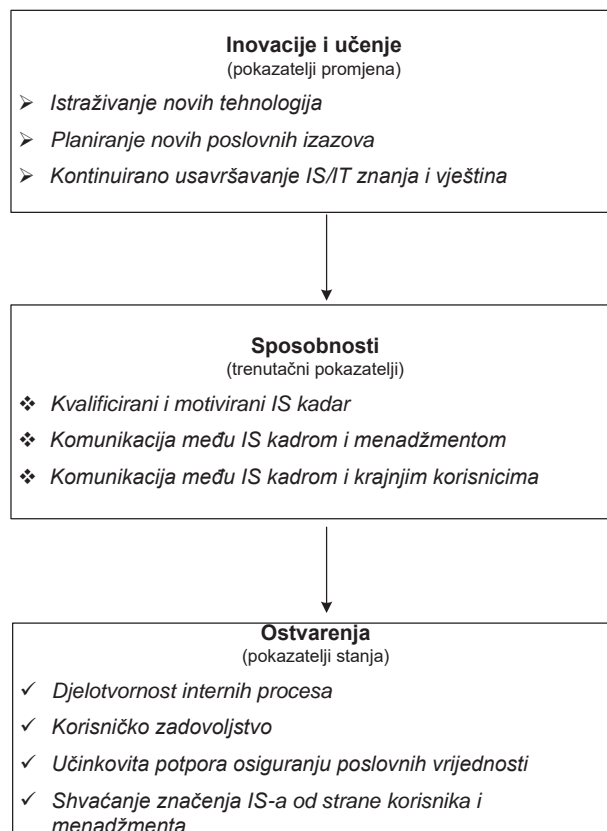
Smjernice za utvrđivanje elemenata SWOT-a iz perspektive unutarnjih procesa moguće je izvesti pomoću tri osnovna procesa u izvedbi odjela IS-a/IT-ja:

- planiranje i prioritet IS/IT projekata (npr. resursi za planiranje i praćenje aktivnosti IS-a...);
- razvoj novih IT aplikacija (npr. resursi za razvoj novih aplikacija, vrijeme za postizanje standardizacije njihove primjene, postojanje aplikacija s ponovnim korištenjem koda, utrošak resursa za popravke i fino prilagođivanje novih aplikacija...);
- rad i održavanje postojećih IT-jeva (npr. broj obrađenih upita krajnjih korisnika, prosječna brzina otklanjanja korisničkih problema...).

Osim trenutnog načina korištenja IS-a/IT-ja potrebno je procijeniti spremnost organizacije za korištenje IT-ja koji će se tek pojaviti u budućnosti. To znači da odjel IS-a treba biti u stanju prepoznati mogućnosti primjene IT-ja. Na taj način bit će moguće postaviti strategije za osposobljavanje organizacije za uočavanje primjene i prihvaćanje novih IT-ja. To znači da u pogledu spremnosti na budućnost treba utvrditi indikatore i ciljeve koji će u kontekstu rizika korištenja IS-a/IT-ja omogućiti otklanjanje

nedostataka organizacije i stvaranje pogodnosti, što treba osigurati dugoročnu stabilnost razvoja organizacije. U tu svrhu indikatore i ciljeve potrebno je tražiti u kontekstu:

- kontinuiranosti usavršavanja vještina IS-a/IT-ja IT stručnjaka za razvoj novih primjena IT-ja i ostalih zaposlenika za njihovo korištenje (npr. odnos sredstava za profesionalno usavršavanje i ukupnog proračuna IS-a/IT-ja, stručnost za specifične i buduće IT-jeve bitne za poslovanje, prosječna starost osoblja IS-a, usavršavanje IT-ja kod ostalih zaposlenika);
- redovitosti osvježavanja aplikacijskog portfelja (npr. tehničke performanse aplikacijskog portfelja, korisničko zadovoljstvo aplikacijskim portfeljem...);
- važnosti istraživanja prijeko potrebnih tehnologija i njihova značenja za poslovanje (npr. odnos istraživačkog i ukupnog proračuna IS-a/IT-ja, zadovoljstvo vodećeg menadžmenta s izvještavanjem o mogućnostima primjene novih IT-ja...).



Slika 2.12 Utjecaj inovacija i učenja na buduća ostvarenja organizacije (Martinsons, Davidson, & Tse, 1999)

Utvrđivanjem utjecaja među strateškim ciljevima ostvaruje se povezivanje strateških ciljeva s različitim pogledima na organizaciju. Na taj način objedinjuju se dugoročni i kratkoročni IS/IT i poslovni ciljevi organizacije u svrhu ostvarenja zadanih strateških ciljeva. Primjer utjecaja dugoročnih ciljeva u kontekstu IS/IT strategija na buduća ostvarenja organizacije prikazan je na slici 2.12.

Kontinuiranim usavršavanjem IS/IT znanja i vještina te istraživanjem novih tehnologija dobivaju se nova znanja kojima se postižu kompetencije IT kadra. Planiranjem primjene IT-ja u svrhu rješavanja poslovnih izazova, IS/IT stavlja se u kontekst usluge korisnika i menadžmentu razumljivih okvira poslovne strategije. Na taj način ostvaruje se odlična komunikacija IS kadrova i krajnjih korisnika te menadžmenta. Ovakve organizacijske sposobnosti u budućnosti trebaju rezultirati djelotvornošću poslovnih procesa, korisničkim zadovoljstvom i učinkovitom potporom osiguranju poslovnih vrijednosti. To u konačnici znači da korisnici i menadžment shvaćaju važnost IS-a.

## 2.5 Literatura

1. Alamri, S., Almutiri, N., Ballahmar, H., & Zafar, A. (2016). Strategic Information System Planning: A Case Study of a Service Delivery Company. *International Advanced Research Journal in Science, Engineering and Technology* Vol. 3, Issue 5, 78–84.
2. Bernroider, E. (2002). Factors in SWOT Analysis Applied to Micro, Small-to-Medium, and Large Software Enterprises: An Austrian Study. *European Management Journal* Vol. 20, No. 5, pp. 562–573;.
3. Brumec, J., Dušak, V., & Vrček, N. (2000). The Strategic Approach to ERP Design and Implementation. *Journal of Information and Organisational Science* 24(1), str.15–25.
4. Dahlberg, T., Kivijärvi, H., & Saarinen, T. (2015). *The Role of IT Investment Consistency among the Enablers behind the Success of IT Deployment*. System Sciences, HICSS 48.
5. Gibbons, R., & Kaplan, R. S. (2015). Formal Measures in Informal Management: Can a Balanced Scorecard Change a Culture? *American Economic Review : Papers & Proceedings*, 447–451.
6. Hammer, M. (1990). *Reengineering Work: „Don't Automate, Obligate“*. Harvard Business Review.
7. Hill, T., & Westbrook, R. (1997). SWOT Analysis: It's Time for a Product Recall. *Long Range Planning*, Vol. 30, No. 1, pp. 46–52.
8. Kaplan, R., & Norton, D. (1992). *The Balanced Scorecard - Measures that Drive Performance*. Harvard Business Review.
9. Kaplan, R. S., & Norton, D. P. (1996). *Using the balanced scorecard as a strategic management system*.
10. Kaplan, R., & Norton, D. (2014). *Organizational capital: Supporting the change agenda that supports strategy execution*. Harvard Business School Publishing.
11. Martinsons, M., Davidson, R., & Tse, D. (1999). The Balanced scoredcard: A fudation form the Strategic Management of Information System. *Decision Support System* 25, str. 71–88.
12. Porter, M. (1985). *Competitive Advantage: Creating and Sustaining Superior Performance*. New York: The Free Press.
13. Robson, W. (1997). *Strategic Management & Information Systems*. NY: Prentice Hall.
14. Stacey, D. R. (2007). *Strategic Management and Organisational Dynamics*. NY: Prentice Hall.

15. Ward, J., & Peppard, J. (2002). *Strategic Planning for Information Systems*. John Wiley and Sons.
16. Ward, J. & Griffiths, P. (1996) *Strategic Planning for Information Systems*. 2nd edn. Wiley.
17. Wehrich, H., & Koontz, H. (1993). *Management*. Victoria Publishing.





## 3 Upravljanje provedbom strategije razvoja informacijskih sustava

Cilj ovog poglavlja je prikazati matematički dokazan postupak upravljanja provedbom strategije IS-a/IT-ja i njegovu konkretnu primjenu. Sam proces upravljanja, osim faze planiranja, sadrži i faze izvođenja i kontrole. To je ciklus koji se periodično ponavlja. Ovdje prikazan proces upravljanja provedbom strategije IS-a/IT-ja polazi od definiranja poslovnog okvira u kojima se može tražiti utjecaj IT-ja.

Vodeći element u sustavu vrijednosti organizacije jasno je formulirana, sažeto izrečena i javno objavljena misija organizacije. Ovako prezentirana misija je poruka o poslovnoj usmjerenosti organizacije svim svojim zaposlenicima, dobavljačima, kupcima te korisnicima njihovih proizvoda i usluga. Kao takva, misija je sredstvo komunikacije unutar organizacijskih okvira, ali i njezine okoline. Kvaliteta ostvarenja misije ovisi o uspjehu strateškog upravljanja organizacijom. Zbog toga je misija opći kriterij prema kojem se mjeri uspješnost strateškog upravljanja organizacijom.

### 3.1 Planiranje učinaka razvoja IS-a na poslovni sustav

#### 3.1.1 Prva priča iz konkretnog života za lakše razumijevanje poglavlja

Prema teoriji sustava, sustav je uređena trojka (funkcija - F, struktura - S i elementi - E). U tom kontekstu, poslovni sustav ima svoju funkciju (ostvarenje poslovnih ciljeva), strukturu (poslovnu tehnologiju, tj. skup poslovnih pravila) i elemente (entitete poslovnog sustava koji u međusobnoj interakciji, sukladno poslovnim pravilima/strukтури, ostvaruju poslovne ciljeve/funkciju). Nadalje, podsustav je sustav niže razine koji možemo promatrati kao zasebnu cjelinu unutar nekog sustava. To znači da grupiranjem entiteta sukladno njihovoj ulozi možemo formirati podsustave nekog poslovnog sustava.

S druge strane, model je sa znanstvenog aspekta skup pretpostavki pomoću kojih se teorijski opisuje neki sustav. On je uvijek pojednostavljenje realnog sustava, što znači da se prilikom modeliranja zanemaruju određeni detalji koji nisu potrebni za opisivanje ostvarenja funkcije koju se želi modelirati. Ovo nije jednostavan postupak jer se ne smije dogoditi da se zbog pojednostavljenja izgube neki naizgled nebitni elementi.

Iz tog se razloga koriste različite metode i tehnike, tj. skup pravila na koji se način izrađuju modeli. U skladu s prethodnim, u ovom se poglavlju kreće od funkcije poslovnog sustava, tj. od poslovnih ciljeva, i to dugoročnih, odnosno strateških ciljeva. Naime, funkcija je poslovnog sustava dinamična, što znači da se ostvaruje kroz vrijeme

i kroz niz ostvarenja prethodnih strateških ciljeva (lanca strateških ciljeva). Pod pojmom strateških ciljeva podrazumijevaju se dugoročni ciljevi koji se ostvaruju tijekom vremena (dinamika) zbog poduzimanja aktivnosti koje dovode do ostvarenja parcijalnih ciljeva. Primjerice, da bi ostvarili bolju plaću, zadovoljstvo u radu, mogućnost napredovanja i sl., tj. sve ono što opisuje bolje radno mjesto, studenti unapređuju vlastito znanje i stežu radno iskustvo. Čin diplomiranja (jedan od zadanih strateških ciljeva u periodu studiranja) događa se sam po sebi, nakon što se polože svi ispiti (poduzimanjem niza aktivnosti).<sup>1</sup>

Međutim, sam čin diplomiranja i dalje ne garantira ostvarenje „snova“. To je samo jedna od stepenica u ostvarenju vizije. U biti, što je vaša vizija? Koliko je točno možete iskazati? Čest je slučaj da studenti prilikom izbora završnog rada nisu sigurni u izbor teme. Moje je prvo pitanje, kao nastavnika: „Što biste htjeli raditi kad završite fakultet?“ Isto pitanje postavite i sebi. Vidjet ćete koliko je to teško odgovoriti. Naravno, ne znate koji se sve poslovi nude, kako se kreću plaće... (vanjski elementi SWOT-a ili pSWOTa iz prethodnog poglavlja). S druge strane, tijekom studija upoznajete se s različitim područjima, metodama, tehnikama, susrećete se s različitim izazovima, pa i nekim dosadnim stvarima pitajući se: „Što će to meni? Što će meni sva ta matematika iz ovog poglavlja?“ (unutarnji elementi SWOT-a ili pSWOTa).

Vratimo se sada na viziju (želju) budućeg zaposlenja. Pokušajte isti algoritam ovog poglavlja (računalni model) primijeniti na ostvarenje vašeg životnog poziva. Prva je stvar kvantificirati pojam željenog zanimanja, npr. kolika je to plaća koja vas zadovoljava, što to znači mogućnost napredovanja koja vas zadovoljava... To je ono što vi zadajete kao vlasnici svoga izbora. Iz tog razloga to zovemo *zadani strateški ciljevi*. Oni su proizašli iz vaše vizije za dobrim radnim mjestom, npr. *imati plaću 10000 kn (BSC perspektiva financija) tako da završim fakultet s minimalnim prosjekom 4,0 (BSC perspektiva rast i razvoj) i uspješno se predstavim 30 potencijalnih poslodavaca (BSC perspektiva odnos s poslodavcima)*. Kako biste definirali kako završiti fakultet sa željenim prosjekom i kako se prezentirati potencijalnim poslodavcima, trebate napraviti pSWOTa i izvesti nove strateške ciljeve. S obzirom na to da se oni izvode određenim postupkom, zovu se *izvedeni strateški ciljevi*. Oni su rezultat poduzetih aktivnosti, npr. povećati broj sati učenja za 20 %, povećati vidljivost svojih sposobnosti korištenjem društvenih mreža (10 % veći broj lajkova potencijalnih poslodavaca na moje objave). Iako postoje i drugi načini promocije, namjerno sam ubacio društvene mreže kao primjer korištenja IS-a/IT-ja prilikom definiranja strateškog plana u segmentu poslovnih procesa. Time smo obuhvatili sve četiri klasične perspektive (pogleda na sustav) metode BSC. Ostvarenje izvedenih strateških ciljeva dovest će vas sve bliže vašoj viziji predloženoj zadanim strateškim ciljevima. Jednako tako, vaš je

---

<sup>1</sup> Pogledati razliku među *lagging* i *leading* indikatora metode BSC iz prethodnog poglavlja.

ZSC imati 10000 kn plaću mjesečno i to je zadano, ali pitanje je hoće li to biti moguće odmah kod prvog zaposlenja. To ne znači da ste promijenili ZSC, samo ste produljili vrijeme ostvarenja strateškog plana. To može biti uzrokovano i drugim različitim vanjskim uvjetima (prijetnje prema SWOT-u) na koje ne možete utjecati (npr. pandemija COVID-19). To ne znači da mijenjate strategiju nego njezinu implementaciju, tj. strateški plan. Iz tog je razloga bitno da mjerite sve parametre (ciljeve) ostvarenja strateškog plana. Nije rijetka zabluda studenata koji ne uče cijelu godinu da će to riješiti u ispitnom roku. Da naprave ovakve analize, vrlo brzo bi došli do zaključka da to nije moguće. S obzirom na to da su strateški ciljevi teško mjerljivi jer je praktički nemoguće utvrditi točne vrijednosti, na kraju strateškog ciklusa zadovoljavamo se utvrđivanjem ispravnosti trendova. Drugim riječima, predviđene poduzete aktivnosti trebaju pozitivno utjecati na ostvarenje sljedećih strateških ciljeva u lancu, npr. završetak preddiplomskog studija s prosjekom 4,0 je 3/5 ostvarenja zadanog strateškog cilja „diplomiranje“.

Modeliranjem utjecaja među strateškim ciljevima dobivamo hijerarhijsku mrežu strateških ciljeva. Reprezentant hijerarhijske mreže je matrica. Na taj način problem upravljanja strateškim ciljevima svodimo na problem matričnog računa. Dokaz mogućnosti svođenja hijerarhijske mreže na matrični račun prikazan je u poglavlju 3.3, prije čega će biti navedena druga priča iz života za lakše razumijevanje poglavlja.

### 3.1.2 Postavljanje zadanih strateških ciljeva

Ovdje prikazan osnovni koncept strateškog upravljanja razvojem IS-a zasniva se na pretvaranju strateških zamisli vlasnika i uprave u operativne aktivnosti na provedbenim organizacijskim razinama. Formalizirani oblik strateških zamisli vlasnika u obliku organizacijske vizije transformira se u zadane strateške ciljeve organizacije. Provođenjem svih tipova (IS-a i poslovnih) aktivnosti, organizacija bi trebala kroz buduće razdoblje iz trenutnog stanja (*as is*) postići željeno stanje (*to be*), tj. ostvariti zadane strateške ciljeve.

Dinamično okruženje u kojem organizacija djeluje mijenja se za trajanja ili odgode planiranih aktivnosti. Zbog toga je sposobnost sustavnog usklađivanja strateškog plana razvoja, kako IS-a tako i poslovanja, u novonastalim okolnostima preduvjet za uspješno ostvarenje zadanih strateških ciljeva. Ovakav pristup zahtijeva da proces strateškog upravljanja razvojem, osim planiranja i organiziranja, sadrži i kontrolu razine ostvarenosti svih strateških ciljeva. Iz tog razloga je potrebno sustavno mjerenje učinkovitosti i djelotvornosti<sup>1</sup> organizacije te trajanja strateškog ciklusa. Redefiniranje

---

<sup>1</sup> Učinkovitost (engl. *effectiveness*, sinonim: efektivnost) je postizanje ciljeva. Djelotvornost (engl. *efficiency*, sinonim: efikasnost) je postizanje ciljeva s najmanjom količinom resursa. U kontekstu ovog rada, povećanje djelotvornosti promatra se povećanjem učinkovitosti. Ovo proizlazi iz činjenice da se povećanjem učinkovitosti uz istu količinu resursa povećava i djelotvornost organizacije.

postojećeg strateškog plana odvija se u skladu s izmjerenim učincima koji se postižu provođenjem ciljano usmjerenih aktivnosti.

Za razliku od misije koja ima ugrađen trajni smisao postojanja organizacije, vizija je buduća slika organizacije iz perspektive vlasnika. Na početku novog ciklusa strateškog planiranja menadžment predlaže viziju koju potom daje na usvajanje vlasniku. Vlasnik organizacije prihvaća prijedlog vizije ili ga šalje na doradu. Dorađena vizija ponovno prolazi postupak usvajanja vlasnika.

Usvojenu viziju organizacije potrebno je transformirati u ciljeve.<sup>1</sup> Ciljevi izvedeni iz vizije organizacije pripadaju strateškoj razini i određuju *to be* stanje organizacije. Ovi ciljevi s organizacijskog stajališta su zadani strateški ciljevi (ZSC). Određeni ZSC raspoređuju se unutar perspektiva BSC na kojima se mjeri ukupna učinkovitost organizacije. Forma zapisa ZSC-a u koju se transformiraju stavovi sadržani u viziji mora sadržavati sljedeće semantičke elemente:

- oznaka cilja – sastoji se od velikog prvog slova pripadajuće perspektive BSC<sup>2</sup> i indeksa cilja u promatranoj perspektivi (npr. ako su dva cilja raspoređena u financijsku perspektivu, njihove oznake su F1 i F2, dok bi za ciljeve iz korisničke perspektive oznake bile K1 i K2);
- smjer promjene – označuje smjer promjene koji je potrebno ostvariti (smanjiti ili povećati);
- objekt promjene – opis objekta nad kojim se želi ostvariti promjena, a ujedno i objekt mjerenja (npr. profit, troškovi, broj korisnika, prosječan broj žalba po radnom mjestu...);
- naziv mjere – opisni naziv mjere koja prikazuje razinu ostvarenosti cilja;
- oznaka mjere – formira se tako da se prije oznake cilja dopiše slovo m (oznaka cilja);
- *as is* – vrijednost mjere kojom je određena početna razina ostvarenosti cilja;
- *to be* – vrijednost mjere kojom je određena željena razina ostvarenosti cilja;
- mjerna jedinica.

Prethodno navedenu formu moguće je prikazati u obliku tablice 3.1, u kojoj je naveden i primjer zapisivanja ZSC-a.

---

<sup>1</sup> Pod pojmom "cilj" podrazumijeva se kvantificirana vrijednost koja se mjeri nekom mjerom, a nastoji se postići tijekom određenog razdoblja.

<sup>2</sup> Za izvedene strateške ciljeve koristi se malo prvo slovo pripadajuće perspektive BSC.

Tablica 3.1 Tablica zadanih strateških ciljeva

Oznaka cilja	Smjer promjene	Objekt promjene	Naziv mjere	Oznaka mjere	As is	To be	Mjerna jedinica
F1	Povećati	Prihod od novog proizvoda	Udio prihoda od novog proizvoda u ukupnom prihodu	mF1	10	20	%

Svaki redak tablice opisuje jedan ZSC, a u skladu s navedenim primjerom čita se: „Cilj F1: Povećati prihod od novog proizvoda tako da udio prihoda od novog proizvoda u ukupnom prihodu (mF1), umjesto početnih 10 %, na kraju perioda bude 20 % .“

Ovakvom formalizacijom ZSC-a menadžment kvantitativno konkretizira viziju. Na taj način definirane su koordinate odredišta u kojem se organizacija treba naći nakon planiranog razdoblja za koje se ciljevi definiraju. Prijedlog ZSC-a menadžment predaje na odobrenje vlasnicima. Odobrenjem prijedloga vlasnici potvrđuju usklađenost ZSC-a s vizijom organizacije. Sve buduće aktivnosti organizacije i ostali strateški ciljevi moraju biti usmjereni prema ostvarenju ZSC-a. Zbog toga je ZSC ključan za organizacijski sustav u cjelini.

Sljedeći koraci u razvoju modela strateškog upravljanja odnose se na utvrđivanje načina ostvarenja svih ZSC-ova. Cjelokupni postupak zahtijeva dobro poznavanje temeljne djelatnosti organizacije i okoline u kojoj ona djeluje.

### 3.1.3 Strukturiranje učinaka razvoja IS-a/IT-ja na poslovni sustav

Polazeći od analize trenutnog stanja organizacije, menadžment procjenjuje spremnost organizacije za dostizanje ZSC-a. Na temelju provedene analize stanja određuju se strategije, konkretne aktivnosti za njihovu provedbu te ostali strateški ciljevi preko kojih će se realizirati ZSC. Na ovaj način, ostali strateški ciljevi izvode se iz ZSC-a, zbog čega se nazivaju *izvedeni strateški ciljevi* (ISC).

Postupak utvrđivanja ISC-a jednim se dijelom zasniva na konceptu metode pSWOTa. Ovdje prikazan postupak podijeljen je u četiri logički povezana koraka, čiji se rezultati prikazuju u formi jedne tablice.

**Prvi korak je određivanje elemenata SWOT-a.** Za svaki zadani strateški cilj potrebno je utvrditi elemente SWOT analize: prednosti (S), nedostatke (W), prilike (O) i prijetnje (T). Kao i kod pSWOTa, općenito se preporučuje da se za način ostvarenja svakog ZSC-a utvrde do tri (najviše četiri) elementa SWOT-a.

Metoda pSWOTa pretpostavlja da se za svaki ZSC izradi jedna tablice pod nazivom *Polazni elementi SWOT analize*. Iako se za različite ZSC-ove najčešće utvrđuju različiti

elementi SWOT-a, naravno da postoji mogućnost da se isti element SWOT-a ponovi kod dvaju ili više ZSC-ova.

Utvrđene elemente svih ZSC-ova upisuju se u jednu tablicu. Zbog sistematičnosti predlaže se popisivanje elemenata SWOT-a prema kategorijama (S, W, O, T) bez obzira na zadani strateški cilj na temelju kojeg su utvrđeni. Svaki utvrđeni element SWOT-a potrebno je jasno opisati i označiti tipom i indeksom ( $S_i$ ,  $W_j$ ,  $O_k$  ili  $T_l$ ), gdje je  $i=1, \dots, s$ ,  $j=1, \dots, w$ ,  $k=1, \dots, o$  i  $l=1, \dots, t$ . Utvrđene elemente moguće je zapisati u obliku tablice 3.2.

Tablica 3.2 Tablični prikaz elemenata SWOT-a

Opis $S_1$	...	Opis $S_s$	Opis $W_1$	...	Opis $W_w$	Opis $O_1$	...	Opis $O_o$	Opis $T_1$	...	Opis $T_t$
$S_1$	...	$S_s$	$W_1$	...	$W_w$	$O_1$	...	$O_o$	$T_1$	...	$T_t$

**Drugi korak je definiranje strategija IS-a/IT-ja** koje će omogućiti ostvarenje ZSC-a. Definiranje IS i IT strategije<sup>1</sup> radit će se na temelju elemenata SWOT-a iz tablice 3.2. Za formalno označavanje strategija koristi se sintaksa, kao npr.  $DS_1^{\circ}DT_1$ , što znači da se prednost označena sa  $S_1$  koristi za otklanjanje prijetnje označene s  $T_1$ , dok  $DO_1^{\circ}DO_2^{\circ}DW_1$  znači da se dvije vanjske pogodnosti označene s  $O_1$  i  $O_2$  koriste za otklanjanje slabosti označene s  $W_1$ . Oznaka D (dio) označuje mogućnost odnosa 1:n (utjecaj jednog elemenata SWOT-a na više drugih elemenata) i  $m:1$  (kombiniranje više elemenata SWOT-a za ostvarenje utjecaja na jedan element SWOT-a). Elementi SWOT-a zbog kojih se definira strategija (O, W ili T) nazvani su odredišni elementi (u primjerima iz prethodne rečenice to su elementi SWOT-a označeni s  $T_1$  i  $W_1$ ). Prema pSWOT-u, strategije općenito mogu biti korektivne, agresivne ili obrambene.

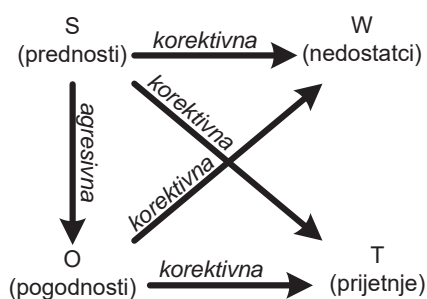
Ovdje korištena logika određivanja korektivnih strategija sadržana je u odgovoru na pitanje: Što je moguće poduzeti kako bi se poništile/umanjile slabosti (W) ili prijetnje (T), s obzirom na prednosti (S) i pogodnosti (O)? Zbog toga korektivna strategija ima dijelove tipa  $(DS_i^{\circ}DW_j)$ ,  $(DS_i^{\circ}DT_l)$ ,  $(DO_k^{\circ}DW_j)$ , odnosno  $(DO_k^{\circ}DT_l)$ .

Korektivne strategije nisu dovoljne za ostvarenje ZSC-a. Preduvjete za ostvarenje ZSC-a potrebno je tražiti u prednostima (S) po kojima se organizacija razlikuje od svoje

<sup>1</sup> Krivulja iskustva bazira se na promatranju organizacije i njezine efikasnosti s obzirom na opseg proizvodnje. Kako se organizacija kreće niz krivulju učenja, smanjujući troškove, može se smanjiti i cijenu proizvoda. Tako je krivulje cijene i troškova moguće povezati s idejom životnog ciklusa proizvoda i različitim strategijama.

konkurencije te pogodnostima (O) u kojima se te prednosti mogu iskoristiti. Navedeno predstavlja logiku određivanja agresivnih strategija koje su tipa (DS<sub>i</sub>®DO<sub>k</sub>).

Nakon definiranja korektivnih i agresivnih strategija može se dogoditi da neke slabosti (W) i/ili prijetnje (T) ostanu neotklonjene. Kao takve, one su prepreka ostvarenju ZSC-a, zato je potrebno pronaći nove resurse kojima će se otkloniti preostale slabosti i prijetnje. Na takav način definiraju se obrambene strategije. Formalizacijom novih resursa u obliku elemenata SWOT-a, obrambena strategija može se svesti na korektivnu. Tako, primjerice, ako za otklanjanje neke slabosti (W) ili prijetnje (T) nedostaje radne snage, potrebno je ispitati postojanje potrebne radne snage na tržištu rada. Ponavljanjem prvog koraka (određivanje elemenata SWOT-a), postojanje radne snage na tržištu formalizira se u obliku pogodnosti (O). Na taj se način obrambena strategija predviđena metodom pSWOTa može prevesti u korektivnu. Ako se dogodi da ostane značajan broj prijetnji i slabosti koje nije moguće ni s čim otkloniti, potrebno je preispitati izvedivost ZSC-a. Konačna odluka o preispitivanju ZSC-a radi se nakon optimizacije strategije. U skladu s prethodnim, sve vrste i tipove mogućih strategija moguće je prikazati na slici 3.1.



*Slika 3.1 Vrste i tipovi korištenih strategija*

Utvrđivanje strategije provodi se pomoću matrice SWOT i matrice strategija koje su sastavni dijelovi tablice SWOT. Matrica SWOT konstruira se tako da se omogući sistematična provedba nalaženja svih mogućih elemenata strategija za ostvarenje ZSC-a te uočavanje rezidualnih slabosti i prijetnji. Niz elemenata SWOT-a iz tablice 3.2 postaje naslovni redak matrice SWOT. U naslovni stupac prvog sljedećeg retka prepisuje se prvi odredišni element ( $W_j$ ,  $O_k$  ili  $T_l$ ) iz niza elemenata SWOT-a u naslovnom retku matrice SWOT. Na taj način otvara se novi redak tablice. Otvaranjem novog retka otvoren je i postupak utvrđivanja strategije. Svaki redak predstavljat će jednu moguću strategiju.

Vrsta strategije određena je odredišnim elementom koji je zapisan u naslovnom stupcu promatranog retka matrice SWOT. Polja u retku mogu poprimiti vrijednosti 1 ili 0. Redom se promatra određuju li jedan ili kombinacija više elemenata SWOT-a iz

naslovnog retka, upareni s odredišnim elementima iz stupaca, strategiju za ostvarenje ZSC-a. Ako element stupca SWOT ili kombinacija više njih, uparena s odredišnim elementom (SWOT elementom retka), određuje strategiju, onda se u polja određena tim retkom i stupcima upisuje vrijednost 1. U ostala polja promatranog retka upisuje se vrijednost 0. Tip i opis strategije upisuje se u nastavku istog retka u *matrici strategija*. Tako je jednim retkom predočena jedna strategija. Ako se za jedan odredišni element uoče dvije ili više strategija, otvara se novi redak s istim odredišnim elementom za upis sljedeće uočene strategije. Postupak se ponavlja dok se ne iscrpe sve strategije. Forma tablica SWOT prikazana je tablicom 3.3.

Tablica 3.3 Tablica SWOT

Matrica SWOT												Matrica strategija			
Elementi SWOT-a	Opis S <sub>1</sub>	⋮	Opis S <sub>s</sub>	Opis W <sub>1</sub>	⋮	Opis W <sub>w</sub>	Opis O <sub>1</sub>	⋮	Opis O <sub>o</sub>	Opis T <sub>1</sub>	⋮	Opis T <sub>t</sub>			
	S <sub>1</sub>	⋮	S <sub>s</sub>	W <sub>1</sub>	⋮	W <sub>w</sub>	O <sub>1</sub>	⋮	O <sub>o</sub>	T <sub>1</sub>	⋮	T <sub>t</sub>	Tip strategije	Opis strategije	
Opis W <sub>1</sub>	W <sub>1</sub>														
⋮	⋮														
Opis W <sub>w</sub>	W <sub>w</sub>	0	⋮	1	0	⋮	0	0	⋮	1	0	⋮	0	Δ S <sub>s</sub> ∪ Δ O <sub>o</sub> * D W	Koristiti S <sub>s</sub> i O <sub>o</sub> za otkl. W <sub>w</sub>
Opis O <sub>1</sub>	O <sub>1</sub>														
⋮	⋮														
Opis O <sub>o</sub>	O <sub>o</sub>														
Opis T <sub>1</sub>	T <sub>1</sub>														
⋮	⋮														
Opis T <sub>t</sub>	T <sub>t</sub>														

**U trećem koraku** za svaku definiranu strategiju određuje se konkretna aktivnost za njezinu provedbu. Pritom vrijedi da se izvršenjem jedne aktivnosti može ostvariti više strategija, tj. dopušten je odnos  $m:1$ , gdje je  $m$  broj strategija. Za svaku tako utvrđenu aktivnost definira se izvedeni strateški cilj kojim će se moći utvrditi izvršenost



aktivnosti. Mjerenje razine izvršenja aktivnosti mjeri se jednim ISC-om. Zbog toga se može konstatirati kako je veza između aktivnosti i ISC-a 1:1. Definiranje aktivnosti i njezinih ISC-ova radi se tako da se tablica SWOT (tablica 3.3) proširi *matricom aktivnosti i izvedenih strateških ciljeva*. Prve su dvije kolone matrice oznaka i opis aktivnosti. Oznaka aktivnosti sastoji se od slova A (aktivnost) i indeksa. Opisom aktivnosti potrebno je, u skladu s izrazom (1.1), odgovoriti na pitanja „Što?“, „Tko?“ i „Čime?“ treba obaviti posao kako bi se ostvarila utvrđena strategija.<sup>1</sup>

Kao što je već rečeno, svaka se aktivnost provodi s nekim ciljem. Iz tog razloga se u nastavku prikazuju ciljevi utvrđenih aktivnosti. Semantičke komponente ISC-a iste su kao i kod ZSC-a. Potrebno je naglasiti da se kod označavanja ISC-a prvo slovo perspektive piše malim slovom, a indeks mora biti jednak indeksu aktivnosti zbog kojeg se ona provodi. Ovakvo proširenje tablice SWOT prikazano je tablicom 3.4.

Tablica 3.4 Proširenje tablice SWOT – matrica aktivnosti i ISC-a

Matrica aktivnosti i izvedenih strateških ciljeva									
Id. A.	Opis aktivnosti	Oznaka cilja	Smjer promjene	Objekt promjene	Naziv mjere	Oznaka mjere	As is	To be	Mjerna jedinica
Indeks aktivnosti (Ai)	Opis specifične aktivnosti	Prvo slovo perspektive i indeks	Povećati ili smanjiti	Što se želi ostvariti aktivnošću	Opis naziva	Oznaka cilja Ci	Sadašnja vrijednost mjere	Željena vrijednost mjere	Mjerna jedinica mjere

<sup>1</sup> Formulom 1.1 strateški plan je formaliziran u obliku funkcije strateški plan = f (što, tko, čime, kada). Odgovor na pitanje "Kada?" analizira se u četvrtom poglavlju prilikom analiziranja dinamike strateškog upravljanja.

Ovakvim proširenjem u tablici SWOT jednim su retkom prikazani svi koraci izvoda jednog ISC-a. ZSC-ovi su izvedeni direktno iz vizije, zbog čega ne sadrže elemente prethodnih koraka. Međutim, ZSC i ISC imaju iste semantičke elemente svog opisa. Iz tog razloga je moguće, bez narušavanja konzistentnosti cjelokupnog prikaza, u retke ispod ISC-a prepisati ZSC. Tako je dobiven cjelokupni prikaz ZSC-a, elemenata SWOT-a korištenih za definiranje strategija, aktivnosti za njihovu provedbu i ISC-a za mjerenje izvršenja aktivnosti. Potrebno je još uspostaviti odnose i utjecaje među ciljevima.

**Posljednji, četvrti korak je strukturiranje strategije.** Cilj je ovog koraka uspostaviti direktne odnose među strateškim ciljevima. Mogućnost velikog broja ISC-ova<sup>1</sup> zahtijeva sistematiziran pristup određivanju veza među strateškim ciljevima. Iz tog razloga je napravljeno još jedno, posljednje proširenje tablice SWOT matricom strukture ciljeva.

Matrica strukture ciljeva (MSC) tablični je prikaz odnosa i utjecaja među strateškim ciljevima. U svrhu postizanja preglednosti, u prvi stupac MSC-a (naslovni stupac MSC) prepisuju se oznake ciljeva. Potom se istim redosljedom kao u naslovnom stupcu oznake prepisuju u naslovni redak MSC-a<sup>2</sup>. Na taj način dobivena je kvadratna matrica u kojoj svaki redak, sukladno s oznakom, predočuje cilj kojim se vrši direktni utjecaj. Analogno, svaki stupac sukladno s oznakom predočuje cilj na koji se ostvaruje direktni utjecaj. U početnom koraku polazi se od pretpostavke da je direktni odnos uzročnih ciljeva ravnomjerno raspoređen. Zbog toga, ovisno o postojanju direktnog odnosa među ciljevima, elementi MSC-a poprimaju vrijednosti 1 ili 0. Ako postoji odnos među ciljevima, upisuje se vrijednost 1, dok se u suprotnom upisuje vrijednost 0. Prilikom korištenja modela za strateško upravljanje elementi MSC-a mogu poprimiti i druge vrijednosti. Matrica strukture ciljeva prikazana je tablicom 3.5.

*Tablica 3.5 Proširenje tablice SWOT – matrica strukture ciljeva (MSC)*

Matrica strukture ciljeva (MSC)										
Oznake ciljeva	ISC <sub>1</sub>	...	ISC <sub>i</sub>	ISC <sub>i+1</sub>	ISC <sub>i+2</sub>	...	ISC <sub>k</sub>	ZSC <sub>1</sub>	...	ZSC <sub>j</sub>
ISC <sub>1</sub>										
⋮										

<sup>1</sup> Prema pSWOTa preporučuje se do 30-tak ciljeva, relativno ravnomjerno raspoređenih po BSC perspektivama

<sup>2</sup> Naslovni redak MSC smješten je u redak u kojem su popisani nazivi semantičkih

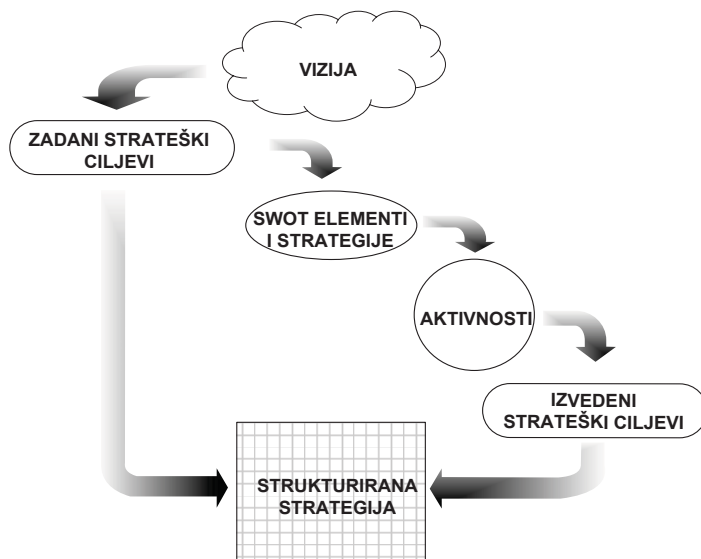
ISC <sub>i</sub>	0	...	0	1	0	...	0	0	...	1
ISC <sub>i+1</sub>										
ISC <sub>i+2</sub>										
⋮										
ISC <sub>k</sub>										
ZSC <sub>1</sub>										
⋮										
ZSC <sub>l</sub>										

Postupak utvrđivanja utjecaja među ciljevima polazi od prvog retka. Prolazeći duž polja u retku, promatra se ima li *cilj predočen retkom* direktni utjecaj na *cilj predočen stupcem*. Ako direktni utjecaj postoji, u polje se upisuje vrijednost 1, dok se u suprotnom upisuje vrijednost 0. Nakon što je ispitan direktni utjecaj cilja iz promatranog retka na sve ciljeve prezentirane stupcima, prelazi se na sljedeći redak, npr. iz tablice 3.5 može se pročitati da ISC<sub>i</sub> ima direktni utjecaj na ostvarenje izvedenog strateškog cilja ISC<sub>i+1</sub> i zadanog strateškog cilja ZSC<sub>l</sub>.

Tablicom SWOT (tablica 3.3) koja sadrži matricu SWOT i matricu strategija, nadopunjenom matricom aktivnosti i ISC-a (tablica 3.4) te zadanim strateškim ciljevima i matricom strukture ciljeva (tablica 3.5) objedinjen je prikaz rezultata strateške analize, a nazvana je *proširena SWOT tablica* (pSWOTt). Sve utvrđene komponente u pSWOTt podložne su promjenama. Ostvareni učinci na kraju svakog razdoblja ukazat će na potrebne promjene u elementima pSWOTt.

Na ovaj način slabo formalizirana vizija u obliku izjava (prikazana oblačićem) transformirana je u opisno i kvantitativno utvrđene ZSC-ove (prikazani zaobljenim pravokutnikom). Postupak određivanja načina njihova ostvarenja zahtijevao je donošenje sudova i strategija koji su zbog svog isključivo opisnog karaktera (prikazano elipsom) poprimili slabiju formu nego ZSC u prethodnom koraku. Skup aktivnosti jasno određuje smjernice buduće forme i može se smatrati proširenjem opisnog dijela ISC-a, zbog čega je prikazan krugom. Ovo proizlazi iz činjenice da se svaka aktivnost poduzima s točno određenim ciljem (1:1). Za razliku od aktivnosti, više strategija može se ostvariti jednom aktivnosti (m:1). Kao i zadani strateški ciljevi, izvedeni strateški ciljevi (prikazani zaobljenim pravokutnikom) sadrže opisne i numeričke semantičke elemente. Visoko formaliziranim MSC-om i numeričkim elementima semantike

strateških ciljeva ostvarena je veza između realnog sustava i njegova matričnog modela (slika 3.2).



*Slika 3.2 Postupak formalizacije strategije (Hell, Vidačić & Garača, 2009)*

Na taj način ostvareni su formalni uvjeti za primjenu matematičkih metoda kod računanja učinkovitosti organizacije.

## 3.2 Mjerenje učinaka strategije IS-a/IT-ja na zadane strateške ciljeve

### 3.2.1 Matrični pristup mjerenju učinaka

Matematika je apstraktna znanost koja je razvijena prema dokazanim zakonitostima. Uspije li se određeni poslovni problem preslikati na matematički problem, potrebno je „samo“ potom primijeniti matematičke zakonitosti. Normalno, to ne znači da će čitatelji daljnjeg teksta postati matematičari nego da ćete moći s njima komunicirati kako biste ukazali na poslovni problem koji želite riješiti.

Matrice nisu ništa drugo nego radna lista u *Excelu*. Redci/stupci radne liste su redci/stupci u matrici. To je dvodimenzionalni prikaz problema. Tumačenje matrice u našem je slučaju opis strukture utjecaja strateških ciljeva. Što to znači? Čitamo li elemente u prvom stupcu matrice, možemo vidjeti (tamo gdje je vrijednost 1) one strateške ciljeve u retku čije ostvarenje utječe na ostvarenje strateškog cilja u prvom stupcu. Kako bismo to dokazali, uvodimo pojam relacije. Pojednostavljeno rečeno, relacija je svako ono preslikavanje koje nije funkcija. Trebali bismo znati da se kod funkcije jedan element domene može preslikati samo na jedan element kodomene. U našem problemu, jedan strateški cilj može biti preduvjet ostvarenja (podređen) nekoliko ciljeva, tj. njih više. Drugim riječima, može se pojavljivati u više stupaca. To

znači da ne možemo govoriti o funkciji. Ako nije funkcija, onda je relacija. Što to znači? Primjer relacije potpunog uređaja na skupu svih prirodnih brojeva je *biti manji*, tj.  $1 < 2 < 3 < 4 \dots$ . U našem slučaju, nemamo ni to nego ono što znamo je da je npr. SC1 podređen SC2 podređen SC3 i znamo da, ako je SC1 podređen SC2, ne može vrijediti da je SC2 podređen SC1 (antisimetričnost). Također, iz prirode problema znamo da, ako je SC1 podređen SC2, a SC2 podređen SC3, i SC1 podređen SC3 (tranzitivnost). To se zove relacija parcijalnog uređaja. Ne ulazeći dalje u matematiku, svaka relacija parcijalnog uređaja može se dopuniti do relacije potpunog uređaja. Konkretno, u čemu je problem sa strateškim ciljevima? Ako pogledamo prirodne brojeve, oni predstavljaju niz i znamo koji ide prije kojeg (manji ide naprijed). Međutim, struktura strateške mape ciljeva je mrežna struktura. To znači da prilikom spuštanja niz mrežu ciljeva možemo otići u više smjerova koji nas mogu dovesti do različitih podređenih ciljeva koji nisu ni u kakvoj relaciji podređenosti. E, to je relacija parcijalnog uređaja. Iz prirode problema znamo da je ta relacija antisimetrična i tranzitivna, pa ju je moguće dopuniti do relacije potpunog uređaja, tj. pojednostavljeno rečeno dogovoriti se koji ćemo staviti prije kojeg. To će se riješiti tako što će se pojam *podređen* zamijeniti pojmom relacija *biti podređen*, što je pokazano u nastavku.

Neka je u skladu s postupkom prikazanim u odjeljku 3.1 utvrđeno  $l$  zadanih strateških ciljeva te na temelju njih u skladu s postupkom prikazanim u odjeljku 3.2 izvedeno  $k$  izvedenih strateških ciljeva. Ovako utvrđeni ZSC i ISC su elementi konačnog skupa svih strateških ciljeva organizacije koje je moguće zapisati izrazom (3.2).

$$\tilde{C} = \{C_1, C_2, \dots, C_n\}, \quad n = k + l \quad (3.2)$$

Izostave li se prazni i naslovni redci i stupci u MSC, direktni utjecaj među ciljevima općenito je moguće prikazati u strogoj formi kvadratne matrice.

$$MSC = \begin{bmatrix} 0 & c_{12} & c_{13} & \dots & c_{1n} \\ c_{21} & 0 & c_{23} & \dots & c_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & c_{n3} & \dots & 0 \end{bmatrix}. \quad (3.3)$$

Prelaskom na prikaz strukture ciljeva u obliku matrice, oznake ciljeva iz tablice 3.5 zamijenjene su indeksom reda, odnosno stupcem matrice. Red kvadratne matrice MSC označuje ukupni broj strateških ciljeva. U skladu s postupkom formiranja MSC,  $c_{ij} \in \{0,1\}$ . Formiranje zapisa u skladu s poretком ciljeva od nadređenog prema

podređenom postiže se svođenjem MSC-a na donju trokutastu matricu<sup>1</sup>. U nastavku teksta dokazana je mogućnost svođenja MSC-a na donju trokutastu matricu.

Na skupu strateških ciljeva  $\tilde{C} = \{C_1, C_2, \dots, C_n\}$ , u skladu s utvrđenim direktnim utjecajima prikazanim u MSC-u, neka je definirana relacija<sup>2</sup> *biti podređen* (BP). Nadalje, neka su iz  $\tilde{C}$  proizvoljno odabrani ciljevi  $X, Y, Z \in \tilde{C}$ . Na temelju koncepta BSC slijedi da, ako je element  $X$  podređen cilju  $Y$ ,  $Y$  nije podređen elementu  $X$ , tj.

$$X \text{ BP } Y \not\Rightarrow Y \text{ BP } X \quad (3.4)$$

Može se zaključiti kako je relacija BP antisimetrična. Za ovako definiranu relaciju baze podataka (BP) moguće je konstatirati i svojstvo tranzitivnosti, tj.

$$Z \text{ BP } X \wedge X \text{ BP } Y \Rightarrow Z \text{ BP } Y \quad (3.5)$$

Antisimetrična (3.4) i tranzitivna (3.5) relacija zove se relacija parcijalnog uređaja, stoga je definirana relacija BP-a relacija parcijalnog uređaja.

Drugi važan tip relacije na skupu  $\tilde{C}$  je relacija potpunog uređaja. Relacija potpunog uređaja "<" je relacija parcijalnog uređaja u kojoj su svaka dva elementa  $X$  i  $Y$  usporediva, tj.

$$X < Y \quad \text{ili} \quad Y < X \quad (3.6)$$

Važan matematički teorem kaže da se svaka relacija parcijalnog uređaja može nadopuniti do relacije potpunog uređaja (Horvatić, 2004). Dakle, relacija BP može se nadopuniti do relacije potpunog uređaja "<". Ne smanjujući općenitost, smije se pretpostaviti da je

$$C_1 < C_2 < \dots < C_n. \quad (3.7)$$

Dokaz da je u ovom poretku ciljeva  $C_i$  matrica MSC donja trokutasta provodi se tako da se pretpostavi suprotno, tj. da je  $c_{ij} > 0$ , gdje je  $j > i$ . Iz toga bi slijedilo da je dio  $C_i$  podređen dijelu  $C_j$ , tj.

$$C_i \text{ BP } C_j, \text{ no onda je } i < C_j, \text{ što je u proturječju s } C_j < C_i \text{ (što slijedi iz (3.7)).}$$

Ovime je dokazana mogućnost svođenja MSC-a na donju trokutastu matricu. Svođenjem na donju trokutastu matricu ostvarena je preglednost direktnih utjecaja.

---

<sup>1</sup> Kvadratna matrica je donja trokutasta ako su joj svi elementi iznad glavne dijagonale jednaki nula.

<sup>2</sup> Relacija na skupu  $\tilde{C}$  je podskup Kartezijeva produkta  $\tilde{C} \times \tilde{C}$ .

Dobiveni redosljed zapisa ciljeva opisuje mrežnu strukturu utjecaja među ciljevima. To znači da su nadređeni ciljevi, sukladno lancima uzroka i posljedica i gledajući prema indeksu stupaca ili redaka u MSC-u, zapisani prije njima podređenim ciljevima.

Promatrajući stupce matrice MSC, jednostavno je identificirati direktne veze između nadređenog cilja prezentiranog stupcem i njemu podređenog cilja prezentiranog retkom. Vrijednost elementa koji se nalazi na sjecištu uočenog retka i stupca identificira (ne)postojanje<sup>1</sup> direktnog utjecaja između ciljeva koje taj redak i stupac prezentiraju. Na ishodišni cilj ne utječe ostvarenje ni jednog drugog cilja. Zbog toga su elementi stupca ishodišnih ciljeva jednaki nuli. Analogno, svi elementi retka odredišnog cilja jednaki su nuli jer se njima ne ostvaruje utjecaj ni na jedan drugi cilj. Na taj način, čitanjem MSC-a lako je uočiti koji niže pozicionirani ciljevi posljedično uzrokuju promjenu razine ostvarenja njemu nadređenih ciljeva i koji su mu to nadređeni ciljevi. Izostave li se iz MSC-a ishodišni i odredišni ciljevi, dobije se podmatrica na koju se, primjenjujući isti postupak, određuju ishodišni i odredišni ciljevi.

Ovakav pristup može se koristiti prilikom crtanja strateških mapa što je, prema BSC-u, osnovni grafički prikaz veza među aktivnostima koje se poduzimaju radi ostvarenja svih strateških ciljeva. Iz svega prethodnog može se zaključiti da je ovakav oblik zapisa MSC-a zapravo matrični zapis strateških mapa ciljeva. Kao što je već rečeno, razina ostvarenosti cilja mjeri se mjerom. Sve mjere nalaze se na strateškoj mapi mjera. Strateška mapa mjera strukturni je ekvivalent strateškoj mapi ciljeva. Prema BSC-u, logička osnova za njezinu konstrukciju temelji se na postavci da se razina ostvarenja svakog cilja mjeri u određenim vremenskim intervalima, a izmjerene veličine uspoređuju kako bi se ustanovilo daju li poduzete aktivnosti očekivani učinak. Prilikom definiranja ciljeva i njihovih mjera, utvrđivanjem *as is* i *to be* vrijednosti, određen je i raspon promjene mjere. Prema formuli (3.1), izmjerenu apsolutnu vrijednost mjere na kraju promatranog perioda moguće je pretvoriti u relativnu vrijednost promjene mjere.

U svrhu računanja utjecaja ostvarenja pojedinih ciljeva na ostvarenje drugih ciljeva uvode se koeficijenti utjecaja. Koeficijenti utjecaja predočuju težinu kojom vrijednosti jedne mjere utječu na vrijednost druge mjere. Opseg promjene mjere posljedičnog cilja jednak je sumi umnožaka opsega promjene mjere uzročnih ciljeva i njihovih koeficijenata utjecaja. Što to znači?

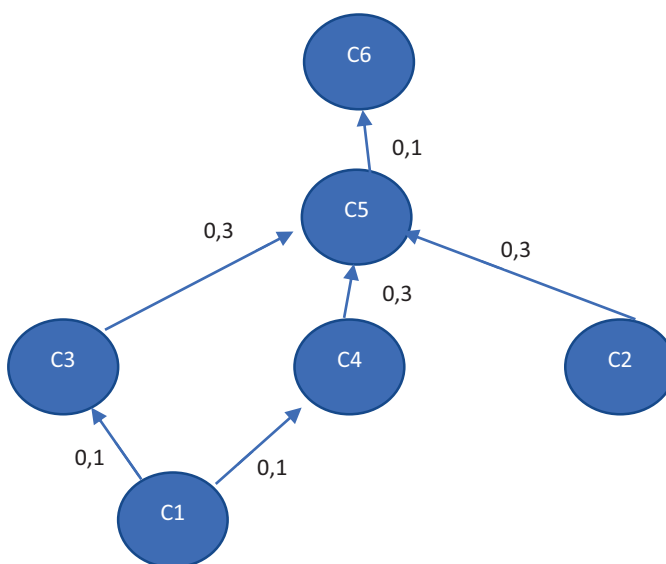
Uzmimo za primjer da je donesena odluka o gradnji kuće. Da bi se sagradila kuća, potrebne su određene aktivnosti koje ću ovdje promatrati kroz sljedeće ciljeve (svaka

---

<sup>1</sup> Prema načinu popunjavanja matrice strukture ciljeva u pSWOTt, vrijednost 1 označuje postojanje direktnog utjecaja, dok vrijednost 0 označuje nepostojanje direktnog utjecaja.

aktivnost rezultira nekim ciljem): C1 –izraditi projektnu dokumentaciju, C2 – osigurati financijska sredstva, C3 – ishoditi građevinsku dozvolu, C4 – ugovoriti građevinske radove, C5 – provesti građevinske radove, C6 – upisati nekretninu u knjige (iako je spremna za useljenje, zakonski je ovo formalno jedino ispravno). Tek nakon svih napravljenih koraka može se reći da je kuća gotova. Pojednostavimo priču – ostvarenje cilja C3 ovisi o postotku realizacije C1. Osiguranje novaca (C2) preduvjet je da bi se započelo s gradnjom (C5). Cijeli proces izrade dokumentacije sastoji se od nekoliko poslova koje je moguće jasno razgraničiti i reći kolika je izvršenost. S druge strane, ostvarenje C1 jedini je preduvjet za pokretanje potrebnih aktivnosti za ostvarenje cilja C3. To znači da je koeficijent utjecaja C1 na C3 jednak 1 (jer je jedini). To znači da završenost nekog cilja ovisi o dvama uvjetima – o ostvarenju uzročnih ciljeva (preduvjeta) i o realizaciji aktivnosti za ostvarenje promatranog cilja.

Promatrajmo sada samo uvjetovanost realizacije na temelju preduvjeta. Jednostavno pitanje glasi: „Koliko ostvarenje 10 % cilja C1 doprinosi ostvarenju zadanog strateškog cilja C6?“ Kako bismo lakše razumjeli sljedeći izračun, struktura plana prikazana je primjerom na slici 3.3.



Slika 3.3 Strateška mapa ciljeva

Redom možemo izračunati učinke na svaki sljedeći strateški cilj:

- 10%\*1 (ostvarenost C1 pomnožena s koeficijentom utjecaja C1 na C3 koji iznosi 1)=0,1;
- 10%\*1 (ostvarenost C1 pomnožena s koeficijentom utjecaja C1 na C4 koji iznosi 1)=0,1;



$$c) 0,1((iz a)*0,33 (C3 na C5) + 0,1 (iz b))*0,33 (C4 na C5) + (X\% posto C2)*0,33 (C2 na C5) = 0,0666+X\%*0,333 (izvršenost cilja C5);$$

$$d) (0,0666+X\%*0,333)*1 (C5 na C6) = 0,0666+X\%*0,333.$$

To znači da 10 % ostvarenosti C1 i X % ostvarenosti C2 znači i 0,0666+X%\*0,333 ostvarenosti zadanog strateškog cilja C6.

Poopćavanje ovog primjera dovodi do sustava jednadžbi koje je matematički moguće prikazati formulom (3.12). Daljnji dokaz algoritma ide u smjeru primjene beskonačne sume geometrijskog reda.

Na ovaj način možemo izračunati utjecaj parcijalne realizacije cilja u kontekstu zadanih strateških ciljeva. U kontekstu mjerenja učinaka IS-a/IT-ja na poslovni sustav, moguće je procijeniti koliko ostvarenje izvedenog strateškog cilja strategije razvoja informacijskog sustava ima učinak na zadani strateški cilj koji je dio poslovne strategije.

Poopćavanjem prethodnog, polazni koeficijenti utjecaja mogu se izvesti iz MSC-a tako da se MSC normira po stupcima na temelju izraza (3.8):

$$k_{ij} = \begin{cases} \frac{c_{ij}}{\sum_{i=1}^n c_{ij}} & \text{za } \sum_{i=1}^n c_{ij} \neq 0 \\ 0 & \text{za } \sum_{i=1}^n c_{ij} = 0 \end{cases}.$$

(3.8).

Matrica dobivena normiranjem stupaca izračunatih ciljeva iz MSC-a je matrica strukture mjera (MSM) i prikazana je izrazom (3.9)

$$MSM = \begin{bmatrix} 0 & k_{12} & k_{13} & \cdots & k_{1n} \\ k_{21} & 0 & k_{23} & \cdots & k_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ k_{n1} & k_{n2} & k_{n3} & \cdots & 0 \end{bmatrix},$$

(3.9)

gdje se elementi  $k_{ij}$  računaju prema izrazu (3.8). Elementi MSM-a predočuju težinu kojom vrijednost mjere u retku utječe na vrijednost mjere u stupcu. Pritom vrijedi da je poredak mjera u MSC-u isti kao i poredak njihovih ciljeva u MSM-u. Suma elemenata po stupcima koji predočuje utjecaje na izračunate mjere jednaka je jedan, a suma elemenata po stupcima koji predočuje utjecaje na upisne mjere jednaka je nuli jer se na njih ne ostvaruje nikakav utjecaj.

Koncept računa prvo je prikazan na općem primjeru triju ciljeva ( $X$ ,  $Y$  i  $Z$ ) s izračunatim mjerama i četiriju ciljeva ( $C1$ ,  $C2$ ,  $C3$  i  $C4$ ) s upisnim mjerama. Nakon primjera prikazan je matricni model za proizvoljan broj ciljeva. Navedenih sedam ciljeva tvore skup svih strateških ciljeva promatranog primjera. Njihova struktura opisana je izrazom (3.10).

$$MSC = \begin{matrix} & \begin{matrix} X & Y & Z & C1 & C2 & C3 & C4 \end{matrix} \\ \begin{matrix} X \\ Y \\ Z \\ C1 \\ C2 \\ C3 \\ C4 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

(3.10)

Uz matricu su, radi lakšeg uočavanja utjecaja, dopisani naslovni redci i stupci u kojima se nalaze prethodno definirane oznake ciljeva. Prema (3.8) i (3.9) računa se matrica strukture mjera. Nakon izračunavanja MSM je definiran izrazom (3.11).

$$MSM = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0,25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0,25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,5 & 1 & 0 & 0 & 0 & 0 \\ 0,25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0,25 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

(3.11)

Kao što je već rečeno, elementi MSM-a predočuju intenzitet kojim vrijednost jedne mjere utječe na vrijednost druge mjere. Izračunavanje vrijednosti ukupnih relativnih promjena mjere moguće je svesti na rješavanje sustava jednadžbi. Sustav jednadžbi za promatrani primjer prikazan je izrazom (3.12).

$$\left\{ \begin{array}{l} 0,25 \cdot m_R Y + 0,25 \cdot m_R Z + \phantom{0,25 \cdot m_R C1} + 0,25 \cdot m_R C3 + 0,25 \cdot m_R C4 = m_R X \\ \phantom{0,25 \cdot m_R Y} + 0,5 \cdot m_R C1 + 0,5 \cdot m_R C2 = m_R Y \\ \phantom{0,25 \cdot m_R Y} + \phantom{0,5 \cdot m_R C1} + 1 \cdot m_R C2 = m_R Z \end{array} \right.$$

(3.12)

Dobiveni sustav sadrži tri jednadžbe (tri izračunate mjere) sa sedam nepoznanica (sedam mjera). Ovaj sustav ima četveroparametarsko rješenje. Sustav se rješava tako da se odaberu četiri nepoznanice iz sustava (broj nepoznanica – broj jednadžbi = broj parametara) koji će se parametrizirati. Sukladno primjeru, upravo su to četiri upisne mjere koje se ne izračunavaju nego upisuju. Tako se problem izračunavanja ukupnih relativnih vrijednosti promjena izračunatih mjera svodi na rješavanje sustava tri jednadžbe s tri nepoznanice.

Neka su vrijednosti relativnih promjena upisnih mjera:  $C_1=0,33$ ;  $C_2=1$ ;  $C_3=0,83$ ;  $C_4=1$ . Tada je sustav jednadžbi, u skladu s prethodnim, moguće zapisati izrazom (3.13).

$$\begin{cases} 0,25 \cdot m_R Y + 0,25 \cdot m_R Z + 0,46 = m_R X \\ 0,67 = m_R Y \\ 1 = m_R Z \end{cases} \quad (3.13)$$

Stupac sa slobodnim koeficijentima (treći stupac) rezultat je direktnog uvrštavanja vrijednosti upisnih mjera. U kontekstu BSC-a, stupac sa slobodnim koeficijentima moguće je tumačiti kao parcijalnu promjenu vrijednosti izračunatih mjera koja se temelji na direktnom utjecaju (direktno uvrštavanje) vrijednosti upisnih mjera. Računanje indirektnog utjecaja dobiva se uvrštavanjem vrijednosti  $m_R Y=0,67$  i  $m_R Z=1$  u prvu jednadžbu te množenjem s pripadajućim koeficijentima utjecaja. U kontekstu BSC-a moguće je tumačiti da promjena vrijednosti  $m_R C_1$  i  $m_R C_2$ , preko  $m_R Y$  i  $m_R Z$ , indirektno utječe na promjenu vrijednosti  $m_R X$ . Zbrajanjem sa slobodnim koeficijentom iz trećeg stupca (zbroj vrijednosti temeljen na indirektnom i direktnom utjecaju) izračunava se ukupna promjena  $m_R X \approx 0,876$ , tj. u kontekstu BSC-a vrijednost ukupne relativne promjene  $m_R X$ .

Kao što se može vidjeti iz prethodnog primjera, koraci izračunavanja vrijednosti ukupne relativne promjene izračunatih mjera ekvivalentni su koracima rješavanja sustava jednadžbi. Poopćavanje prethodnog primjera omogućit će istovremeno računanje ukupne razine ostvarenosti za neograničeni broj ciljeva.

Neka je s  $(m_R C_1, \dots, m_R C_n)$  označena uređena  $n$ -torka vrijednosti relativne promjene svih mjera na kraju promatranog perioda. Redoslijed zapisa definiran je poretkom mjera u MSM-u. Uređenu  $n$ -torku moguće je zapisati i u obliku *vektora ukupne razine ostvarenosti ciljeva* prikazanog izrazom (3.14).

$$M = [m_R C_1 \quad \dots \quad m_R C_n]. \quad (3.14)$$

Promjena vrijednosti mjera niže pozicioniranih ciljeva u lancu uzroka i posljedica (LUP) rezultira promjenom vrijednosti mjera svih njemu nadređenih ciljeva. To znači da se

utjecaj ostvaruje i indirektno, tj. putem posrednih ciljeva. Računom je potrebno odrediti komponente vektora  $M$ , tj. vrijednosti ukupnih relativnih promjena mjera.

Račun ukupnih vrijednosti mjera može se podijeliti u  $m-1$  koraka<sup>1</sup>.

$$MSM^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0,125 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0,375 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \quad (3.19)$$

Koeficijent  $k_{14}=0,125$  iz kvadriranog MSM-a u izrazu (3.19) označuje utjecaj upisne mjere  $m_{RC1}$  na  $m_{RX}$  (iako ne postoji direktni utjecaj). Na temelju indirektnog utjecaja  $m_{RC1}$  i  $m_{RC2}$  ostvarena promjena  $m_{RX}$  iznosi

$$0,125 \times m_{RC1} + 0,375 \times m_{RC2} = 0,125 \times 0,33 + 0,375 \times 1 \approx 0,416. \quad (3.20)$$

Prema (3.13) suma vrijednosti koja se temelji na direktnim utjecajima C3 i C4 (vrijednost u trećem stupcu) iznosila je 0,46. Ukupna vrijednost relativne promjene mjere  $m_{RX}$  jednaka je zbroju relativnih promjena dobivenih direktnim i indirektnim utjecajem.

$$0,46 + 0,416 = 0,876. \quad (3.21)$$

Izračunata vrijednost ukupne razine ostvarenosti cilja  $m_{RX}=0,876$  dobivena ovim načinom jednaka je rješenju sustava jednačbi iz (3.13).

Poopćenjem prethodnog primjera ukupnu relativnu promjenu mjera, umjesto putem sume parcijalnih vrijednosti mjera, moguće je računati direktno na temelju  $M_0$  i utvrđenih direktnih utjecaja.

Izlučivanjem vektora  $M_0$  u izrazu (3.18) dobije se izraz (3.22)<sup>2</sup>.

---

<sup>1</sup> Prema prethodnom primjeru,  $m$  je broj jednačbi, a  $m-1$  broj koraka (uvršćavanja) izračunatih vrijednosti.

<sup>2</sup>  $I$  je oznaka za jediničnu matricu. Jedinična matrica je kvadratna matrica kojoj su svi elementi na dijagonali jednaki jedan, a svi ostali elementi matrice jednaki nuli. Primjer jedinične matrice  $n$ -tog reda

$$M = M_0 \times (I + MSM + MSM^2 + \dots + MSM^{m-1} + \dots) \quad (3.22)$$

Prema sumi beskonačnog reda<sup>1</sup>:

$$M = M_0 \times (I - MSM)^{-1} \quad (3.23)$$

Izraz s kojim se množi vektor  $M_0$  u formuli (3.23):

$$MSU = (I - MSM)^{-1}, \quad (3.24)$$

nazvan je *matrica sume utjecaja* (MSU) i koristi se za računanje vektora ukupne razine ostvarenosti ciljeva na temelju ostvarenih vrijednosti upisnih mjera.

$$M = M_0 \times MSU. \quad (3.25)$$

Ovime je u potpunosti definiran postupak izračunavanja ukupne razine ostvarenosti ciljeva kroz koje se promatra učinkovitost organizacije.

### 3.3 Optimizacija učinaka IS-a/IT-ja primjenom metode linearnog programiranja

Kako bi se mogla provesti optimizacija, potrebno je imati ograničenja i funkciju cilja koju želimo maksimirati ili minimizirati. U ovom slučaju, govorimo o maksimiranju u kontekstu poslovnih rezultata. Ograničenja su najčešće resursi koji nam stoje na raspolaganju. Kad imamo milijun kuna, a za provedbu plana potrebno je sto tisuća, optimizacija nije potrebna jer imamo više nego nam je potrebno pa će se rezultat ostvariti 100 %. Međutim, ovakve su situacije rijetke.

Drugi tip ograničenja proizlazi iz same strukture strategije. Primjera radi, vratimo se na proces studiranja u svrhu ostvarivanja boljeg radnog mjesta. Biste li počeli raditi poslove konobara za 10000 kn mjesečno?<sup>2</sup> Znači, bez potrebe završavanja fakulteta i

$$\text{je } I = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

<sup>1</sup> Red  $I + MSM + MSM^2 + MSM^3 + \dots + MSM^m + \dots$  konvergira i ima sumu  $(I - MSM)^{-1}$ ,

ako konvergira svih  $n^2$  redova  $\sum_{r=1}^{\infty} k_{ij}^r$ , gdje  $k_{ij}^r$  označuje element u  $i$ -tom retku i  $j$ -tom stupcu  $j$  matrice

$MSM^r = MSM \cdot MSM \cdot \dots \cdot MSM$  ( $r$  puta). Kao što je već rečeno, elementi  $m$ -te potencije matrice  $MSM$  označuju intenzitet  $m$ -tog indirektnog utjecaja kojim vrijednosti upisnih mjera utječu na vrijednost izračunatih mjera. Zbog toga svaki od  $n^2$  elemenata matrice teži prema nuli kad  $m \rightarrow \infty$ , tj. matrica  $MSM^m$  konvergira prema nul-matrici (matrica čiji su svi elementi jednaki nuli).

<sup>2</sup> Ovime se ne želi podcijeniti nijedno zanimanje, nego ukazati na formalni način donošenja odluka.

svih izazova koje studiranje nosi imati primanja koja ćete možda realizirati tek za nekoliko godina. Normalno, nije sve u kratkoročnim primanjima od 10000 kn. To je, pojednostavljeno gledajući, samo jedan dio vizije i za njega je potrebno, primjerice, samo napraviti prezentaciju potencijalnim poslodavcima (prva niža razina). Za ostvarenje ostalih dijelova vizije potrebno je puno više koraka. Drugim riječima, strateški plan ima puno više razina strateških ciljeva (gledajući lance uzroka i posljedica). Proces optimizacije u ovoj situaciji balansira između brzog ostvarenja vizije (ograničenja uzrokovana strukturom strategije) i dubinskog izbora ciljeva za dugoročnije osiguranje ostvarenja vizije (ograničenja uzrokovana raspoloživim resursima).

Vi ste svi izabrali studiranje, no nekima to životne prilike (resursi) nisu omogućili. Ovdje su prikazane dvije krajnosti od kojih obje zadovoljavaju dio vizije iz prethodnog primjera, a to je primanja od 10000 kn mjesečno. Rijetka je situacija da moramo birati binarno (da/ne). Postupak optimizacije trebao bi dati odgovor na pitanje što je optimalno. Na isti način, vlasnici poduzeća na kraju godine odlučuju hoće li podijeliti dividendu ili će taj prihod uložiti u daljnji razvoj IS-a/IT-ja. Opet binarno! Pojednostavljeno rečeno, u biti bi trebalo postaviti problem za optimizaciju i izračunati omjere isplate dividende i ulaganja u IS/IT razvoj.

Klasični koncept BSC u fazi planiranja učinaka podrazumijeva provođenje utvrđenih aktivnosti. Međutim, u realnom sustavu provođenje aktivnosti može biti uvjetovano različitim ograničenjima. Zbog toga je željenu razinu ostvarenosti ciljeva potrebno uskladiti s mogućim ograničenjima. Prikazani koncept strateškog upravljanja istaknuo je dva tipa ograničenja.

Klasični tip ograničenja za postizanje željene razine ostvarenosti ciljeva je raspoloživost resursa za provođenje aktivnosti kojima se ti ciljevi ostvaruju. Zbog toga je za svaku aktivnost potrebno ustanoviti potrebne resurse za njenu provedbu. Pritom se jedan resurs može koristiti u više aktivnosti. Kako bi se obuhvatili svi potrebni resursi, postupak je sistematiziran i formaliziran u obliku matrice resursa prikazane tablicom 3.6.

Tablica 3.6 Matrica resursa

Matrica resursa						
Mjerna jedinica	$m/R_1$	...	$m/R_i$	$m/R_{i+1}$	...	$m/R_r$
Opis resursa	Opis $R_1$	...	Opis $R_i$	Opis $R_{i+1}$	...	Opis $R_r$

Oznaka	$R_1$	...	$R_i$	$R_{i+1}$	...	$R_r$
Utrošak resursa u retku određenom aktivnošću						
	0	...	100	0	...	50
Ukupna raspoloživost $R_i$			70			100

Priroda aktivnosti ukazuje na potrebne resurse za njezinu provedbu. Utvrđivanjem pripadajućih izvedenih strateških ciljeva jasno su definirana mjerila i raspon promjena koje bi se trebale ostvariti provedbom planiranih aktivnosti. To znači da se svi potrebni resursi mogu generirati na temelju matrice aktivnosti i ISC-a (tablica 3.4). Matrica resursa dopisuje se u nastavku pSWOTt, s desne strane.

Naslovni redci matrice resursa moraju sadržavati oznaku i opis resursa te njegovu mjernu jedinicu. Prolazeći po redcima u matrici aktivnosti i ISC-u (tablica 3.4), potrebno je utvrditi resurs za provedbu promatrane aktivnosti. Njegova oznaka, opis i mjerna jedinica zapisuju se u naslovnim redcima matrice resursa (tablica 3.6). Vrijednosti polja u matrici resursa, određene retkom promatrane aktivnosti i stupcem resursa, označuju potreban utrošak resursa za 100 % ostvarenja ISC-a zbog kojeg se promatrana aktivnost provodi. Tako je prema primjeru prikazanom u tablici 3.6 za provođenje aktivnosti  $A_i$  (iz primjera tablice 3.4) potrebno 100  $m_j R_i$  resursa  $R_i$  i 50  $m_j R_r$  resursa  $R_r$ . Nakon što su utvrđeni svi potrebni resursi za provedbu promatrane aktivnosti, prelazi se na sljedeću aktivnost, tj. novi redak u matrici aktivnosti i ISC-u.

Prilikom definiranja resursa potrebno je pogledati je li resurs već definiran. Ako definicija resursa već postoji, potreban utrošak upisuje se u ćeliju postojeće kolone. Nakon što su analizirane sve aktivnosti, u zadnjem retku matrice resursa upisuju se raspoložive vrijednosti resursa. Ovako formiranu tablicu moguće je prevesti u matrični oblik kojim se opisuje ograničenje uzrokovano raspoloživošću resursa.

Neka je utvrđeno  $k$  ISC-a i aktivnosti za njihovo ostvarenje te  $l$  ZSC-a. Neka je, nadalje,  $M$  vektor ukupne razine ostvarenosti ciljeva koji je opisan izrazom (3.14). S  $MR$  neka je označena matrica resursa tipa  $n \times r$ . Elementi prvih  $k$  redaka opisuju potrebnu količinu  $j$ -tog resursa,  $j=1, \dots, r$ , u  $i$ -toj aktivnosti,  $i=1, \dots, k$ . Posljednjih  $l$  redaka matrice predstavljaju potrebne resurse za ostvarenje ZSC-a. Kako se ZSC ostvaruje isključivo kroz ostvarenje ISC-a, elementi  $l$  redaka jednaki su nuli.

Posljednji redak iz matrice resursa, koji označuje raspoloživu količinu resursa, neka je predstavljen vektorom  $R$ . Tada se ograničenje može prikazati matričnom jednadžbom

$$M \times MR \leq R \quad (3.26)$$

tj.

$$[m_R C_1 \quad \dots \quad m_R C_n] \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1r} \\ r_{21} & r_{12} & r_{23} & \dots & r_{2r} \\ \vdots & \vdots & \vdots & & \vdots \\ r_{k1} & r_{k2} & r_{k3} & \dots & r_{kr} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \leq [R_1 \quad \dots \quad R_r]. \quad (3.27)$$

Nakon množenja matrica dobije se sustav nejednadžbi koji je prikazan izrazom (3.28):

$$\begin{cases} r_{11} \cdot m_R C_1 + \dots + r_{k1} \cdot m_R C_k \leq R_1 \\ r_{12} \cdot m_R C_1 & r_{k2} \cdot m_R C_k \leq R_2 \\ \vdots & \vdots & \vdots \\ r_{1r} \cdot m_R C_1 + & + r_{kr} \cdot m_R C_k \leq R_r \end{cases} \quad (3.28)$$

Svaka nejednadžba u (3.28) predočuje ograničenje uzrokovano raspoloživošću jednog resursa. Ovim je definiran i matematički zapisan skup ograničenja nad ukupnom razinom ostvarenosti izvedenih strateških ciljeva na temelju raspoloživosti resursa.

Sljedeći tip ograničenja proizlazi iz same strukture strategije. Ona je posljedica utjecaja među vrijednostima mjera. Razina ostvarenja niže pozicioniranih ciljeva preduvjet je ostvarenja učinka aktivnosti koje se provode u svrhu njima nadređenih ciljeva. Koeficijenti utjecaja mjera izvedeni su i prikazani izrazima (3.8) i (3.9). Vektor ukupne razine ostvarenosti  $n$  ciljeva opisan je izrazom (3.14).

Neka je MSM reda  $n$ , u kojem je utvrđeno  $i$  stupaca čije su sve vrijednosti jednake nuli, tj.  $i$  mjera na koje ne utječe ostvarenje nijednog drugog cilja ( $i$  upisnih mjera). To znači da postoji  $n-i$  izračunatih mjera, tj.  $n-i$  mjera čije su vrijednosti uvjetovane





Na ovaj način definiranjem:

- funkcije cilja opisane izrazom (3.31);
- ograničenja uzrokovanih raspoloživošću resursa opisanih izrazom (3.28);
- ograničenja uzrokovanih strukturom ciljeva opisanih izrazom (3.29) te
- uvjeta nenegativnosti i maksimalne vrijednosti mjere opisanog izrazom (3.30);

razmatrani problem sadrži sve potrebne elemente za primjenu linearnog programiranja u svrhu traženja optimalne strategije.<sup>1</sup> Dobiveni rezultat predočuje potrebne razine ostvarenosti ISC-a i na temelju njih maksimalne razine ostvarenosti ZSC-a. Množenjem optimiranih vrijednosti mjera s MR-om moguće je izračunati ukupni utrošak resursa prema aktivnosti. Potrebno je podsjetiti da koncept polazi od pretpostavljenih utjecaja među ciljevima. Na osnovi mjerenja može se pokazati drukčiji odnos među ciljevima. To će promijeniti i strukturu strategije, a samim time i ograničenja koja su iz nje proizašla. U tom je slučaju potrebno preispitati ograničenja i ponoviti postupak optimizacije.

#### 3.4 Upravljanje korištenjem modela

Prilikom određivanja strateških ciljeva menadžment, sukladno prethodno opisanom postupku, utvrđuje početno stanje (*as is*) i pretpostavlja željene vrijednosti budućeg stanja (*to be*) ostvarenosti ciljeva. Primjena matrične algebre u računu učinkovitosti organizacije koristi relativne vrijednosti mjera. Pretvaranje relativnih vrijednosti mjera u apsolutne i obratno jednostavno se ostvaruje pomoću formule (3.1). Na taj način moguće je relativne vrijednosti prikazati u apsolutnim vrijednostima mjera koje uz vrijednost sadrže i mjerne jedinice. Daljnja analiza rezultata, zbog konzistentnosti prikaza, temelji se na relativnim vrijednostima mjera.

Kao što je već rečeno, relativne vrijednosti mjera moraju biti iz segmenta  $[0,1]$ . Ako mjera poprimi vrijednosti izvan tog segmenta, potrebno je detaljnije istražiti okolnosti koje su do toga dovele. Na osnovi ostvarenih vrijednosti mjera procjenjuje se zadovoljstvo razinom ostvarenosti ciljeva. U skladu s tim, menadžment donosi odluke o budućim aktivnostima.

Prilikom upotrebe modela u svrhu strateškog upravljanja mjere se stvarne vrijednosti upisnih mjera. One se mjere tijekom cijelog strateškog ciklusa. Vrijednosti izračunatih mjera računaju se na osnovi prethodno opisanog matričnog modela.

---

<sup>1</sup> Zapis općeg slučaja standardnog problema linearnog programiranja s  $n$  varijabli i  $m$  ograničenja može se pogledati u Babić, Z. (2005).

Procjena zadovoljstva razinom ostvarenosti ciljeva radi se na osnovi triju područja:

1. loše stanje (crveno),
2. srednji uspjeh (žuto) i
3. ciljano ili željeno stanje (zeleno).

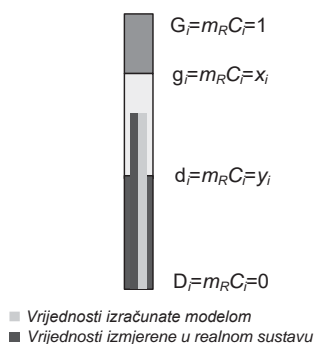
Kako bi se vrijednost mjere mogla rasporediti u neko od navedenih područja, potrebno je odrediti četiri granične vrijednosti. Prema tome, granične vrijednosti su vrijednosti unutar kojih se očekuju stvarne i izračunate vrijednosti mjera.

Relativne vrijednosti svih mjera na početku strateškog ciklusa, tj. u trenutku stanja *as is*, jednake su nuli. To znači da relativna vrijednost mjere koja je jednaka nuli označuje da se promjena nije ostvarila. Zbog toga je donja granica (D) lošeg stanja za svaku mjeru, izraženo relativnom vrijednošću, jednaka nuli. Relativna vrijednost mjere koja je jednaka jedan označuje da je promjena u potpunosti ostvarena, tj. ostvarenost željenog stanja *to be*. Zbog toga je gornja granica (G) najbolja očekivana vrijednost mjere i za svaku mjeru, izraženo relativnom vrijednošću, iznosi jedan.

Prethodno prikazan postupak optimizacije željene vrijednosti mjera usklađuje se s ograničenjima iz realnog sustava. Zbog toga se logički nameće da je donja granica željenog zadovoljstva određena realnim mogućnostima ostvarenja ciljeva. To znači da je svaka razina ostvarenosti ciljeva iznad realnih mogućnosti željeno stanje. Vrijednosti ispod realnih mogućnosti mogu pripadati području srednjeg uspjeha ili lošeg stanja. Granicu između lošeg stanja i srednjeg uspjeha utvrđuje menadžment s obzirom na pretpostavljeno zadovoljstvo vlasnika s razinom ostvarenosti ZSC-a.

Iz tog razloga je donju granicu srednjeg uspjeha (d) najjednostavnije opisati kao postotak ostvarenosti ZSC-a definiran gornjom granicom srednjeg uspjeha. Gornja granica srednjeg uspjeha (g) određena je vrijednostima mjera dobivenih optimizacijom. Razlog odstupanja od planirane razine ostvarenja ciljeva može biti neizvršavanje planiranih aktivnosti ili pogrešna pretpostavka uzročno-posljedičnih odnosa među ciljevima. U svrhu provjere valjanosti modela potrebno je izmjeriti sve vrijednosti mjera i usporediti ih s vrijednostima izračunatim pomoću modela. Ispravnim modelom izračunate vrijednosti mjera trebaju biti jednake izmjerenim vrijednostima mjera u realnom sustavu.

Grafički prikaz izmjerenih i izračunatih vrijednosti mjera moguće je prikazati kao na slici 3.4.



*Slika 3.4 Grafički prikaz razine ostvarenosti ciljeva*

Gornja (G) i donja (D) granica jednake su za sve mjere  $G=1$  i  $D=0$ . Srednja gornja granica (g) rezultat je optimizacije i za svaku mjeru može biti drukčija  $g=x_i$ . Donja srednja granica (d) određuje se na temelju zadovoljavajućeg postotka optimalne razine ostvarenosti ZSC  $d=y_i$ . Vrijednosti izračunate (sivo) i izmjerene (crno) mjere cilja  $C_i$  prikazane su stupcima.

U slučaju da se dogodi statistički signifikantno odstupanje ostvarenih i izmjerenih vrijednosti mjera, prvo je potrebno provjeriti izvršenost utvrđenih aktivnosti. Ako su planirane aktivnosti izvršene, provjerava se valjanost modela.

Provjera valjanosti modela zasniva se na činjenici da su modelom pretpostavljeni uzročno-posljedični odnosi među ciljevima. Ako je model valjan, onda će mjere niže pozicioniranih ciljeva dovesti do željene promjene mjera njima nadređenih ciljeva. Isto tako, ako zavisne mjere ne ostvaruju željene vrijednosti u skladu s utjecajima vrijednosti niže pozicioniranih mjera, a model se ponaša u skladu s očekivanim trendovima, tada je model formalno dobar. U tom je slučaju potrebno preispitati postavljene odnose među ciljevima.

Praćenje mjera kroz njihove trendove omogućuje identificiranje onih mjera koje nisu dostigle željenu vrijednost, ali tome teže. Očito je da, bez obzira na smjer promjena apsolutnih vrijednosti mjera, smjer promjena relativnih vrijednosti mjera mora imati uzlazni trend – od nula prema jedan. Smjer trenda apsolutnih vrijednosti određen je prilikom definiranja strateških ciljeva.

Analiza kretanja vrijednosti mjera radi se na kraju svakog razdoblja. Uspoređujući trendove svake pojedine mjere, moguće je uočiti približava li se organizacija ili udaljuje od željenog cilja. Nakon svakog mjerenja, odnosno izračuna, uspoređuju se dobiveni rezultati s rezultatima prethodnog mjerenja, odnosno izračuna. Očekivani smjer trenda podređene mjere mora rezultirati očekivanim trendom njoj nadređene mjere. U suprotnom se značenje trenda neke od mjera tumači s obzirom na područje

zadovoljstva u kojem je vrijednost mjere raspoređena. Kad se pojavljuje neočekivani trend, a vrijednost mjere je u području lošeg stanja, znači da je organizacija u „dubokoj“ krizi, a trendovi ne vode izlasku iz krize. U tom je slučaju potrebno poduzeti drastične aktivnosti za promjenu trendova. Ako se neočekivani trend pojavi kada je vrijednost mjere u području srednjeg uspjeha, znači da je moguće definirati aktivnosti za „izlazak iz krize“. Kada se pojave neočekivani trendovi, a vrijednost mjera je u željenom području, znači da „dolazi krizno razdoblje“ te je potrebno poduzeti aktivnosti kako organizacija ne bi ušla u krizu. U skladu s prethodnim tumačenjem utvrđuju se daljnje aktivnosti.

### 3.5 Optimizacija dinamike provedbe strategije IS-a/IT-ja primjenom sustavske dinamike

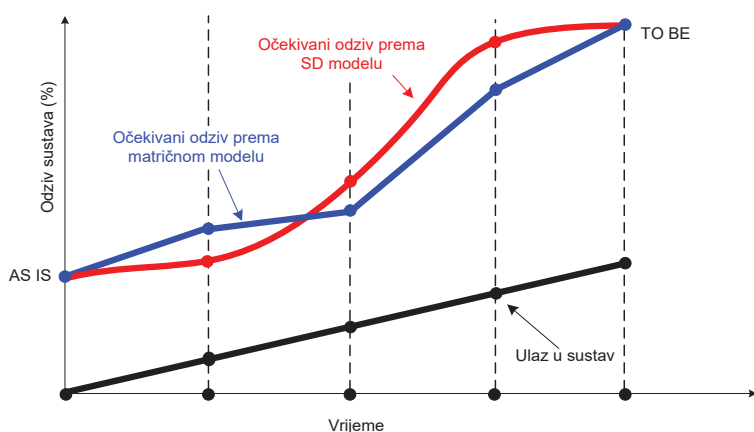
Objasnimo prvo jednostavnim rječnikom pojam optimizacija dinamike. Neke aktivnosti pokazuju efekte s određenim zakašnjenjem, primjerice poduzimanje aktivnosti za oglašavanje. Kad ostvarite sve aktivnosti koje su vezane za oglašavanje, potrebno je vrijeme da se pojave prvi kupci. Zbog toga se može koristiti metoda sustavske dinamike koja u svojim modelima uračunava kašnjenja. Iako ćemo se sa sustavskom dinamikom detaljnije baviti u predmetu simulacija poslovnih procesa, potrebno je naglasiti dva osnovna tipa veza. To su pozitivna i negativna veza kojom se opisuje proporcionalnost (+), odnosno obrnuta proporcionalnost (-) između dviju varijabli. Odnosi među varijablama prikazani su u strukturi *osnovnog simulacijskog modula* na slici 3.5. Osnovni simulacijski modul prikazuje sve elemente (varijable) realizacije aktivnosti koja troši neke resurse u svrhu ostvarenja nekog cilja koji je prikazan tim modulom.

Kao što je već rečeno, model organizacijskog sustava je uvijek njegovo pojednostavnjenje. To znači da se prilikom razvoja modela stvarni organizacijski sustav reducira na osnovna svojstva koja karakteriziraju interesno područje izučavanja. Model dinamičkog sustava je dobar ako dovoljno vjerno prikazuje ne samo trenutačna stanja sustava već i dinamiku promjene tih stanja. Prethodno opisani matrični model za potporu strateškom upravljanju je dinamički model. U skladu s opisanim postupkom on omogućuje upravljanje organizacijom kako bi se postigli strateški ciljevi. Ciklus strateškog plana podijeljen je u niz perioda. Po završetku svakog perioda provodi se analiza ostvarenih učinaka. Ostvarene vrijednosti mjera prezentiraju trenutačno stanje organizacije koje je posljedica poduzetih aktivnosti u proteklom razdoblju. Analizom trendova dobiva se podatak o promjenama promatranih mjera. Trendovi kretanja promjena mjera i njihove ostvarene vrijednosti omogućuju pravovremeno sagledavanje izvedivosti strateških ciljeva. S tim u skladu, na temelju modela određuju se aktivnosti koje trebaju u preostalom razdoblju putem lanaca uzroka i posljedica dovesti organizaciju u željeno stanje. Na taj način primjena

matričnog modela formalizira strateški plan koji odgovara budućem razdoblju razvoja organizacije.

Iz prethodnoga je lako uočiti da se matrični model zasniva na povijesnim podacima. Naime, vrijednosti mjera unutar jednog perioda računaju se na osnovi pretpostavljenih utjecaja s njegova početka. Drugim riječima, aproksimacija dinamike promjena stanja unutar jednog perioda je linearna (dužina omeđena vremenskim trenutcima početka i kraja perioda). Ovakva aproksimacija prema periodima može dovesti do odstupanja između vrijednosti mjera izračunatih modelom i njihovih izmjerenih vrijednosti iz stvarnog sustava. Odstupanja između vrijednosti mjera izračunatih matričnim modelom i izmjerenih vrijednosti u realnom sustavu registriraju se na kraju promatranog perioda. U skladu s registriranim odstupanjima vrši se korekcija modela kako bi on u sljedećim periodima dovoljno vjerno prikazao dinamiku promjene stanja. Ovakav način korištenja matričnog modela za potporu strateškom upravljanju organizacijskim sustavom dovoljno vjerno opisuje stanja i dinamiku njihove promjene. Matrični model polazi od pretpostavke da su svi resursi raspoloživi u trenutku kada su potrebni. Međutim, kod složenih sustava određeni resursi ne moraju biti odmah raspoloživi. To može prouzročiti ograničenja brzine provođenja aktivnosti. U tom je slučaju raspoloživost resursa potrebno promatrati u funkciji vremena. Jednako tako, ovisno o prirodi cilja može se očekivati da provođenje aktivnosti, kako bi se cilj ostvario, može imati veći učinak ako je veća razina ostvarenosti niže pozicioniranih ciljeva prema lancu uzroka i posljedica (LUP). To znači da je i intenzitet aktivnosti osim u funkciji mjera niže pozicioniranih ciljeva također potrebno promatrati u funkciji vremena. Neka je npr. matričnim modelom utvrđena potreba dvije tisuće radnih sati i dva zaposlenika za provođenje dviju aktivnosti. Prema matričnom modelu to je sto dana po deset sati rada svaki dan za svakog zaposlenika. Međutim, postavlja se pitanje je li u svrhu skraćivanja trajanja projekta bolje krenuti s obje aktivnosti istovremeno ili prvo oba zaposlenika angažirati na realizaciji prve aktivnosti pa ih potom oboje angažirati na drugoj aktivnosti. Naime, posljedica prve aktivnosti može nastupiti kasnije nego što je ona pokrenuta.

To znači da se javlja kašnjenje između trenutka pokretanja aktivnosti i njezine posljedice. Zbog toga je potrebno, ovisno o mogućem kašnjenju, pričekati s provođenjem posljedične aktivnosti. Proširenje analize dinamike promjene stanja u kontekstu sustavsko-dinamičkog pristupa (SD) dat će nova upotrebna saznanja u svrhu upravljanja vremenski promjenjivim veličinama. Odstupanje očekivanog odaziva sustava aproksimiranog matričnim i SD modelom moguće je slikovito prikazati kao na slici 3.5.



*Slika 3.5 Očekivani odziv sustava prema matričnom i SD modelu*

Na vremenskoj osi (os apscisa) označeni su trenutci u kojima se prema matričnom modelu provodi uzorkovanje razine ostvarenosti strateških ciljeva. Izbor vremenske jedinice ovisit će o trajanju ukupnog strateškog perioda. Kako bi se dobio opis trenutačnog ponašanja sustava, potrebno je utvrđeni period matričnim modelom smanjiti na dovoljno malen vremenski korak  $dt$ . Vremenski korak  $dt$  je korak simulacije, a određuje se u skladu s prirodom utvrđenih strateških ciljeva.

Dinamika ponašanja organizacijskog sustava promatra se ostvarenjem postavljenih strateških ciljeva. Zbog toga se odziv sustava (os ordinata) promatra kroz relativnu vrijednost (%) razine ostvarenosti pojedinog strateškog cilja. Dinamika ostvarenja ciljeva koji se nalaze na početku LUP-a ne ovise o ostvarenju nijednog drugog cilja. Iz tog razloga je dinamiku njihova ostvarenja moguće proizvoljno planirati tijekom ciklusa strateškog plana. Ciljevi na početku lanca uzroka i posljedica ujedno su ulaz u model dinamike ostvarenja ciljeva. Na slici je njihova dinamika opisana grafom crne boje. Očekivani odziv sustava prema matričnom modelu (graf plave boje) zasniva se na linearnim aproksimacijama po periodima, tj. jednolikoj brzini potrošnje resursa i ostvarenja niže pozicioniranih ciljeva u LUP-u. Kod SD modela period aproksimacije svodi se na minimum. Na taj način omogućeno je uračunavanje kašnjenja koje je uzrokovano trenutačnom dostupnošću resursa i kašnjenjem efekta ostvarenosti niže pozicioniranih ciljeva. Vjerojatno je da će u tom slučaju ponašanje sustava odgovarati krivulji (graf crvene boje). Mogućnost upravljanja brzinama potrošnje resursa i provedbe aktivnosti može rezultirati skraćanjem vremena postizanja željene razine ostvarenosti strateških ciljeva.

Ovakva analiza dinamike provedbe strateškog plana nije obuhvaćena računom konačnog utjecaja na kojem se zasniva račun matričnog modela. U matričnom modelu ona je prepuštena intuitivnom zaključivanju menadžera prilikom modeliranja utjecaja

među ciljevima (na kraju svakog razdoblja). Zbog toga je za razvijeni matični model napravljeno proširenje analize dinamike provedbe strateškog plana. Drukčije rečeno, proširenje matičnog modela zasniva se na promatranju dinamike provođenja aktivnosti i potrošnje resursa kao funkcija ovisnih o vremenu. Ono će rezultirati SD modelom za potporu strateškom upravljanju čija primjena treba omogućiti optimalnu dinamiku provođenja aktivnosti i alokaciju resursa u svrhu minimalizacije vremena postizanja razina ostvarenosti ciljeva utvrđenih matičnim modelom. Najprikladnija metodologija za ovakvo promatranje dinamike provedbe strateškog plana je sustavska dinamika.

Dosadašnji pristup modeliranju strateškog upravljanja organizacijom u kontekstu ovog rada ukazao je na važnost promatranja organizacijskog sustava kao cjeline. Jedino tako moguće je ostvariti sinergijski efekt unutar organizacije kojim se povećava njezina učinkovitost. Kako bi se održao sinergijski efekt tijekom cijelog strateškog ciklusa, potrebno je kontinuirano upravljanje tijekom resursa i aktivnosti u realnom sustavu koji dovode do ostvarenja matičnim modelom utvrđenih ciljeva. U tu je svrhu potrebno odabrati elemente matičnog modela koji će se preslikati u objekte SD modela.

Vrijednosti mjera iz matičnog modela u SD modelu predočuju stanja organizacijskog sustava. S obzirom na ograničenja uzrokovana ukupnom raspoloživošću resursa i utvrđenom strukturom ciljeva, primjenom matičnog modela optimira se razina ostvarenosti ISC-a u cilju maksimiranja razina ostvarenosti ZSC-a. Vrijednosti polja u matrici resursa predočuju ukupni utrošak resursa za 100 % ostvarenje ISC-a zbog koje se aktivnost provodi. Ova vrijednost u SD modelu nije promjenjiva, zbog čega je predstavljena konstantom. Prema matrici resursa i optimalnim vrijednostima mjera, matičnim modelom izračunava se dodijeljena količina resursa za provedbu svake aktivnosti. To znači da je planirani utrošak resursa za postizanje optimalne vrijednosti mjera utvrđen matičnim modelom. Zbog toga se i ovim vrijednostima u SD modelu pridružuje objekt konstante. SD model u svom računu koristi relativne vrijednosti mjera. Zbog toga vrijednosti početnog i konačnog stanja nisu potrebne u SD modelu, osim ako se prilikom generiranja izvješća o ostvarenju ciljeva ne zahtijeva apsolutni prikaz vrijednosti. Tada se početno i željeno stanje preuzima iz matičnog modela. Koeficijenti utjecaja (elementi MSM-a) u matičnom modelu predočuju težinu kojom vrijednost uzročne mjere utječe na vrijednost posljedične mjere. Kod matičnog modela koeficijenti utjecaja usklađuju se s izmjerenim rezultatima s kraja razdoblja, tj. promjenjivi su. Razlog tome je što matičnim modelom nije moguće uračunati kašnjenja između uzroka i posljedice unutar realnog sustava. Naime, u realnim sustavima posljedice promjena mogu nastupiti kasnije nego što su promjene pokrenute, zbog čega u matičnom modelu može doći do potrebe za promjenom koeficijenata utjecaja. Osnovni koncept SD modela omogućuje uračunavanje



kašnjenja. Koeficijent utjecaja u SD modelu predočuje udio planiranog ostvarenja uzročnog cilja u opsegu promjene planirane razine ostvarenja posljedičnog cilja. Zbog toga je koeficijent utjecaja u SD modelu prikazan konstantom, dok se posljedice promjena računaju kroz preduvjete za pokretanje aktivnosti kojima se ostvaruje posljedični cilj. Više o tome rečeno je kod modeliranja dinamike aktivnosti i ostvarenja ciljeva. Preslikavanje objekata iz matričnog u SD model moguće je prikazati tablicom 3.7. U slučaju rekonstrukcije strateških mapa potrebno je prilagoditi i cjelokupni SD model, a samim time i vrijednosti ulaznih varijabli izračunatih rekonstruiranim matričnim modelom.

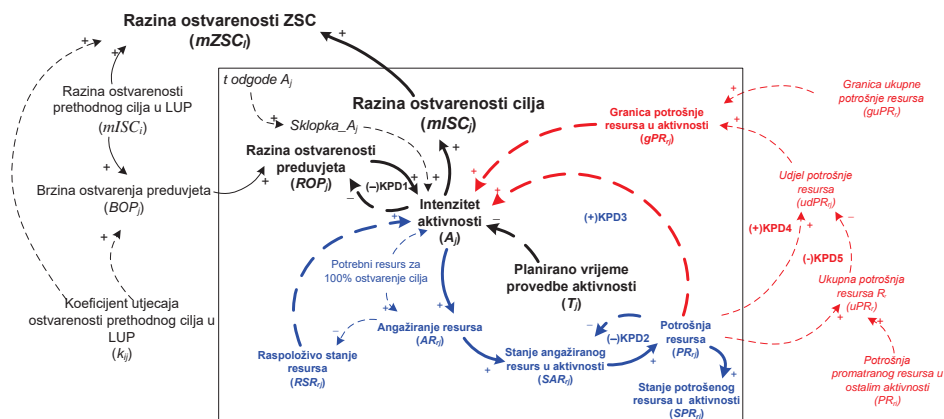
Tablica 3.7 Preslikavanja objekata iz matričnog u SD model

Matrični model		SD model	
Objašnjenje		Objašnjenje	Vrsta objekta
Mjera (razina ostvarenosti cilja)	$\mapsto$	Mjera (razina ostvarenosti cilja)	Varijabla stanja
Dodijeljena količina resursa za optimalnu razinu ostvarenosti cilja	$\mapsto$	Stanje raspoloživog resursa	Konstanta
Potreba resursa za 100 % ostvarenje cilja	$\mapsto$	Potreba resursa za 100 % ostvarenje cilja	Konstanta
Koeficijent utjecaja	$\mapsto$	Koeficijent utjecaja	Konstanta

Primjena sustavske dinamike polazi od opisa sustava pomoću uzročno-posljedičnih veza (UPV) među objektima modela. U sljedećem opisu SD sustava uočeni objekti označeni su kurzivom, dok se oznake varijabli koje se koriste u matematičkom modelu nalaze u zagradama. U sljedećem tekstu opisan je model dinamike provođenja aktivnosti i ostvarenja ciljeva. Polazište opisa su niže pozicionirani ciljevi koji putem brzine provođenja aktivnosti utječu na ostvarenje posljedičnog cilja prema LUP-u. Ovisno o broju ciljeva ovisit će kompleksnost SD modela. U svrhu pojednostavnjenja razvoja SD model podijeljen je u module. Svaki izvedeni strateški cilj predočen je jednim modulom, koji se naziva *osnovni simulacijski modul* (OSM). Ovakav način specijalizacije polazi od činjenice da se ostvarenje svakog izvedenog strateškog cilja zasniva na istom načelu. To znači da svaka aktivnost zahtijeva vrijeme, resurse i ispunjene preduvjete kako bi ostvarila postavljeni cilj. Iz prethodne analize utjecaja proizlazi da je postupak određivanja intenziteta aktivnosti također moguće generalizirati. Osnovni simulacijski model moguće je prikazati strukturnim dijagramom kao na slici 3.6.

Prema sustavskom pristupu u kontekstu promatranog problema, OSM predstavlja jedan podsustav SD modela promatranog organizacijskog sustava. Ostali dijelovi u sustavu sačinjavaju njegovu aktivnu okolinu. Upravo ta okolina generira ograničenja ostvarenja njegove funkcije, a to je ostvarenje željenog stanja cilja. Zbog toga je

njegovo djelovanje potrebno promatrati u interakciji s cijelim sustavom. Ostale dijelove okoline sačinjavaju podsustav resursa i ostali OSM-ovi drugih izvedenih strateških ciljeva. Model funkcije cilja sustava, tj. dinamika ostvarenja ZSC-a, također je sastavni dio okoline OSM-a.



Slika 3.6 Očekivani odaziv sustava prema matričnom i SD modelu

Definiranjem osnovnog simulacijskog modula pojednostavnjen je postupak modeliranja promatranog problema. Za potrebe ispitivanja skraćenja vremena ukupnog trajanja strateškog ciklusa potrebno je za svaku aktivnost definirati minimalno vrijeme potrebno za postizanje 100 % vrijednosti cilja. Provođenjem simulacija i ispitivanja mogućih scenarija moguće je odrediti optimalno trajanje svake aktivnosti. Iz tog razloga je potrebno korištenje gotovih programskih paketa za simulacijsko modeliranje.

### 3.6 Formalizacija operative upotrebe matematičkih modela

Predloženim pristupom potrebno je formalizirati viziju u oblik zadanih strateških ciljeva. Time je stvoren radni okvir u kojem je za zadane strateške ciljeve potrebno redom:

- procijeniti spremnost organizacije za njihovo ostvarenje,
- odrediti potrebne strategije s obzirom na spremnost organizacije,
- odrediti aktivnosti kojima će se strategije operacionalizirati i ciljeve kojima će se mjeriti njihova provedenost,
- strukturirati odnose među strateškim ciljevima,
- optimalno raspodijeliti resurse u skladu s utvrđenom strukturom,
- utvrditi dinamiku provođenja aktivnosti.

U svrhu lakše implementacije i korištenja razvijenih (matričnog i SD) modela BSC-a za strateško upravljanje organizacijom, u daljnjem tekstu prikazan je model tijekom aktivnosti AFD-a (engl. *activity flow diagram*). Ovaj model objašnjava operativnu upotrebu razvijenih modela BSC-a. Njegovi su elementi: *radna mjesta*, kojima su opisane funkcionalne uloge (vlasnik, menadžer i dizajner modela) uključene u operativnu upotrebu modela; *aktivnosti*, koje su skup radnji i odluka koje koriste funkcionalne uloge; i *informacijski tijekovi*, kojima se opisuju veze između aktivnosti.

Sve aktivnosti popisane su i opisane u tablici 3.8. U prvom stupcu abecednim su popisom navedeni nazivi aktivnosti, dok su u drugom stupcu prikazani njihovi opisi.

*Tablica 3.8 Naziv i opis aktivnosti operativne upotrebe matričnog i SD modela*

Naziv aktivnosti	Opis aktivnosti
<b>Analiza kašnjenja efekata ostvarenosti ciljeva</b>	U slučaju potvrde neispravnosti SD modela, prvo se provjeravaju pretpostavljena kašnjenja između uzročnih i posljedičnih ciljeva. Ispitivanje ispravnosti pretpostavljenih kašnjenja radi se uspoređivanjem s povijesnim podacima (vrijednosti iz proteklog razdoblja). Ako se vrijednosti mjera rekonstruiranog modela statistički podudaraju s (povijesnom) dinamikom ostvarenja ciljeva u realnom sustavu, može se smatrati da je model ispravan. Kao takav, može se ponovno koristiti za potporu strateškom upravljanju. Ako nije moguće utvrditi odgovarajuća kašnjenja kojima bi se opisala (povijesna) dinamika ostvarenja ciljeva iz realnog sustava, potrebno je provesti analizu pretpostavljenih težina utjecaja među ciljevima.
<b>Analiza težinâ utjecaja među ciljevima</b>	Razvijeni modeli polaze od pretpostavljenih težina utjecaja među ciljevima. Pod pretpostavkom da su planirane aktivnosti izvedene u potpunosti, a analiza kašnjenja efekata ostvarenosti ciljeva nije polučila željene rezultate, potrebno je preispitati pretpostavljene težine utjecaja među ciljevima. Promjenom težinskih koeficijenta u skladu s logikom promatranog problema ispituje se može li razvijeni model opisati izmjerenu razinu ostvarenosti ciljeva.  Ako je moguće pronaći odgovarajuće težinske koeficijente, potrebno je odrediti veze među mjerama i model podvrgnuti novoj optimizaciji. U suprotnom je potrebno preispitati postavljene strategije i definirati nove aktivnosti kako bi se ostvarili željeni ciljevi. Razlog odstupanja vrijednosti modela i stvarnog sustava moguće je tražiti i u promijenjenoj okolini. Iz tog razloga se potrebno vratiti na aktivnost za iznalaženje elementa SWOT-a i ponoviti cjelokupni postupak razvoja modela.

### 3: Upravljanje provedbom strategije razvoja informacijskih sustava

<b>Definiranje aktivnosti</b>	Konkretizacija utvrđenih strategija formalizira se u obliku konkretnih aktivnosti za njezinu provedbu. Formalizirani oblik aktivnosti u obliku indeksa i njezina opisa zapisuje se u matricu aktivnosti i izvedenih strateških ciljeva. Svaka aktivnost ostvaruje neki cilj i troši raspoložive resurse. Iz tog razloga je u nastavku iste tablice potrebno dopisati ciljeve kojima se mjeri provedba aktivnosti.
<b>Definiranje funkcije cilja</b>	Zadani strateški ciljevi predočuju ključne ciljeve organizacije jer se kroz njih ostvaruje vizija organizacije. Zbog toga je funkcija cilja ponderirana suma mjera zadanih strateških ciljeva. Ovako definirana funkcija cilja poslužit će za postupak optimizacije razine ostvarenosti ostalih strateških ciljeva.
<b>Definiranje ograničenja mjera</b>	Sukladno jačini utjecaja među mjerama, ovisit će i ostvarenje svih posljedičnih ciljeva u lancu uzroka i posljedica. To znači da razina ostvarenosti nekog cilja ovisi o razini ostvarenosti cilja koji je niže pozicioniran u lancu uzroka i posljedica. Zbog toga je moguće definirati skup ograničenja uzrokovan razinom ostvarenosti ciljeva. Skup ograničenja izvodi se iz matrice strukture mjera, a koristi se u postupku optimizacije razine ostvarenosti strateških ciljeva.
<b>Definiranje strategijâ</b>	Na osnovi utvrđenih sudova o stanju organizacije i njezine okoline, formira se matrica SWOT, na temelju koje se generiraju strategije za usmjeravanje organizacijskih procesa prema ostvarenju zadanih strateških ciljeva. Strategije se izvode uparivanjem sudova (S,O), (S,W), (S,T), (O,W) i (O,T). Forma strategije mora sadržavati tip strategije i njezin opis. Ovako utvrđene strategije potrebno je konkretizirati u obliku aktivnosti i njima pripadajućih ciljeva.
<b>Definiranje strateških ciljeva i njihovih mjera</b>	Definiranjem strateških ciljeva menadžment formalizira viziju u oblik mjerljivih veličina kojima će se mjeriti razina ostvarenosti vizije. Svaki strateški cilj mora sadržavati oznaku, smjer promjene, objekt promjene, oznaku mjere, početno ( <i>as is</i> ), željeno ( <i>to be</i> ) stanje i mjernu jedinicu. Prijedlog strateških ciljeva dostavlja se na potvrdu vlasnicima.
<b>Definiranje vizije</b>	U svrhu što kvalitetnijeg ispunjenja misije menadžment predlaže buduću sliku organizacije i smjernice njezina ostvarenja. Vizija mora biti sažeto i jasno izrečena kako bi bila razumljiva i lako prevodljiva u mjerljive veličine. Definirana vizija dostavlja se u obliku prijedloga na potvrdu vlasnicima.

<b>Iznalaženje potrebnih resursa</b>	Svaka aktivnost troši resurse. U svrhu optimizacije razine ostvarenosti ciljeva potrebno je za svaku aktivnost utvrditi količinu resursa za ostvarenje željenog ( <i>to be</i> ) stanja cilja zbog kojeg se aktivnost provodi.
<b>Iznalaženje elemenata SWOT-a</b>	Definiranjem zadanih strateških ciljeva utvrđena su buduća stanja (mjerljive vrijednosti) koja organizacija želi dostići. Kako bi se ustanovila spremnost organizacije za ostvarenje zadanih strateških ciljeva, potrebno je utvrditi elemente SWOT-a (S – prednosti, O – pogodnosti, W – nedostaci, T – prijetnje). Elementi SWOT-a poslužit će za utvrđivanje sudova o trenutačnom stanju organizacije i okoline u kojoj ona djeluje te definiranje strategije kojom će se ostvariti zadani strateški ciljevi. Svaki sud formalizira se oznakom i opisom.  U kontekstu SPIS-a, prilikama i prijetnjama procjenjuju se učinci nove tehnologije, dok se prednostima i nedostacima procjenjuje spremnost organizacije za njihovo prihvaćanje.
<b>Oblikovanje matričnog modela BSC-a</b>	Utvrđeni skup aktivnosti i svih strateških ciljeva potrebno je oblikovati u prikaz prilagođen korisniku. Dizajner modela stoga u prikaz treba uključiti sljedeće: opis planiranih aktivnosti i ciljeva, područja zadovoljstva ostvarenjem ciljeva, trendove kretanja ostvarenosti ciljeva i mogućnost analize pretpostavljenih utjecaja. Ovako oblikovan matrični prikaz pruža cjeloviti uvid u trenutačno stanje i buduća stanja prema razdobljima. Kao takav daje se na potvrdu menadžeru. Primjena prihvaćenog matričnog modela omogućuje njegovu upotrebu kao sustava za potporu strateškom upravljanju.
<b>Oblikovanje SD modela BSC-a</b>	Na osnovi prihvaćenog matričnog modela razvija se SD model. SD modelom moguće je obuhvatiti kašnjenja i ograničenja brzine trošenja resursa te na toj osnovi izvršiti optimizaciju dinamike provođenja aktivnosti. Iz tog razloga je potrebno na temelju optimalnih vrijednosti iz matričnog modela razviti simulacijske scenarije te odabrati najpovoljniji. Cilj koji se želi ostvariti SD modelom je skraćivanje vremena strateškog ciklusa.
<b>Očitavanje vrijednosti mjera iz realnog sustava</b>	U svrhu provođenja validacije razvijenih modela potrebno je očitavanje vrijednosti mjera iz realnog sustava. Očitavanje modela može se automatizirati tako da se model poveže s pripadajućim izvorima informacija u postojećem informacijskom sustavu.
<b>Odobrenje matričnog modela BSC-a</b>	Razvijen matrični model pruža cjeloviti uvid u dinamiku odvijanja strateškog plana prema razdobljima. Na osnovi njega menadžment dobiva kompletnu sliku budućih stanja organizacije

### 3: Upravljanje provedbom strategije razvoja informacijskih sustava

	<p>na putu k ostvarenju ciljeva. Menadžment mora potvrditi očekivanu dinamiku strateškog plana. Zbog toga dizajner modela šalje matični model na prihvaćanje menadžerima. Ako vrijednosti modela nisu usklađene s željama menadžera, provodi se ponovna analiza kako bi se utvrdili novi elementi za ostvarenje želja vlasnika. Prihvaćanjem modela utvrđeno je i područje ciljanog stanja.</p>
<b>Odobrenje strateških ciljeva</b>	<p>Prijedlog strateških ciljeva potrebno je dostaviti na potvrdu vlasnika. Odobrenjem strateških ciljeva vlasnici potvrđuju mišljenje menadžmenta o usklađenosti ciljeva s vizijom. U slučaju neprihvaćanja, prijedlog se šalje na doradu. Usvajanjem prijedloga strateški ciljevi postaju ključni ciljevi organizacije zbog čega dobivaju pridjev zadani.</p>
<b>Odobrenje vizije</b>	<p>Vlasnik mora potvrditi viziju koju je predložila uprava. Vlasnici provjeravaju usklađenost prijedloga vizije s politikom razvoja. U slučaju neprihvaćanja, prijedlog se šalje na doradu. Odobranjem prijedloga vizije vlasnici prihvaćaju mišljenje menadžmenta o smjeru budućeg razvoja organizacije što je osnova za određivanje zadanih strateških ciljeva</p>
<b>Određivanje ograničenja resursa</b>	<p>Jedan resurs moguće je koristiti u više aktivnosti. Međutim, organizacija se mora mjeriti s raspoloživim količinama resursa. Iz tog razloga je potrebno, nakon definiranja količine potrebnih resursa, utvrditi njihovu raspoloživost. U skladu s tim se postavljaju ograničenja potrošnje resursa, na osnovi kojih se provodi optimizacija razine ostvarenosti strateških ciljeva.</p>
<b>Određivanje ostalih ciljeva i mjera</b>	<p>Provođenjem aktivnosti ostvaruju se pripadajući ciljevi. Kako su ovi ciljevi izvedeni iz strategija i aktivnosti za njihovu provedbu, nazivaju se izvedeni strateški ciljevi. Izvedeni strateški ciljevi moraju imati isti semantički opis kao i zadani strateški ciljevi.</p>
<b>Određivanje veza među ciljevima</b>	<p>Svaka utvrđena aktivnost vodi k ostvarenju pripadajućeg cilja koji bi trebao približiti organizaciju željenom stanju, tj. ostvarenju zadanih strateških ciljeva. To znači da ostvarenje nekog od ciljeva, prema strategiji iz koje je posredno izveden, treba imati utjecaj na ostvarenje nekog zadanog strateškog cilja. Iz tog razloga je potrebno u tablici strukture ciljeva utvrditi na koje sve druge ciljeve utječe njegovo ostvarenje. Na taj način utvrđuju se lanci uzroka i posljedica među strateškim ciljevima. Forma prikaza odnosa među ciljevima je matrica strukture ciljeva, čiji elementi predočuju intenzitet kojim uzročni cilj doprinosi ostvarenju posljedičnog cilja. Grafički prikaz svih lanaca uzroka i posljedica je</p>

	strateška mapa ciljeva. Na temelju matrice strukture ciljeva formira se matrica strukture mjera.
<b>Određivanje veza među mjerama</b>	Strukturi ciljeva odgovara struktura pripadajućih mjera. To znači da, ako postoji veza između dvaju ciljeva, mora postojati i veza između njihovih mjera. Elementi matrice strukture mjera prikazuju težine utjecaja vrijednosti mjere uzročnog cilja na vrijednost mjere posljedičnog cilja. Matrica strukture mjera određuje se normiranjem matrice strukture ciljeva po njezinim stupcima.
<b>Optimizacija ciljeva</b>	Želje vlasnika određene su funkcijom cilja, dok su mogućnosti određene ograničenjima proizašlim iz raspoloživih resursa i strukture mjera. Tako postavljeni problem je klasični problem maksimuma linearnog programiranja. Rješenje problema dat će optimalne vrijednosti mjera i alokacije resursa u svrhu maksimiranja funkcije cilja. Optimalne vrijednosti mjera predočuju donju granicu područja uspjeha organizacije.
<b>Upravljanje korištenjem modela</b>	Koristeći razvijene modele, menadžment upravlja organizacijom i vodi je prema ostvarenju strateških ciljeva. Provođenjem simulacijskih scenarija menadžment je u mogućnosti analizirati buduće događaje, što pojednostavnjuje proces strateškog upravljanja organizacijom.
<b>Utvrdjivanje politike razvoja</b>	Na osnovi izvješća o ostvarenosti ciljeva vlasnici utvrđuju buduću politiku razvoja.
<b>Validiranje modela</b>	Pod pretpostavkom ostvarenja planiranih aktivnosti u realnom sustavu, odstupanje vrijednosti modela od očitanih vrijednosti iz stvarnog sustava može biti uzrokovano kašnjenjem efekata među uzročnim i posljedičnim ciljevima ili krivo pretpostavljenim težinama utjecaja ( <i>ako utjecaj među ciljevima ne postoji, vrijednost utjecaja jednaka je nuli</i> ). Zbog toga je potrebno provoditi validaciju modela. Ako je model održiv, tj. odstupanje modeliranog i stvarnog statistički nije značajno, model se može koristiti kao sustav za potporu strateškom upravljanju. Zbog toga je ispravan model ujedno potvrda dobro odabranih strategija, aktivnosti i ciljeva te postavljenih veza među njima. Održivost modela formalizirana je u obliku potvrde o ispravnosti modela. Ako model nije održiv, potrebno je analizirati razloge odstupanja. Neodrživost modela formalizirana je u obliku potvrde o neispravnosti modela.

Za odvijanje aktivnosti prikazanih u tablici 3.8 potrebne su informacije koje se generiraju nekom drugom aktivnošću. Na taj način dobivaju se tijekom informacija.

Informacijski tijekovi koje koriste i generiraju aktivnosti u operativnoj upotrebi modela prikazani su u tablici 3.9. U prvom stupcu tablice navedeni su nazivi tijekova poredani abecednim redom, dok je u drugom stupcu tablice dan njihov opis.

*Tablica 3.9 Popis i opis tijekova u operativnoj upotrebi matričnog i SD modela*

Naziv tijeka	Opis tijeka
<b>Aktivnosti</b>	Aktivnost je konkretizacija jedne ili više strategija. Svaka aktivnost ima pripadajući cilj i provodi se radi ostvarenja tog cilja.
<b>Funkcija cilja</b>	Funkcija cilja je linearna kombinacija mjera zadanih strateških ciljeva. Koeficijenti uz mjere predočuju težinske vrijednosti kojima zadani strateški cilj utječe na ostvarenje vizije. Suma koeficijenta je jednaka jedan. Funkcija cilja osnovni je element problema maksimuma linearnog programiranja.
<b>Izvedeni strateški ciljevi (ISC)</b>	Izvedeni strateški ciljevi su oni strateški ciljevi koji su izvedeni iz strategija i pripadajućih aktivnosti. Svaki izvedeni strateški cilj mora sadržavati oznaku cilja, smjer promjene, objekt promjene, naziv mjere, oznaku mjere, vrijednosti <i>as is</i> i <i>to be</i> i mjernu jedinicu.
<b>Izvešće o realizaciji ciljeva</b>	Izvešće dobiveno iz modela BSC-a koje menadžment dostavlja vlasnicima u svrhu utvrđivanja buduće politike razvoja organizacije.
<b>Matrica sume utjecaja (MSU)</b>	Matrica sume utjecaja je matrični zapis ukupnog (direktnog i indirektnog) utjecaja među mjerama. Elementi matrice predočuju ukupni utjecaj vrijednosti mjere prikazane u retku na vrijednost mjere prikazane u stupcu. Matrica sume utjecaja koristi se u matričnom modelu za računanje ukupne razine ostvarenosti ciljeva.
<b>Matrica strukture ciljeva (MSC)</b>	Matrica strukture ciljeva je matrični zapis svih direktnih utjecaja među ciljevima. Elementi matrice predstavljaju jačinu direktnog utjecaja cilja zapisanog u retku na cilj zapisan u stupcu. Nepostojanje direktnog utjecaja između dva cilja označuje se vrijednošću elementa matrice nula. Matrica strukture utjecaja svedena na donju trokutastu matricu predočuje matrični zapis strateške mape ciljeva, tj. grafičkog prikaza svih lanaca uzroka i posljedica.
<b>Matrica strukture mjera (MSM)</b>	Matrica strukture mjera je matrični zapis svih koeficijenta utjecaja među mjerama, tj. težinskih vrijednosti kojima vrijednost mjere zapisane u retku utječe na vrijednost mjere zapisane u stupcu.



	<p>Matrica strukture mjera izvodi se iz matrice strukture ciljeva tako da se ona normira po stupcima. Kao takva, ona je matični zapis strateške mape mjera.</p>
<p><b>Matični model BSC-a</b></p>	<p>Matični model BSC-a je model koji omogućuje menadžmentu planiranje, optimiranje, organiziranje i kontrolu ostvarenja strateških ciljeva u vremenu. U tom kontekstu, to je sustav za potporu strateškom upravljanju.</p>
<p><b>Neostvarive strategije</b></p>	<p>Ako nije moguće ustanoviti ni nove vrijednosti kašnjenja, ni nove vrijednosti direktnog utjecaja ciljeva, moguće je konstatirati da postoje neostvarive strategije. Neostvarive strategije potrebno je promijeniti ili ih konkretizirati u obliku novih aktivnosti.</p>
<p><b>Nove vrijednosti direktnog utjecaja ciljeva</b></p>	<p>Ako nije moguće ustanoviti nove vrijednosti kašnjenja kojima će se dovoljno vjerno opisati dinamika ostvarenja ciljeva, potrebno je ustanoviti nove vrijednosti direktnog utjecaja ciljeva. Nove vrijednosti upisuju se u matricu strukture ciljeva, iz koje se automatizmom računaju koeficijenti utjecaja.</p>
<p><b>Nove vrijednosti kašnjenja</b></p>	<p>Na osnovi potvrde neispravnosti modela potrebno je ispitati razloge odstupanja modela od realnog ponašanja organizacije. Prvi korak je ispitivanje pretpostavljenih kašnjenja efekta ostvarenosti niže pozicioniranih ciljeva u lancu uzroka i posljedica. Ako je moguće, potrebno je ustanoviti nove vrijednosti kašnjenja na osnovi kojih se može opisati dotadašnje ponašanje organizacije.</p>
<p><b>Ograničenja uzrokovana raspoloživošću resursa</b></p>	<p>Ograničenja uzrokovana raspoloživošću resursa ograničavaju provođenje aktivnosti. Naime, svaka aktivnost troši određene resurse. Ako resurs nedostaje, aktivnost nije moguće provesti u potpunosti. Može se dogoditi da je ukupno potrebna količina resursa za provođenje svih aktivnosti manja od raspoložive količine resursa. Zbog toga je potrebno pronaći optimalne vrijednosti mjera izvedenih strateških ciljeva kojima se maksimiraju vrijednosti zadanih strateških ciljeva u skladu s raspoloživim resursima.</p>
<p><b>Ograničenja uzrokovana strukturom mjera</b></p>	<p>Ograničenja uzrokovana strukturom mjera proizlaze iz matrice strukture mjera. To znači da neostvarenje niže pozicioniranog cilja prema lancu uzroka i posljedica rezultira neostvarenjem više pozicioniranog cilja. Zbog toga je preduvjet pokretanja aktivnosti za ostvarenje posljedičnog cilja ostvarena razina niže pozicioniranog cilja. U prethodnom kontekstu moguće je definirati ograničenja uzrokovana strukturom mjera koja su zajedno s</p>

### 3: Upravljanje provedbom strategije razvoja informacijskih sustava

	ograničenjima uzrokovanim raspoloživošću resursa i funkcijom cilja klasični problem linearnog programiranja.
<b>Optimalne vrijednosti mjera</b>	Optimalne vrijednosti mjera su vrijednosti mjera izvedenih strateških ciljeva kojima se kroz promatrani period postiže maksimalna moguća vrijednost linearne kombinacije zadanih strateških ciljeva (funkcije cilja) s obzirom na ograničenja uzrokovana raspoloživim resursima i utvrđenom strukturom ciljeva. Optimalne vrijednosti računaju se metodom linearnog programiranja.
<b>Ostvarene vrijednosti mjera</b>	Ostvarene vrijednosti mjera očitavaju se iz realnog sustava kako bi se provela validacija modela.
<b>Politika razvoja</b>	Politika razvoja su okvirne smjernice razvoja organizacije koje utvrđuju vlasnici organizacije. Ona je osnovni kriterij za prihvaćanje prijedloga vizije. Vizija koju predlaže menadžment mora biti u okvirima politike razvoja.
<b>Potvrda ispravnosti modela</b>	Potvrđuje menadžmentu da je rezultat validacije modela pozitivan i da se model može koristiti kao sustav za potporu strateškom upravljanju.
<b>Potvrda neispravnosti modela</b>	Potvrđuje menadžmentu da je rezultat validacije modela negativan i da se model ne može koristiti kao sustav za potporu strateškom upravljanju te da ga je potrebno preispitati.
<b>Prijedlog strateških ciljeva</b>	Lista kvantificiranih strateških ciljeva formirana na osnovi vizije, a zahtijeva potvrdu vlasnika organizacije.
<b>Prijedlog vizije</b>	Prijedlog budućeg stanja organizacije koji daje menadžment prije potvrde vlasnika.
<b>Razlika modeliranog i stvarnog</b>	Razlika očitanih vrijednosti mjera i vrijednosti izračunatih modelom.
<b>Resursi</b>	Sve ono što je potrebno za provođenje aktivnosti i utječe na razinu ostvarenosti cilja zbog kojeg se aktivnost provodi. Za svaku aktivnost potrebno je identificirati količinu resursa za stopostotno ostvarenje pripadajućeg cilja. Na taj način formira se matrica resursa iz koje se izvode ograničenja za provođenje optimizacije.
<b>SD model BSC-a</b>	SD (sustavsko-dinamički) model BSC-a je model koji omogućuje planiranje, simuliranje, optimiranje, organiziranje i kontrolu brzine ostvarenja strateških ciljeva.

<b>Strategije</b>	Strategije su moguće ponašanje organizacije u svrhu ostvarenja zadanih strateških ciljeva. Rezultat su SWOT analize. U daljnjim koracima se konkretiziraju u obliku aktivnosti.
<b>Elementi SWOT-a</b>	<i>Strengths, Weakness, Opportunities i Threats</i> su elementi SWOT-a, tj. sudovi o trenutačnom stanju organizacije i njezine okoline. Njihovim uparivanjem (S,W), (S,O), (S,T), (O,W) i (O,T) utvrđuju se moguće strategije za ostvarenje zadanih strateških ciljeva koje se u daljnjim koracima konkretiziraju u obliku aktivnosti.
<b>Vizija</b>	Buduće stanje organizacije kako je vidi menadžment, koje je usklađeno s politikom razvoja i koje je potvrdio vlasnik.
<b>Zadani strateški ciljevi (ZSC)</b>	Kvantificirani strateški ciljevi izvedeni iz vizije na prijedlog menadžmenta, a koje je potvrdio vlasnik. Kao takvi, predočuju glavne ciljeve organizacije, zbog čega imaju pridjev zadani. Svaki zadani strateški cilj mora sadržavati oznaku cilja, smjer promjene, objekt promjene, naziv mjere, oznaku mjere, <i>as is</i> i <i>to be</i> vrijednosti te mjernu jedinicu.

Na slici 3.7 prikazan je dijagram tijeka aktivnosti za operativnu upotrebu razvijenog matričnog modela i SD modela. U modelu su definirana tri radna mjesta (funkcijska uloga): vlasnik, menadžer i dizajner modela. Oni izvode aktivnosti navedene u tablici 3.9. Aktivnosti su izvorišta ili odredišta informacijskih tijekova prikazanih u tablici 3.9. Informacijski tijekovi u objašnjenju prikazanom u daljnjem tekstu označeni su *kurzivom*, dok su aktivnosti označene *kurzivom i boldano*.

Na početku novog strateškog ciklusa menadžeri provode aktivnost **definiranje vizije**. *Prijedlog vizije* menadžment dostavlja vlasniku na odobrenje. Provođenjem aktivnosti *odobrenje vizije* vlasnici utvrđuju usklađenost *prijedloga vizije s politikom razvoja*. Vlasnik odobrava i dostavlja *viziju* menadžmentu za provođenje aktivnosti **definiranje strateških ciljeva**. U slučaju da vlasnik ne prihvati *prijedlog vizije*, šalje ga menadžmentu na doradu. **Definiranjem strateških ciljeva** formira se *prijedlog strateških ciljeva*. Provođenjem aktivnosti **odobrenje strateških ciljeva** vlasnik prihvaća *prijedlog strateških ciljeva* ili ga šalje na daljnju doradu. Prihvaćanjem *prijedloga strateških ciljeva* oni postaju **zadani strateški ciljevi** koje menadžment koristi u daljnjem postupku.

Prva sljedeća aktivnost je **iznalaženje elemenata SWOT-a**, koja rezultira elementima *S,W,O,T* na temelju kojih se provodi aktivnost **definiranje strategija**. Nakon što je menadžment definirao *strategije*, provodi se **definiranje aktivnosti**. Na taj se način *strategije* prevode u konkretne *aktivnosti* za njihovo ostvarenje. Za svaku *aktivnost*

dizajner modela provodi **određivanje ciljeva i mjera** za praćenje njezine provedbe. Na taj su način dobiveni *izvedeni strateški ciljevi*.

Provođenjem aktivnosti **određivanje veza među ciljevima** utvrđuje se *matrica strukture ciljeva*, koja se provedbom aktivnosti **određivanje veza među mjerama** prevodi u *matricu strukture mjera* i *matricu sume utjecaja*. Na osnovi *matrice strukture mjera* provodi se aktivnost **definiranje ograničenja nad mjerama**. Uz utvrđena *ograničenja uzrokovana strukturom mjera*, potrebno je utvrditi i ostala ograničenja.

Za utvrđene *aktivnosti* potrebno je **iznalaženje potrebnih resursa**. Tako se utvrđuju potrebni *resursi* za provođenje *aktivnosti* u svrhu stopostotnog ostvarenja *izvedenih strateških ciljeva*. Postoji mogućnost da raspoloživi resursi ne zadovoljavaju ukupne potrebe. Iz tog razloga je potrebno **određivanjem ograničenja resursa** za svaki *resurs* utvrditi *ograničenja uzrokovana raspoloživošću resursa*.

Aktivnošću **definiranje funkcije cilja** utvrđuju se koeficijenti linearne kombinacije *zadanih strateških ciljeva*. Na taj način definirana je *funkcija cilja*, *ograničenja uzrokovana raspoloživošću resursa* i *ograničenja uzrokovana strukturom mjera*, što je ulazni tijek aktivnosti **optimizacija ciljeva**. Izlazni tijek iste aktivnosti su *optimalne vrijednosti mjera* na osnovi kojih, uz ulazni tijek *matrica sume utjecaja*, dizajner modela provodi **oblikovanje matičnog modela BSC-a**. Dizajner modela dostavlja *matični model BSC-a* na odobrenje menadžeru. Ako elementi sadržani u *matičnom modelu BSC-a* ne zadovoljavaju menadžera, potrebno je ponoviti postupak aktivnosti **iznalaženje elementa SWOT-a** kako bi se pronašli novi elementi koji će dovesti bliže željenom stanju.

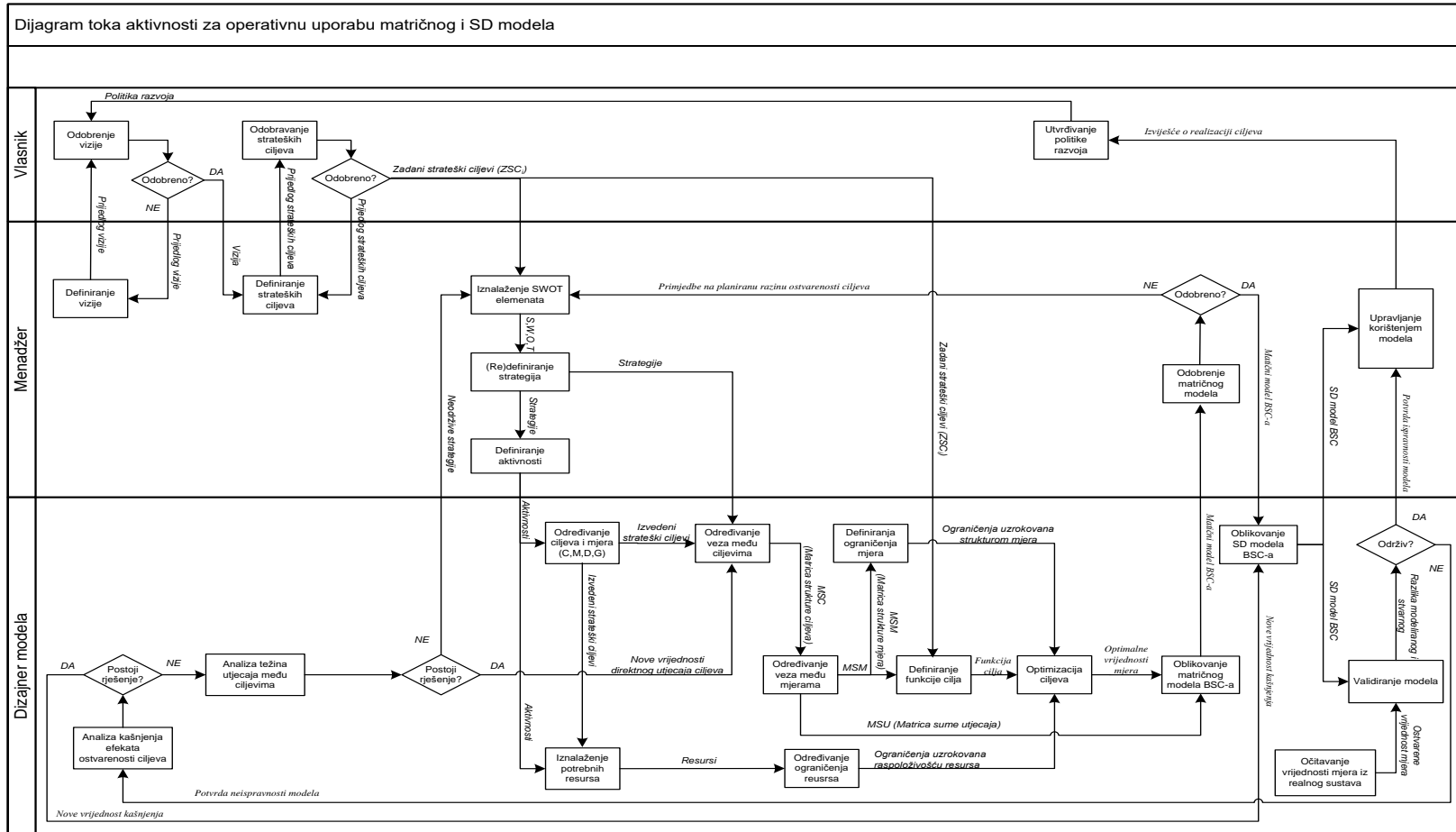
U suprotnom, potvrda prihvaćenosti *matičnog modela BSC-a* omogućuje dizajneru modela da pristupi aktivnosti **oblikovanja SD modela BSC-a**. *SD model BSC-a* koristi se u aktivnosti **upravljanje korištenjem modela** ako postoji *potvrda ispravnosti modela*. Rezultati **upravljanja korištenjem modela** dostavljaju se vlasnicima u obliku *izvješća o realizaciji ciljeva*. Na temelju *izvješća o realizaciji ciljeva* vlasnici **definiraju politiku razvoja** u obliku informacijskog tijeka *politike razvoja*. *Politika razvoja* koristi se u aktivnosti **odobrenje vizije** za provjeru usklađenosti *prijedloga vizije* koju je dostavio menadžment.

*SD modelu BSC-a* potrebna je verifikacija. **Očitavanjem vrijednosti mjera iz realnog sustava** dobivaju se *ostvarene vrijednosti mjera*. Verifikacijom *modela* utvrđuje se *razlika modeliranog i stvarnog*. Održivost modela prema menadžmentu formalizira se *potvrdom o ispravnosti modela*. U suprotnom, *potvrdom o neispravnosti modela* zahtijeva se **analiza kašnjenja efekata ostvarenosti ciljeva**. Pozitivni rezultat analize karakterizira postojanje *novih vrijednosti kašnjenja*, kojima je moguće opisati dotadašnje ponašanje organizacije. *Nove vrijednosti kašnjenja* uključuju se u model

kroz aktivnost **oblikovanje SD modela**. Negativni rezultat analize karakterizira nepostojanje rješenja.

U tom slučaju potrebno je provesti **analizu težinâ utjecaja među ciljevima**. Pozitivni rezultat ove analize karakterizira postojanje *novih vrijednosti direktnog utjecaja ciljeva* koje se uključuju u aktivnost **određivanje veza među ciljevima**, a to znači i nova *matrica strukture ciljeva*. Zbog toga je potrebno ponoviti postupak s novom *matricom strukture ciljeva*. U slučaju da ni ova analiza ne bi rezultirala pozitivnim rješenjem, znači da postoje *neodržive strategije*. Razlog postojanja neodrživih strategija može biti i u promijenjenoj okolini. Iz tog razloga je potrebno, vodeći se *neodrživim strategijama*, iznaći nove elemente SWOT-a koji će doprinijeti njihovoj održivosti.

### 3: Upravljanje provedbom strategije razvoja informacijskih sustava



Slika 3.7 Dijagram tijeka aktivnosti za operativnu upotrebu matričnog i SD modela

### 3.7 Literatura

1. Bakić, D. (2004). *Linearna algebra*. Golden marketing – Tehnička knjiga.
2. Dobrović, Ž., Tomičić, F. M. (2020). SWOT Analysis in the Strategic Planning Process - Meta-modelling Approach, in *IEEE 2020 IEEE 10th International Conference on Intelligent Systems (IS)* pp. 574–579. doi:10.1109/IS48319.2020.9199983
3. Dobrović, Ž. (2002). Strategic Planning under Uncertainty: Building the metamodel, *Journal of Information and Organizational Sciences*, vol. 25, nr. 1, 2002, pp.11–26.
4. Hell, M. i Dragičević, T. (2014). How to Use Linear Programming for Information System Performances Optimization. *Business Systems Research*, 5 (3), 49–66. <https://doi.org/10.2478/bsrj-2014-0019>
5. Hell, M., Vidačić, S., & Garača, Ž. (2009). Methodological Approach to Strategic Performance Optimization. *Management*, Vol. 14, pp. 21–42.
6. Hell, M., Vidačić, S., & Brumec, J. (2013). Dynamic aspects of the strategic planning of information systems, *Journal of Information and Organizational Sciences*, 37(1), pp. 23–43.
7. Hell, M. (2009). *Dinamički aspekti metodike strateškog planiranja informacijskog sustava*. [doktorska disertacija, Fakultet organizacije i informatike]. Varaždin
8. Kaplan, R. S., Norton, D. P. (2000). *The Strategy-Focused Organization: How Balanced Scorecard Companies Thrive in the New Business Environment* - 1st edn., USA: Harvard Business Review





## 4 Pristup i metode razvoja informacijskih sustava

Cilj je ovog poglavlja približiti čitatelju proces razvoja IS-a, kao i preduvjete za njegovu uspješnu realizaciju. Neka opća pitanja i dileme s kojima se susreću menadžeri prilikom informatizacije organizacije jesu:

- Zašto planirati razvoj IS-a?
- Koliko dugo je potrebno ulagati u IS?
- Kako pristupiti problemu?
- Je li je naša organizacija uopće spremna prihvatiti nove IT-jeve?
- Tko će izvršiti informatizaciju?
- Možemo li to napraviti sami?

Odgovor na prva dva pitanja sažet je u prethodnim poglavljima. Možemo reći da smo do sada naučili kako odrediti optimalnu ulogu IS-a u kontekstu poslovnog sustava. Iz navedenog je jasno da ni na jedno od gore postavljenih pitanja nije moguće dati jedinstveni odgovor koji je primjenjiv na svaku organizaciju.

### 4.1 Osnovni pojmovi razvoja IS-a

U prethodnim poglavljima prikazane su neke metode i tehnike za mjerenje učinaka IS-a/IT-ja na razvoj poslovnog sustava. Međutim, postavlja se pitanje: „Što je sve potrebno da bismo razvili informacijski sustav?“ Odgovor na to pitanje moguće je prikazati u nekoliko točaka:

- razumjeti poslovni sustav,
- istražiti ponašanje sustava u radu,
- poznavati procese na svim razinama,
- moći preustrojiti organizacijski sustav,
- poznavati informatičke tehnologije,
- izabrati optimalnu tehničku potporu,
- projektirati i programirati programe IS-a,
- uskladiti programe s poslovanjem,
- funkcionalno i ekonomski optimirati IS,
- uvesti novi IS i educirati korisnike,
- osigurati zaštitu IS-u tijekom rada.

U skladu s genetičkim pristupom IS-u, kao prvo i osnovno nameće se potreba poznavanja poslovnog sustava. Jedino na taj način bit će jasno što želimo postići

informatizacijom, tj. što je cilj. Nadalje, znamo da je svaka organizacija sustav za sebe te, iako postoji polazna sličnost među organizacijama, to ne znači nužno da će se koristiti ista tehnološka rješenja. Zbog toga je potrebno istražiti ponašanje sustava u radu i shodno tome pristupiti problemu informatizacije. Međutim, za proces razvoja IS-a potrebni su i financijski, ali i drugi resursi. Prilikom razvoja nije isključen reinženjering postojećih poslovnih procesa pa je jako bitno imati potporu vlasnika i rukovodstva organizacije.

Drugi set odnosi se na upoznatost s informatičkim tehnologijama. Naime, kako bi se uspješno unaprijedio sustav poslovanja na temelju IT-ja, potrebno je imati širok spektar znanja o raspoloživim tehnologijama koje se nalaze na tržištu. Iako na tržištu postoje različiti proizvodi za istu namjenu, potrebno je definirati optimalnu tehničku potporu koja je u skladu s poslovnom tehnologijom organizacije. Jedna veličina ne pristaje svima (engl. *one size does not fit all*), što znači da je potrebno ispitati mogućnost organizacije da koristi postojeće IT infrastrukture. U suprotnom, bit će potrebno kupljene softvere uskladiti s poslovnim zahtjevima, što može značiti i dodatno programiranje kako bi se ostvarila poveznica s postojećim IS-om.

Poseban naglasak prilikom razvoja treba staviti na funkcionalnu i ekonomsku optimalnost IS-a. Pritom se potrebno voditi načelom da „maksimum nije uvijek i optimum“. To znači da najskuplja i najbolja tehnologija za svaku organizaciju ne mora biti i optimalna. Iz tog je razloga potrebno napraviti i ekonomsku opravdanost svakog odabranog procesa u skladu s postavljenim ciljevima informatizacije.

Razvijanje IS-a ne završava izradom aplikacija. Razvijeni sustav potrebno je implementirati i osigurati zaštitu tijekom njegova rada. Implementacija IS-a ne podrazumijeva samo tehničku razinu već mora zahvatiti sve segmente organizacije. U tom kontekstu, potrebno je posebnu pozornost pridati edukaciji korisnika. Jedino dobro obučeni korisnici moći će u potpunosti iskoristiti prednosti novih tehnologija i tako doprinijeti ostvarenju željenih ciljeva informatizacije. Ovako implementirani (novi) IS potrebno je pratiti u daljnjem radu.

Sve prethodno ukazuje da razvoj IS-a organizacije nije moguće provesti u trenutku. Naime, nije moguće odrediti područje informatizacije, izdvojiti ga iz cjeline, informatizirati i potom ga vratiti u funkciju. Organizacija je „živi sustav“ koji radi i tijekom provođenja informatizacije. Iz tog je razloga potrebno upravljati razvojem informacijskih sustava.

#### 4.1.1 Klasične faze razvoja IS-a

Faze razvoja IS-a temelje se na fazama životnog ciklusa softverskog inženjerstva.

U svim su modelima razvoja IS-a više ili manje sadržane sljedeće tri aktivnosti koje će ovdje biti primijenjene (Garača, 2008):

- analiza,
- dizajn i
- implementacija.

Faza *analiza* treba rezultirati odgovorima na pitanja što je potrebno napraviti. Ova je faza više orijentirana poslovanju. To znači da se u ovoj fazi analizira funkcionalnost i ekonomska optimalnost, odnosno analiziraju se očekivani utjecaji novog IS-a. Cilj je definirati očekivanja od novog sustava na način da se definiraju prioriteta i set aktivnosti na temelju kojih će provesti informatizacija. Rezultat ove faze je specifikacija zahtjeva korisnika. Na taj način, problem informatizacije spušten je s poslovne na dizajnersku razinu. Sumarno, analiza uključuje modeliranje poslovnih procesa.

Nakon analize postoji sve što je potrebno da bi se krenulo u sljedeću fazu - dizajn novog IS-a. Dizajn uključuje dizajn arhitekture, korisničkog sučelja, strukture baze podataka, pa sve do rasporeda komponenti klijentsko-poslužničkih aplikacija (klijent/server aplikacija). Ova faza provodi se na konceptualnoj i detaljnoj razini. Prema specificiranim zahtjevima oblikuje se arhitektura poslovnih procesa. Konceptualno modeliranje IS-a kritično je za daljnji proces razvoja jer se prema razvijenom konceptualnom modelu pristupa detaljnom oblikovanju IS-a. Detaljno oblikovanje obuhvaća modele podataka, detaljni dizajn programa te u konačnici razvoj aplikacija koje će biti integrirane u sustav.

Integracija razvijenih aplikacija, tj. programa i procedura vrši se u fazi implementacije. Ova je faza najosjetljivija za realni sustav jer je staru tehnologiju potrebno zamijeniti novom za vrijeme rada sustava.

#### 4.1.2 Softversko inženjerstvo

Pristup informatizaciji organizacije kao pojedinačnom slučaju u svim njezinim fazama skup je i dugotrajan posao, zbog čega proizvođači softvera razvijaju jedinstvene proizvode koje je u nekim segmentima moguće primijeniti na veći broj organizacija. Pod pojmom softverskog proizvoda podrazumijeva se skup računalnih programa i pripadne dokumentacije koji se stvaraju zato da bi se prodali (Manger, 2016). Osnovna znanstvena i stručna disciplina koja se bavi svim aspektima proizvodnje softvera je softversko inženjerstvo. Softversko inženjerstvo predstavlja metodološki

okvir razvoja softvera (Garača, 2008). Osnovni su ciljevi: (1) finaliziranje uspješnog softverskog proizvoda te (2) uspješno upravljanje razvojem i održavanjem softvera.

Neke od relevantnih značajka softverskih proizvoda su (Sommerville, 2011):

- mogućnost održavanja, tj. mora se moći mijenjati u skladu s potrebama korisnika;
- pouzdanost i sigurnost, tj. mora se ponašati na predvidljiv način i ne smije izazivati fizičke ili ekonomske štete;
- efikasnost, tj. softver mora imati zadovoljavajuće performanse, odnosno upravljati strojnim resursima na štedljiv način;
- upotrebljivost, tj. softver treba raditi ono što korisnici od njega očekuju, sučelje mu treba biti zadovoljavajuće (engl. *user friendly*) i za njega mora postojati dokumentacija.

Softversko inženjerstvo bavi se modelima, metodama i alatima koji su potrebni da bi se što jeftinije proizvodili softverski produkti s prethodnim značajkama. Softversko inženjerstvo počelo se razvijati kao odgovor na tzv. "softversku krizu" krajem 60-ih godina 20. stoljeća (Sommerville, 2011). Razvojem računala stvorila se potreba za složenijim softverom, no ustanovilo se da dotadašnje neformalne tehnike individualnog programiranja premašuju planirane troškove i rokove. Rješenje navedenoga su razvoj i formaliziranje složenijih metoda razvoja softvera po uzoru na metodologije tradicionalnih tehničkih struka.

Softver se danas proizvodi daleko predvidljivije i efikasnije nego prije. Pri razvoju softverskih proizvoda vrlo je izražena težnja da se u daljnjem radu ponovno iskoriste rezultati prethodnog rada. Objektiva orijentacija (OO) izvorno podupire ponovno korištenje softverskih elemenata. Ponovna iskoristivost softverskih elemenata uvelike utječe na učinkovitost razvoja softverskog proizvoda. Središnja ideja OO-a jako se dobro uklapa u prethodni napredak u programskom inženjerstvu jer OO podrazumijeva da se korištenjem prethodno razvijenih i dobro testiranih komponenata izbjegava manje pouzdan pristup – odnosno izgradnja softvera od nule (Fichman & Kemerer, 1993). Međutim, prilikom odabira softverskog proizvoda potrebno je imati na umu da kupovinom cjelokupnog softverskog proizvoda možda kupujemo i novu tehnologiju koja ne mora uvijek biti najprikladnija za naš poslovni sustav.

### 4.1.3 Modaliteti izgradnje IS-a

Prva ključna odluka u informatizaciji poslovnog sustava izbor je modaliteta razvoja:

- interni razvoj (engl. *insourcing*),
- samostalni razvoj (engl. *selfsourcing*),
- vanjska usluga razvoja (engl. *outsourcing*).

Interni razvoj predstavlja opciju u kojoj se razvoj novog IS-a povjerava vlastitim snagama. To podrazumijeva jake informatičke kadrovske resurse. Iako je ovakav pristup nekada bio primaran, porastom primjene i razvojem informatičkog tržišta trend se promijenio. Međutim, postavlja se pitanje: ako se posao povjeri vanjskoj tvrtki, hoće li ona, s obzirom na specijaliziranost područja, biti u stanju preustrojiti i poslovni sustav ili će samo željeti prodati svoj softverski proizvod? Današnja podjela poslova u procesu informatizacije omogućila je kombinirani pristup. To znači da npr., ovisno o raspoloživim resursima, određene faze razvoja organizacija može raditi samostalno, dok je druge moguće povjeriti softverskim tvrtkama.

Interno razvijene aplikacije ograničene su na zadovoljavanje potreba svojih autora. Osnovna je razlika modaliteta internog razvoja i samostalnog razvoja u potrebi uključivanja IT stručnjaka. Naime, potrebna informatička rješenja odnose se uglavnom na samostalna unapređenja u skladu s vizijom uloge IS-a sa stajališta pojedinca. U slučaju opsežnijih poslova također je potrebno pridržavati se metoda razvoja IS-a.

Vanjska usluga razvoja danas je prevladavajući pristup informatizaciji organizacijskih sustava. Globalna ekonomija istaknula je nekoliko dominantnih proizvođača softvera. Međutim, prilikom razvoja IS-a potrebno je izuzetno poznavanje poslovne tehnologije onoga koji odabire softver. Postoje različite metode izbora softvera, od kojih je jedna od poznatijih metoda AHP (engl. *analytic hierarchy process*). Naime, čest je slučaj da se prilikom odluke o informatizaciji organizacije kupovinom cjelokupnog IS-a kupuje i tehnologija. S druge strane, kupovinom tehnologije dominantnog proizvođača prilagođena poslovna tehnologija omogućuje usklađivanje i povezivanje s globalnim igračima iz poslovnog područja organizacije. Jednako tako, vanjska usluga razvoja ne definira opredjeljenje na globalne igrače nego i na lokalne informatičke tvrtke koje mogu proizvesti jednako kvalitetne softverske proizvode. I u ovom slučaju odgovor je potrebno tražiti u poslovnim ciljevima organizacije.

#### 4.1.4 Strukturno ili objektno modeliranje IS-a

Modeliranje je osnovna aktivnost koja prethodi razvoju dobre programske podrške jer modeliranje pomaže pri razumijevanju problema i pri jednostavnijem odabiru rješenja (Muller, 2013). Modelira se kako bi se bolje shvatio sustav koji se izgrađuje i često se koristi radi pojednostavljenja i mogućnosti ponovne iskoristivosti dijelova tog istog sustava (Martin & Odell, 1997). Modeliranjem se vizualizira i kontrolira arhitektura sustava. Modeliranje je dokazana široko prihvaćena inženjerska tehnika. Primjerice, arhitekti koriste modeliranje kako bi olakšali projektiranje neke kuće ili

zgrade, pri čemu koriste razne matematičke modele kako bi analizirali utjecaje vjetra i potresa na stambene objekte. Naravno, modeliranje nije vezano samo za građevinsku industriju. Bilo bi gotovo nemoguće proizvesti novi avion ili automobil bez prethodne izrade modela – od računalnih i fizičkih modela koji se testiraju u zračnim tunelima pa sve do pravih prototipova. Model pruža na uvid nacрте za izgradnju određenog sustava. Oni mogu sadržati vrlo detaljne nacрте, kao i one generalne koji daju samo prikaz većih cjelina. Dobar model uključuje one elemente koji imaju širok utjecaj, a zanemaruje one manje elemente koji nisu relevantni za promatranu razinu apstrakcije. Svaki sustav može se opisati s više aspekata koristeći više modela pa je samim time svaki model semantički zatvorena apstrakcija sustava. Modeliranjem se postižu četiri cilja (Molnár, Benczúr & Béleczi, 2016; Schmuller, 2004):

- modeli pomažu pri vizualizaciji sustava koji se promatra ili koji se želi izgraditi;
- modeli omogućuju specifikaciju strukture ili ponašanja sustava;
- modeli pružaju predloške koji služe kao vodiči pri izgradnji sustava;
- modeli dokumentiraju odluke koje su donesene.

Strukturna metoda modeliranja ima isti princip kao programiranje, odnosno rezultat je algoritam – niz funkcija. Iz toga slijedi da su upravo procedure ili funkcije glavni građevni elementi programa. Ovaj način razvoja nije loš, ali pokazuje svoje velike nedostatke pri promjenama zahtjeva na sustav. Iz toga proizlazi da su sustavi razvijeni na temelju algoritma vrlo teški za održavanje.

Moderan pogled na razvoj sustava za programsku podršku koristi objektno-orijentirani pristup.<sup>1</sup> Ovdje je glavni građevni blok sustava objekt ili skup objekata koji apstrahiranjem, pojednostavljeno rečeno, čine klasu, tj. skup objekata s istim svojstvima. To je odabrani entitet koji je izveden iz rječnika domene problema ili rješenja, dok je klasa opis kolekcije objekata s istim zajedničkim svojstvima. To znači da je u objektno orijentiranom pristupu naglasak na modeliranju stvarnosti (realnih objekata u domeni razmatranog problema). Svaki objekt ima:

- identitet (može se imenovati ili nekako drukčije razlikovati od ostalih objekata),
- stanje (obično su tu pridruženi nekakvi podaci) i
- ponašanje (mogu se vršiti nekakve operacije nad objektom ili on može vršiti operacije nad drugim objektima).

Objektno orijentirani pristup zauzima središnje mjesto među pristupima razvoja programske podrške jednostavno zato što je dokazao svoju vrijednost u razvoju sustava u raznim problemskim domenama i obuhvaćajući sve razine kompleksnosti.

---

<sup>1</sup> Više o pojmu i primjerima klase i objekta te objektnom pristupu bit će riječi u sljedećim poglavljima.

Većina modernih jezika, operativnih sustava i alata su donekle objektno orijentirani, dajući time veći doprinos objektno orijentiranom pogledu na svijet.

Za razliku od objektnog programiranja, proceduralno programiranje podrazumijeva program kao niz naredbi koje računalo treba odraditi. Na toj razini je definirano podacima (sadržajem memorije) i naredbama strojnog jezika, tj. operacije i podatci su razdvojeni. Pojednostavljeno rečeno, izvođenje programa, počevši s prvom naredbom, slijedi predvidljivi put zapisanih naredbi, koji završava zadnjom naredbom programa. Objektni pristup omogućuje izradu programâ koji se pokreću događajima, npr. klikom miša i sl. Promotrimo li realni sustav i zaboravimo na programiranje i informatičke tehnologije, vidimo da se i njegova promjena stanja zasniva na događajima. Primjerice, objekt *aparata za kavu* bit će u stanju serviranja kave jedino ako smo odabrali vrstu kave. U prethodnom primjeru događaj je bio pokrenut akcijom korisnika, iako on može biti pokrenut i porukama iz sustava ili drugih programa. Iz tog razloga kažemo da objekti međusobno komuniciraju. Ovakav pristup bliži je čovjekovu shvaćanju ponašanja realnog sustava i lakše ga je koristiti prilikom modeliranja IS-a jer je lakše opisati načine izvođenja procesa. Pojednostavljeno rečeno, objektno orijentirani pristup, za razliku od strukturnog pristupa modeliranja IS-a, polazi od realnih objekata koji međusobno komuniciraju kako bi postigli cilj zadatka postavljenog pred cjelokupni sustav. Na taj način pojednostavljuje se konceptualno modeliranje sustava kao i implementacija IS-a u organizaciju. Prednosti objektnog pristupa su:

- veća pouzdanost i fleksibilnost u razvoju;
- lakši rad s nestandardnim tipovima podataka;
- manja ovisnost o sklopovskoj opremi;
- lakša izvedba aplikacije klijentskog poslužitelja;
- usmjerenost na ponašanje, a ne na procedure obrade.

Više o pojmu objekta i načinima korištenja objektnog pristupa kod modeliranja obradit će se u nastavku.

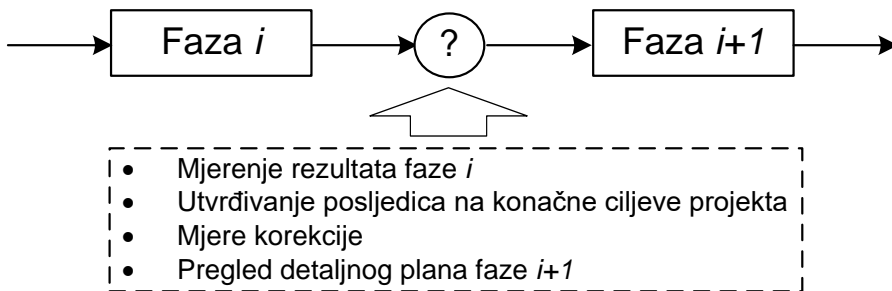
#### 4.1.5 Upravljanje informatičkim projektima

*Project Management Body of Knowledge* (PMBOK) definira projekt kao „... privremeni pokušaj pothvata za stvaranje jedinstvenog proizvoda, usluge ili rezultata“ (PMI, 2017). Pretražimo li na nekoj od tražilica pojam projekta, također ćemo dobiti sličnu definiciju. Projekt je ciljno usmjerena, jednokratna, relativno nova i kompleksna namjera, produkt ili cjelovitost međusobno povezanih aktivnosti čije je trajanje vremenski ograničeno. Privremenost i jedinstvenost projekta strogo razlikuju projekt od redovnih radnih operacija i/ili procesa. Ciljevi projekta ostvaruju se aktivnostima koje mogu biti podijeljene u zadatke. Provođenje aktivnosti karakterizira korištenje resursa i određeni stupanj rizika.

U skladu s prethodnim, informatički projekti su projekti realizacije i implementacije poslovnih rješenja zasnovanih na IT-ju, koji zahtijevaju značajne resurse i imaju ograničeno trajanje. Upravljanje informatičkim projektima obuhvaća sljedeće faze:

- planiranje rada na projektu,
- provođenje rada na projektu,
- kontrolu razine ostvarenosti ciljeva projekta,
- vođenje projektnog tima.

Informatički projekti ostvaruju se u nekoliko faza, a pod fazom se podrazumijeva skup aktivnosti koje se poduzimaju kako bi se ostvarili planirani ciljevi projekta. Kako bi se moglo upravljati informatičkim projektom, svaki od postavljenih ciljeva mora biti kvalitativno i kvantitativno opisan. Na taj je način po završetku planirane faze moguće mjeriti ostvarene rezultate. Na osnovi utvrđenih odstupanja moguće je planirati potrebne korektivne mjere kako bi se ispravili neostvareni učinci planirane faze. Ovisno o uočenom odstupanju, planirane mjere moguće je uklopiti u detaljan plan sljedeće faze ili jednostavno odgoditi sljedeću fazu dok se ne postigne željeni stupanj zadovoljstva ostvarenošću promatrane faze. Ovakav proces upravljanja prikazan je na slici 4.1.



*Slika 4.1 Proces upravljanja informatičkim projektom*

Razlozi odstupanja planiranih i ostvarenih učinaka informatičkih projekata mogu biti:

- Nepotpuna specifikacija korisničkih zahtjeva. Može se dogoditi da specifikacija zahtjeva nije točan ili potpun odraz onoga što korisnik želi pa kada dođe do konačnog rješenja, to se pokaže kao loš rezultat.
- Odabir nekompetentnog voditelja. Jako je bitno da je voditelj projekta osoba s iskustvom i dobrim komunikacijskim vještinama kako bi se održalo sve na okupu. Ako voditelj nije u stanju popratiti i intervenirati u trenutku kada je to potrebno, stvari mogu poći po zlu.



- Nedovoljna ili loše izvedena priprema projekta. Jako je bitno da je projekt dobro pripremljen u svim aspektima. Treba uzeti u obzir sve rizike koji se mogu javiti kako bi se u što kraćem roku odgovorilo na moguće probleme i projekt uspješno provelo do kraja.
- Loš ili nepotpun plan projekta. Sve aktivnosti, kao i potrebni resursi, u projektu moraju biti detaljno razrađeni, npr. vrijeme, novac, ljudi i sl. Dobar plan je osnovan jer umanjuje rizik neuspješnosti projekta.
- Otvoren regulacijski krug upravljanja projektom. Potrebno je kontinuirano mjeriti i uspoređivati sa željenim rezultatima kako bi se mogle provesti odluke o potrebnim izmjenama.
- Neodgovarajući sastav tima. Nije lako odabrati članove tima jer nije rijetkost da dođe do međusobnih konflikata. Razlog tome može biti da neke osobe bolje funkcioniraju same ili da se radi o jako različitim karakteristikama.
- Nedostatak suradnje naručitelja tijekom realizacije projekta. Ako nedostaje suradnje, može se dogoditi da dođe do velikog odstupanja od željenog stanja. Analogno tome, za potrebne izmjene treba više resursa.
- Nedovoljni resursi za realizaciju projekta. Ovo je rezultat loše pripreme projekta i loše razrađenog budžeta. S nedovoljnim resursima rezultat projekta značajno je ugrožen.
- Neizvršavanje aktivnosti predviđenih planom projekta. Ako se aktivnosti ne izvršavaju planom kojim su predviđene, dolazi do kašnjenja ostalih aktivnosti koje o njima ovise. Produženje projekta za sobom povlači i dodatnu potrebu za resursima (novčanim, vremenskim, ljudskim i sl.), što sigurno nije u interesu naručitelja.
- Nedostatak nadzora nad kritičkim faktorima uspjeha (KFU). Potrebno je nadzirati određene aktivnosti projekta čiji će zadovoljavajući rezultati osigurati uspjeh projekta. Ako kritične aktivnosti nisu nadzirane, neuspjeh projekta vrlo je vjerojatan.
- Povećani utrošak resursa na projektu. Svako odvijanje aktivnosti koje nije po planu može uzrokovati povećanu potrebu resursa. Svako povećanje resursa mora odobriti naručitelj, što sigurno nije nešto što će biti lako prihvaćeno. „Probijanje“ budžeta definitivno nije dobar znak statusa projekta.

## 4.2 Životni ciklus informacijskih sustava

Baš kao i većina objekata ljudske tvorevine, tako i IS ima svoje faze nastajanja, razvoja, sazrijevanja i nestajanja. Zbog toga se taj proces naziva životni ciklus informacijskih sustava. Životni ciklus detaljno je podijeljen u faze (Garača, 2008):

1. planiranje informacijskog sustava,

2. prikupljanje zahtjeva,
3. specifikacija zahtjeva,
4. dizajn arhitekture,
5. detaljni dizajn,
6. integracija,
7. implementacija,
8. održavanje.

U daljnjem tekstu ukratko je opisana svaka od ovih faza. Sadržajem ovog kolegija obuhvaćene su prve tri faze koje se posebno analiziraju u sljedećim poglavljima.

### 4.2.1 Planiranje informacijskih sustava

Često se među analitičarima postavlja pitanje je li planiranje razvoja potrebno uzevši u obzir današnju brzinu promjena. Mnogi analitičari tvrde da se promjene događaju nepredvidivo i veoma brzo te da zbog toga nije potrebno planirati. No upravo da bi bilo moguće brzo prilagođavanje i reagiranje na promjene, potrebno je imati nekakvu podlogu – plan. Sigurno da nije moguće jednom donijeti odluku i potom reći da to vrijedi zauvijek. Planiranje razvoja je "pomična meta" koja se s vremenom mijenja i potrebno ju je kontinuirano preispitivati. Tipične su značajke projekata, pa tako i IT projekta, vrijeme i novac. Iz tog razloga je potrebno ne samo planiranje nego i vođenje projekta, o čemu će u sljedećim poglavljima biti više riječi. Pod planiranjem razvoja IS-a ovdje se podrazumijeva utvrđivanje ciljeva razvoja te strategijâ i aktivnosti za njihovo ostvarenje, a sve u svrhu povećanja učinkovitosti i efikasnosti poslovanja promatrane organizacije. To znači da se planiranje IS-a radi na temelju poslovnih ciljeva organizacije. Potom se utvrđuju područja na kojima IS može doprinijeti njihovu ostvarenju. U skladu s utvrđenim područjima planiraju se aktivnosti informatizacije i postavljaju mjerni instrumenti na osnovi kojih će biti moguće uočiti potrebnu dinamiku provedbe utvrđenih planova. Detaljni prikaz planiranja IS-a/IT-ja napravljen je u prethodnim poglavljima ove knjige. Na osnovi tako utvrđenih planova potrebno je utvrditi zahtjeve korisnika.

### 4.2.2 Prikupljanje zahtjeva

Osnovni zadatak u fazi prikupljanja zahtjeva je prikupiti, analizirati i utvrditi zahtjeve korisnika. U svrhu prikupljanja zahtjeva moguće je primjenjivati različite tehnike, kao što su anketiranje, intervjuiranje i sl. Postoje i druge, naprednije tehnike, o kojima će biti riječi u sljedećim poglavljima. Prikupljanje zahtjeva, osim (1) pitanja samih korisnika, može biti i (2) deriviranje zahtjeva iz postojećih informacijskih sustava, (3) sinteza karakteristika korištenih sustava te (4) eksperimentiranje s razvojem sustava (Davis G. B., 1982). Prilikom prikupljanja zahtjeva potrebno je ustanoviti korisnikovo očekivanje o funkcionalnostima budućeg IS-a. Nakon prvog prikupljanja zahtjeva provodi se analiza, u koju se preporučuje uključiti eksperta za promatrano područje.

Tako je, primjerice, kod sustava za potporu odlučivanju potrebno konzultirati eksperte iz područja menadžerskog odlučivanja, kako bi sugerirali koju metodu je najprikladnije koristiti za određeni slučaj te koje softverske alate integrirati kako bi se na jednostavan način povećala učinkovitost u donošenju odluka. Analiza zahtjeva mora rezultirati usklađenošću korisnika i programera – ako IS nije prilagođen korisnicima i njihovim zahtjevima, tada mu se gubi smisao. U ovom koraku se na taj način izbjegavaju kontradiktorna tumačenja zahtjeva, čime se omogućuje definiranje budžeta i trajanja IS/IT projekta. Rezultat ove faze su dokumentirani zahtjevi korisnika za razvoj IS-a.

#### 4.2.3 Specifikacija zahtjeva

Faza specifikacije zahtjeva započinje formalnim modeliranjem korištenjem različitih tehnika. Postoji više pristupa prilikom modeliranja korisničkih zahtjeva (Soaresa, Vranckenb, & Verbraeckb, 2010):

- grafičko modeliranje,
- tekstualno opisivanje te
- kombinacija tekstualnog i grafičkog opisivanja.

Korištenje jednog od takvih alata (dijagramska tehnika alata CASE) u svrhu analize i dokumentiranja modela označuje početak faze specifikacije zahtjeva. Na taj način dokumentirani zahtjevi mogu sadržavati grafičke modele i alatom standardizirane izvještaje, koji finaliziraju dokumentirane zahtjeve u specifikacijske dokumente.

Kod objektno orijentirane analize postoje dvije važne specifikacijske tehnike koje su nam posebno zanimljive – klasni dijagram (engl. *class diagram*) i dijagram slučajeva korištenja (engl. *use case diagram*). Ove dijagramske tehnike namijenjene su podatkovnoj i funkcijskoj specifikaciji. Tipični specifikacijski dokumenti sadrže i ostale zahtjeve, kao što su svojstva, upotrebljivost i sigurnost (nefunkcionalni zahtjevi). Korištenjem specifikacijskih modela omogućuju se različiti pogledi na sustav. Svi specifikacijski modeli moraju se u konačnici preklapati, tj. među njima mora postojati konzistentnost i potpunost. Naime, bitno je da specifikacijski modeli budu razumljivi korisniku i neopterećeni konkretnom IT terminologijom. Jako je važno da ljudi mogu čitati te modele, a uzevši u obzir uključenost velikoga broja dionika s različitim pozadinskim znanjem, od iznimne je važnosti da konačan model bude što manje ovisan o metodologiji (Soaresa, Vranckenb, & Verbraeckb, 2010). U idealnom slučaju specifikacijski modeli moraju biti neovisni i od IT platforme na kojoj će se IS razvijati.

Dakle, specifikacija zahtjeva je temelj za razvoj IS-a, tako da je jako bitno da je ova faza detaljno i precizno odrađena. Ponekad bi, ako je izvedivo, bilo dobro postaviti i

prioritete među pojedinim zahtjevima. U konačnici trebamo imati dobro strukturirane, konzistentne i kontrolirane specifikacijske zahtjeve.

### 4.2.4 Dizajn arhitekture

Izraditi arhitekturu znači odrediti sastavne elemente (module) IS-a i jednoznačno odrediti veze (ili sučelja) između modulâ. U tu svrhu se koriste odgovarajuće metode, npr. analiza osjetljivosti. Uočavanje arhitekture informacijskog sustava može biti autosugestivno jer određene grupe poslovnih procesa i klasâ podataka čine uobičajena funkcijska područja. Međutim, takav pristup ne osigurava optimalnu arhitekturu (Dunđer, Juričić & Pogarčić, 2012). Iz tog razloga su afinitetna analiza i izrada matrice poslovne tehnologije vrlo važni koraci u projektiranju informacijskog sustava. Naredne faze projektiranja su izravne logičke posljedice, što je prvi prepoznao IBM još 1970-ih godina. Za modeliranje različitih pogleda na sustav u svrhu dizajniranja arhitekture u petom poglavlju će se posebno obraditi unificirani jezik za modeliranje (engl. *Unified Modelling Language – UML*) i izrada specifikacijskih dokumenata. Specifikacijski dokumenti predstavljaju "ugovor" između programera i korisnika za razvoj IS-a. Oni sadrže sve zahtjeve koje novi IS treba ispuniti. Specifikacijski zahtjevi se dalje šalju projektantu IS-a da razvije detaljni model arhitekture IS-a u terminima IT platforme na kojoj će sustav biti razvijen.

### 4.2.5 Detaljni dizajn

U fazi detaljnog dizajna sustava opisuju se pojedine aplikacije modula i oblikuju baze podataka. Za svaki od prethodno specificiranih modula razvijaju se detaljni algoritam i struktura podataka. U tipičnom razvoju IS-a moduli se nalaze ili na strani klijenta ili na strani servera. Prilikom detaljnog dizajna modula obavljaju se i poslovi oblikovanja korisničkog sučelja, baze podataka i sl. Ovdje je riječ prepuštena programerima, odnosno razvojnim timovima. Ova faza nije predmet izučavanja u ovoj knjizi.

### 4.2.6 Integracija

Faza integracije karakterizira integraciju razvijenih komponenti na razvojnom modelu IS-a. Iako to naizgled predstavlja jednostavan posao, kod velikih sustava često zna uzeti više vremena nego ijedna druga faza. Osnovna poteškoća se javlja prilikom usklađivanja isprepletene mreže međuovisnosti komponenti cijelog sustava. Integracijsku strukturu je moguće prikazati komponentnim dijagramom, ali nema slijednog izvršavanja pojedinih komponenti već različite instance mogu (u isto vrijeme) pozivati različite komponente. To nas onda vodi do suvremene, servisno orijentirane arhitekture, o kojoj će više riječi biti u nastavku knjige. Objektно orijentirani sustavi moraju biti dizajnirani za integraciju. Svaka komponenta mora biti neovisna, a ovisnost komponenti mora biti minimizirana prilikom faze analize i dizajna. Svaka se komponenta testira u „razvojnom okruženju“. Nakon provjere pojedinačnog rada, pojedine komponente se integriraju u cjelinu (upravo se to

prikazuje komponentnim dijagramom) i testiraju kao cjelina (cjeloviti IS), ali sve još uvijek u razvojnom okruženju. Ako je sve u redu, piše se konačna korisnička dokumentacija.

#### 4.2.7 Implementacija

Nakon što je testirano funkcioniranje IS-a kao cjeline, sve se prenosi u „produksijsko okruženje“, gdje se podižu stvarne baze podataka i pune kataloškim podacima, dodjeljuju prava pristupa, uspostavljaju komunikacije, provodi edukacija korisnika, organizira tranzicija sa starog sustava na novi IS (probni ili paralelni rad) itd. Ova se točka zove „uvođenje“ (ili „implementacija“). Implementacija IS-a uključuje instalaciju naručenog softvera koji treba podržati IS. Neke od ključnih aktivnosti su testiranje, obuka korisnika i instalacija hardverske podrške - ovisno o specifičnosti projekta.

Kod velikih sustava su odgovornosti implementacijskog tima podijeljene u dva segmenta. Jedan dio članova tima programira serverske aplikacije, dok drugi dio tima snosi odgovornost za stranu klijenta IS-a. Odgovornost za konzistenciju bazâ podataka snosi dio članova tima koji se bavi implementacijom na strani servera. Programeri, koji su zaduženi za klijentsku stranu, instaliraju operativne sustave i aplikacije na strani klijenata te ih po potrebi povezuju sa serverskom aplikacijom. Ponekad ova faza može zahtijevati i određene promjene u korisničkom sučelju, kako bi se postigla produktivnost korisnika. Slično je i sa serverskim bazama podataka. Nepredviđeni problemi i teškoće u programiranju ili pohranjivanju podataka mogu uzrokovati doradu predloženog dizajna. Bitno je znati da fazu implementacije nije jednostavno i lako odraditi jer je to onaj trenutak u kojem se sustav pušta u život.

#### 4.2.8 Održavanje

Održavanje informacijskog sustava slijedi nakon predaje cijelog informacijskog sustava na korištenje korisniku. Održavanje karakterizira nekoliko osnovnih skupina radnji. Jedna od njih je rutinsko održavanje kako bi korisnik mogao nastaviti koristiti IS.

Druga skupina radnji odnosi se na adaptaciju, koja uključuje nadzor i kontrolu rada operacija podešavanjem funkcionalnosti. Na taj se način finim podešavanjem u svrhu prilagođavanja okolnim zahtjevima postižu bolje performanse IS-a, a samim time i podižu zadovoljstvo i učinkovitost krajnjih korisnika.

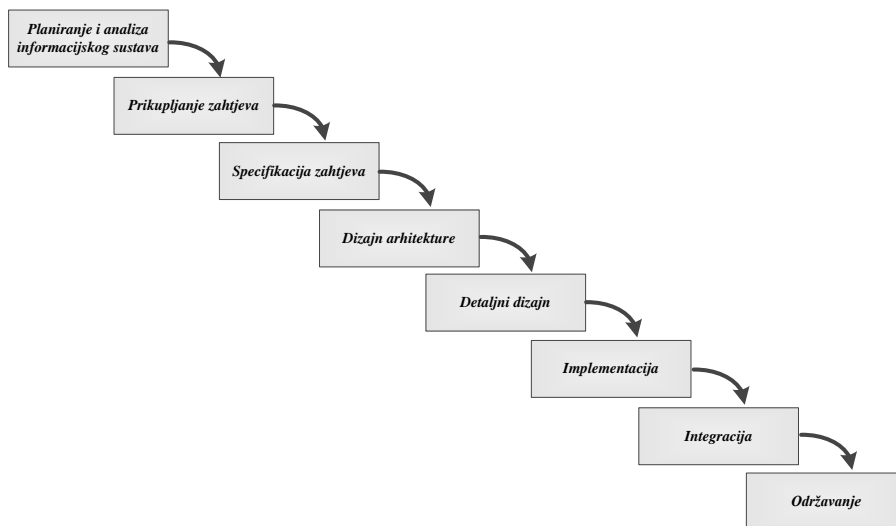
Održavanje unapređenjem postojećeg sustava podrazumijeva treću skupinu radnji. Ova skupina uključuje redizajn i modificiranje IS-a u svrhu njegova prilagođavanja novim i/ili izmijenjenim zahtjevima.

### 4.3 Klasični pristupi razvoju IS-a

Kao što je već spomenuto, osnovna ideja softverskog inženjerstva je da se softver sastoji od građevnih elemenata (kao što je to slučaj kod građevinskih objekata). Zbog toga je trebalo utvrditi pristupe i metodologije razvoja IS-a. Jedan od klasičnih pristupa razvoju je vodopadni pristup (engl. *waterfall methodology*). Osnovna razlika pristupa i metodologije razvoja IS-a je u tome što je metodologija definirani skup metoda koje se koriste prilikom razvoja, za razliku od pristupa, koji se, pojednostavljeno rečeno, može promatrati kao apstraktni model u fazi razvoja.

#### 4.3.1 Vodopadni pristup

Osnovna značajka vodopadnog pristupa je linearnost, prema kojoj nema povratka na prethodne faze životnog ciklusa IS-a. To znači da svaka faza mora završiti prije nego sljedeća započne. Na taj način postignuta je jednostavnost planiranja. Ovaj pristup omogućuje da svaka faza koristi svoje metode. Angažirani ekspertni tim za određenu fazu rješava svoj problem te ga dokumentira i predaje drugom timu u sljedećoj fazi. Vodopadni pristup razvoja prikazan je na slici 4.2.



Slika 4.2 Vodopadni pristup razvoju IS-a

Takav način razvoja je lijepo zamišljen, ali teško praktično primjenjiv. Problem se javlja zbog izostavljanja detaljnog sagledavanja problema. To znači da se prilikom odvijanja neke od faza ne ostvaruju pogledi na sustave koji su predviđeni drugim fazama. Tako se, primjerice, prilikom analize zahtjeva ne promatraju detalji izvedivosti samog sustava. To je problem koji se rješava u kasnijim fazama. Zbog toga, ako takvo rješenje dođe u kasnijim fazama, prema vodopadnom pristupu nema povratka na prethodne faze. Problemi sličnog tipa pojavljuju se i kod krajnjih korisnika kojima se informatizacijom želi povećati učinkovitost njihova rada. Krajnje korisnike se

konzultira u fazi prikupljanja zahtjeva, prilikom opisivanja procesa, kao vlasnike procesa ili sudionike u izvršenju njegovih poslova. I tu prestaje komunikacija s krajnjim korisnicima. Međutim, što ako je cjelokupni projekt trajao više godina? Što ako su se korisnički zahtjevi u međuvremenu promijenili? Zbog toga vodopadni pristup dozvoljava fokusiranje samo na određene probleme.

Vodopadni pristup je prikladno koristiti (Radovan, 1991):

- za ponavljanje unapređenja konkretnih vrsta poslova koje uključuju male promjene (npr. unapređenje e-trgovine mijenjanjem postavki proizvoda),
- kao radni okvir za učenje različitih tehnika i metoda u razvoju softvera,
- kao jedan krug kod spiralnog pristupa,
- kao radni okvir za potporu iterativnom modelu,
- za brzi razvoj malih projekata koje realizira mali broj ljudi.

#### 4.3.2 Spiralni pristup

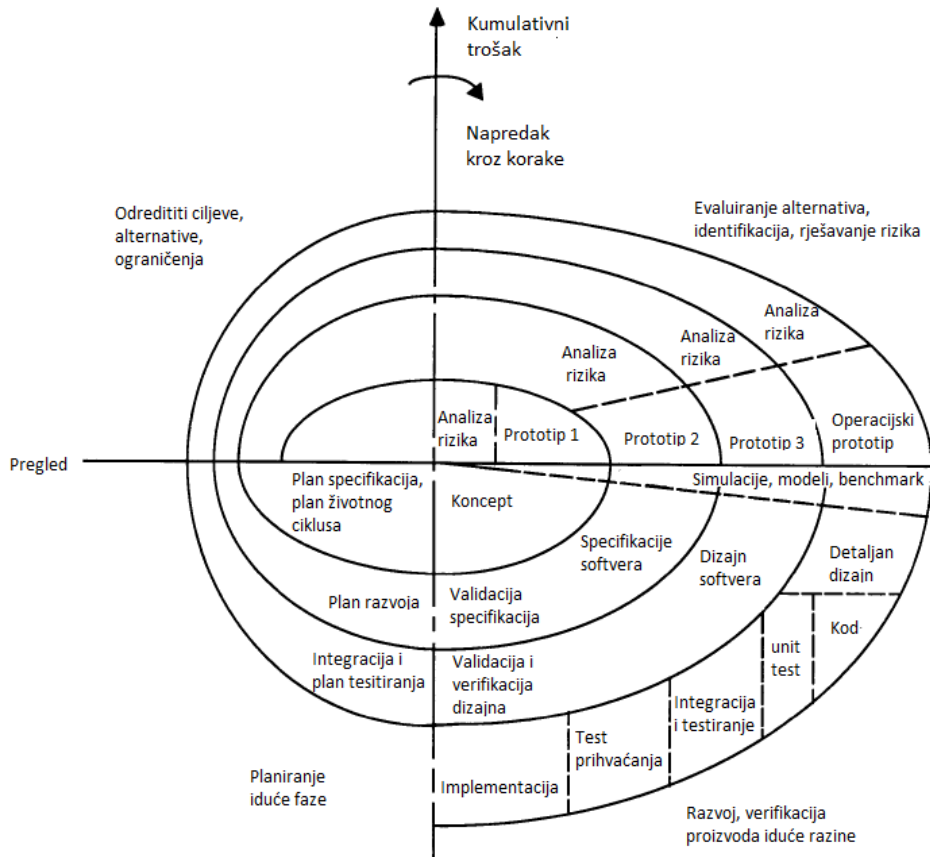
Kao što se može zaključiti iz prethodnog, razvoj IS-a nije moguće ostvariti u jednom prolasku faza životnog ciklusa. Suština spiralnog pristupa je u tome što se faze životnog ciklusa primjenjuju na jedan poslovni modul, pa se zatim prelazi (u istim fazama) na drugi modul itd. Ovi moduli su određeni arhitekturom IS-a. Jedna od takvih metodologija je spiralna metodologija (Boehm, Defense & Group, 1987).

Spiralni model ima dvije glavne značajke: (1) ciklički/iterativni pristup, koji služi postupnom razvoju nekog sustava, (2) umanjujući pritom stupanj rizika (Barry & Wilfred, 2000). Dakle, korištenjem modela prolaze se sve faze razvoja, ali više puta, odnosno postoji više iteracija koje se odvijaju u prosjeku nakon šest mjeseci do dvije godine (SBS, 2018). Za razliku od vodopadnog modela razvoja, u kojem se sve izrađuje iz jednog puta, spiralnom metodologijom se IS razvija postupno, u nekoliko slojeva. Prvo se razvije osnovni radni okvir IS-a, potom se on popunjava sloj po sloj, sve dok se ne zadovolje željena očekivanja naručitelja. Kao i obično, počinje se planiranjem informacijskog sustava. Grafički prikaz spiralnog modela prikazan je na slici 4.3.

Slijedeće faze opisane su životnim ciklusom IS-a. Za razliku od vodopadnog, spiralni pristup dopušta izvođenje daljnjih faza iako prethodne nisu u potpunosti završene. To znači da je unatoč tome što, primjerice, nije u potpunosti dovršena faza specifikacija moguće pristupiti okvirnom dizajnu, implementaciji i integraciji, kako bi se naručitelju pokazali gotovi rezultati. Ovisno o zadovoljstvu korisnika rade se preinake prethodno utvrđenog plana. Jednom kad je IS kompletiran, nakon tri do četiri iteracije, može se pristupiti kompletnoj integraciji sustava.

Neke od prednosti spiralnog pristupa nad vodopadnim su:

- u proces razvoja su uključeni naručitelji;
- može se vidjeti kontinuitet razvoja IS-a;
- fleksibilnost, što znači da je moguće podešavanje širine zahvata i njegova vremenskog trajanja.



Slika 4.3 Spiralni model razvoja IS-a (Boehm, 1988)

Međutim, i spiralni pristup ima svoje nedostatke. U samoj biti on je višekratno ponavljanje vodopadnog pristupa. Problemi vodopadnog pristupa postaju manji, ali i dalje nisu u potpunosti otklonjeni. Zbog toga ju je potrebno kombinirati s drugim pristupima razvoja.

### 4.3.3 Iterativni pristup

Za proširenje koncepta spiralnog pristupa razvoju IS-a potrebna je nadogradnja koja će dopustiti kretanje fazama životnog ciklusa razvoja IS-a u oba smjera. Upravo to nam omogućuje iterativni pristup. Nakon "djelomičnog" završetka jedne faze pristupa se provođenju druge faze. U slučaju uočenih nedostataka inkrementalni model (male i česte promjene) omogućuje vraćanje na prethodnu fazu u svrhu njihovog



otklanjanja (Nerur, Mahapatra, & Mangalaraj, 2005). Na prvi pogled izgleda da se na ovakav način može izazvati potpuni kaos u razvoju sustava. Osnovne značajke iterativnog pristupa koje osiguravaju da do kaosa ne dođe jesu:

- klasične faze podsjećaju što je potrebno raditi u svakom koraku i u kojem se smjeru potrebno kretati;
- dokumenti koji sadrže dijagrame, opise, programski kôd itd. (rezultat obavljenih poslova u pojedinoj fazi) jasno navode na sljedeće korake;
- softverski alati za potporu razvoju i njihova notacija pomažu da se osigura konzistentnost razvojne dokumentacije.

Iterativnim pristupom omogućuje se kontinuirani angažman krajnjeg korisnika i osiguravaju mu se povratne informacije. Izvještavanje se vrši poslije svake iteracije. Na taj način otklanjaju se rizične stavke u svakoj fazi životnog ciklusa, što smanjuje rizik cijelog projekta. Tako razvojni timovi, neopterećeni rizikom željenih učinaka, ostaju usmjereni na razvoj softvera. Na taj način iterativni pristup olakšava taktičke promjene u zahtjevima ili rasporedu izvođenja aktivnosti.

#### 4.4 Racionalni unificirani proces

Racionalni unificirani proces (engl. *rational unified process* - RUP) je metodologija razvoja softvera (Kroll & Kruchten, 2003). On osigurava disciplinarni pristup dodjeli zadataka i odgovornosti u svrhu organizacijskog razvoja. RUP je originalni proizvod tvrtke *Rational Software*, koja je danas sastavnica tvrtke IBM (IBM, 2014). Postoji nekoliko značajki koje ga izdvajaju od prethodnih metodologija:

- produktivnost razvojnog tima,
- orijentiranost modelima,
- vodič za učinkovito korištenje UML-a,
- podržan alatima,
- prilagodljivost razvojnog procesa,
- razvijen na primjerima dobrih praksi.

RUP se razvijao na primjeru razvoja IS-a u uspješnim tvrtkama, tzv. primjerima dobrih praksi. Na temelju takvih dobrih praksi daje raspored potrebnih aktivnosti razvoja IS-a. Za svakog člana razvojnog tima definirane su upute, obrasci i tutorski alati. Polazeći sa stajališta kako razvoj današnjih sofisticiranih IS-a nije moguće provesti u jednom ciklusu, RUP koristi iterativni pristup. Propisani sustav dokumentiranja po fazama minimizira rizik provođenja projekta. RUP definira četiri faze životnog ciklusa IS-a, koje u temeljnim procesima odgovaraju klasičnom životnom ciklusu analiziranom u prethodnom poglavlju. Životne faze IS-a prema RUP-u su:

- 1) početna faza,
- 2) faza razrade,
- 3) faza izrade,
- 4) faza prelaska.

Navedene faze dopuštaju da proces razvoja bude predstavljen jednim ciklusom, kao kod vodopadnog pristupa. Međutim, bit je u ponavljanju ciklusa u navedenim fazama. Svaka od navedenih faza ima specifičan cilj i na kraju kontrolnu točku (engl. *milestone*), koja označuje ostvarenost cilja.

Osnovni cilj početne faze je da adekvatno sagleda opseg sustava kao osnovu za inicijalno provjeravanje troškova razvoja. U ovoj fazi se razvijaju poslovni slučajevi koji uključuju poslovni kontekst i ključne faktore uspjeha. Kao nadopuna poslovnog slučaja potrebno je napraviti osnovni dijagram slučajeva korištenja, projektni plan, inicijalnu procjenu rizika i opis projekta. On sadrži osnovne projektne zahtjeve, ograničenja i ključne značajke. Nakon dovršetka ove faze projekt se provjerava na temelju sljedećih kriterija:

- suglasnost svih dionika o definiciji opsega i procjene odnosa trošak/raspored;
- razumljivost zahtjeva, čime se postiže vjerodostojnost osnovnih dijagrama korištenja;
- vjerodostojnost procjene odnosa troška/rasporeda, prioriteta i rizika i razvojnog procesa;
- dubina i širina svake arhitekture prototipa koji će biti razvijen;
- uspostavljenost skupa osnovnih aktivnosti (engl. *baseline*), s kojim će se uspoređivati stvarni izdatci i planirani rashodi.

Kontrolna točka ove faze se naziva *lifecycle objective milestone*. Ako projekt nije zadovoljio kriterije definirane kontrolnom točkom ove faze, može biti prekinut ili se može ponoviti početna faza.

Primarni cilj faze razrade je identificirati i smanjiti rizike svih ključnih stavki koje se identificiraju analizom do kraja ove faze. Ovo je faza u kojoj projekt počinje dobivati svoj oblik, odnosno provodi se analiza problemske domene, na temelju koje se dobivaju osnovne aktivnosti (kostur) projekta. Kontrolna točka ove faze se naziva *lifecycle architecture milestone*, prema kojoj završetak faze označuje:

- identificiranost i opis većine slučajeva korištenja i njihovih sudionika, što podrazumijeva 80 % izrađenosti modela slučaja korištenja;
- opis arhitekture softvera;

- postojanje izvršne arhitekture, kojom se realizira značenje slučajeve korištenja;
- poslovni slučajevi i lista revidiranih rizika;
- plan odvijanja cjelokupnog projekta;
- prototipovi kojima se ublažava identificirani tehnički rizik.

Ako projektom nisu realizirane prethodne značajke, još uvijek nije kasno za odustajanje od projekta ili ponavljanje ove faze. Prelaskom na sljedeću fazu projekt ulazi u visokorizično stanje, u kojem su promjene vrlo štetne i teško izvodive.

Osnovni cilj faze izgradnje je izgradnja softverskog sustava. U ovoj fazi je glavni fokus na razvoju komponenti i drugih značajki razvijenog sustava. To je faza kad se odvija kodiranje. Izlazni rezultat ove faze je proizvod spreman za predaju krajnjim korisnicima na korištenje, a mora sadržavati:

- softverski proizvod integriran na adekvatnu platformu,
- korisničke upute,
- opis trenutačnog stanja.

Kontrolna točka ove faze se naziva *initial operational capability milestone*. Razvojni kriteriji za evolucijsku fazu su:

- Je li razvijeni proizvod stabilan i zreo za implementaciju u korisničko okruženje?
- Jesu li svi dionici razvoja spremni za prelazak u korisničko okruženje?
- Je li odnos planiranog troška i ostvarenog troška prihvatljiv?

Osnovni cilj faze prelaska jest prelazak s faze razvoja u fazu proizvoda, čineći tako IS razumljivim za krajnjeg korisnika. Osnovne aktivnosti ove faze uključuju beta-testiranje (kako bi se utvrdila usklađenost razvijenog IS-a s očekivanjima krajnjih korisnika) i obuku krajnjih korisnika. Provjerava se kvaliteta proizvoda u skladu s postavljenom razinom kvalitete u početnoj fazi. Kontrolna točka ove faze naziva se *product release milestone* i označuje kraj razvojnog ciklusa, a opisana je sljedećim kriterijima:

- ostvarenost upotrebljivosti sustava,
- ostvarenost konkurentske prednosti koja je određena temeljnim planom,
- ostvarenost zadovoljstva naručitelja.

### 4.5 Agilne metode

Iako su klasične metodologije donedavno dominirale na području razvoja sustava, brojna mišljenja i nekoliko istraživanja jasno pokazuju rastuću popularnost agilnih

metodologija (Aziz Butt, Piñeres-Espitia, Ariza-Colpas & Tariq, 2022). Točnije, prema izvještaju *State of Agile Report* poduzeća *VersionOne*, 97 % organizacija primjenjuje agilne metodologije u nekom od oblika (VersionOne, 2018). Agilna metodologija je specifičan pristup koji se primjenjuje kod razvoja softvera. Agilno označuje "kvalitetu agilnosti; spremnost za kretanje; bespriječnost, aktivnost, spretnost u pokretu." Drugim riječima, navedeno se odnosi na metode razvoja softvera koje kontinuirano pokušavaju udovoljiti krajnjem korisniku uz pomoć bržih i jednostavnijih procesa razvoja softvera. Odgovor na potrebne iteracije je agilnost jer se upravo kod agilnih metodologija zahtjevima upravlja iterativno za vrijeme čitavog procesa razvoja (Schön, Thomaschewski, & Escalona, 2017).

Navedeno znači da je implementacija inkrementalnih promjena u zahtjevima dobrodošla, bez obzira na fazu u kojoj se razvoj nalazi, pogotovo ako je riječ o redizajnu (Curcio, Navarro, Malucelli, & Reinehr, 2018).

Također, agilni pristup omogućuje da se sustav za vrijeme razvoja lakše prilagođava povratnim informacijama korisnika, ali i kontinuiranom mijenjanju samih zahtjeva (Maguire, 2013). Autori (Heikkila, Damian, & Paasivaara, 2015) navode kako je agilni pristup adekvatniji i omogućuje lakšu prilagodbu kada nastupe promjene, bilo u prioritetima zahtjeva ili promjena u samom sadržaju zahtjeva.

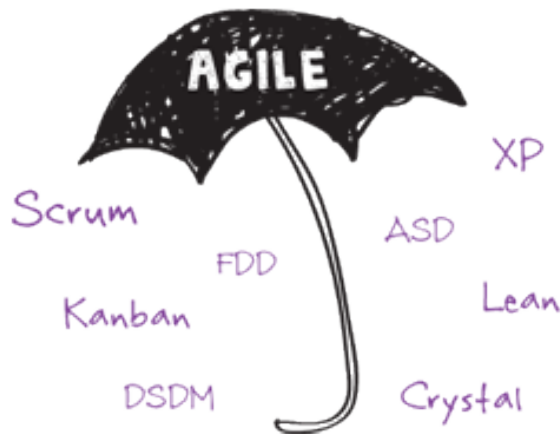
Centralne vrijednosti agilne metodologije su (Abrahamsson, Salo, Ronkainen, & Warsta, 2002):

- individue i interakcije imaju prioritet nad procesima i alatima,
- razvijen softver je važniji od kompleksno izrađene dokumentacije,
- kontinuirana suradnja s krajnjim korisnicima (kupcima) je važnija od specifikacije zahtjeva,
- adekvatno odgovaranje na promjene ima prioritet nad slijepim pridržavanjem plana.

Pojava agilnih metodologija je podijelila zajednicu dizajnera softvera na (1) tradicionaliste i (2) agiliste – naravno, svatko od njih ističe superiornost vlastite metodologije (Nerur, Mahapatra, & Mangalaraj, 2005).

U kategoriju najpoznatijih agilnih metodologija ubrajamo metode kao što su (Campanelli & Parreiras, 2015): (1) *scrum*, (2) *dynamic system development - DSDM*, (3) ekstremno programiranje – XP, (4) *feature-driven development – FDD*, (5) *adaptive software development – ASD*, (6) *kanban*, (7) *lean* i (8) *crystal*.

Osnovna razlika među agilnim metodologijama je faktor prilagodljivosti.



*Slika 4.4 Najpoznatije agilne metode (Collabnet, 2017)*

Osnovne karakteristike najčešće korištenih agilnih metodologija prikazane su u tablici 4.1. u nastavku.

*Tablica 4.1 Najpoznatije agilne metode (Collabnet, 2017)*

Metodologija	Karakteristike
SCRUM	Neovisnost, malen opseg, razvojni timovi, ciklus od 30 dana
DSDM	Vremensko određivanje
XP	Razvoj vođen korisnikovim potrebama, mali timovi, dnevni napredak
FDD	Proces od pet koraka, objektno orijentirano modeliranje
ASD	Kolaboracija, razvoj vođen misijom, iterativnosti

## 4.6 Metodologija DASPIS

DASPIS (dinamički aspekti strateškog planiranja informacijskih sustava) je metodologija koja objedinjuje prethodne metode i tehnike strateškog planiranja informacijskog sustava. Štoviše, DASPIS je puno više od predstavljenih jer uključuje sve faze životnog ciklusa IS-a, a ne samo fazu planiranja.

Razumijevanje potreba unapređenja, odnosno uvođenja IS-a/IT-ja, u skladu s razmišljanjima iz prethodnog odjeljka, ukazuje na potrebu poznavanja cjelokupnog poslovnog sustava. U tu svrhu potrebno je upoznati poslovne procese svih razina te njihovo ponašanje. Upravo bi promatranje rada organizacije trebalo dopuniti analizu

trenutačnog stanja uočavanjem kritičnih točaka u kojima IS može doprinijeti učinkovitosti organizacije. Ovisno o rezultatima analize trenutačnog stanja može se ukazati potreba za kompletnim preustrojem organizacijskog sustava, što je u tom slučaju također potrebno izvesti. Cjelokupni IS/IT potrebno je funkcijski i ekonomski optimirati. Kako bi IS u organizaciji mogao profunkcionirati, potrebno ga je implementirati i educirati korisnike, kao i osigurati sve potrebne uvjete za zaštitu i praćenje učinaka njegova rada.

Sljedećom definicijom jasno je definiran osnovni pristup metodologije strateškog planiranja informacijskih sustava (SPIS). Strateško planiranje informacijskih sustava je dugoročno planiranje korisnih učinaka informacijskog sustava i upotrebe informacijskih tehnologija u poslovanju, a u sklopu strateškog planiranja razvoja poslovnog sustava kao cjeline (Brumec & Vrček, 2001).

### 4.6.1 Formalizacija DASPIS-a

Metodologija SPIS pomaže napraviti IS koji bolje odgovara zahtjevima organizacije. Njezina primjena rezultira strateškim planom razvoja IS-a koji odgovara budućem razdoblju. Kao i svaki proces, tako i proces strateškog planiranja informacijskog sustava mora sadržavati ulazne vrijednosti, koje obradom prelaze u izlazne vrijednosti. U skladu s "tradicionalnim mudrostima", ulazne vrijednosti procesa strateškog planiranja trebaju obuhvatiti vanjsko i unutarnje okruženje, i to (Hill & Westbrook, 1997):

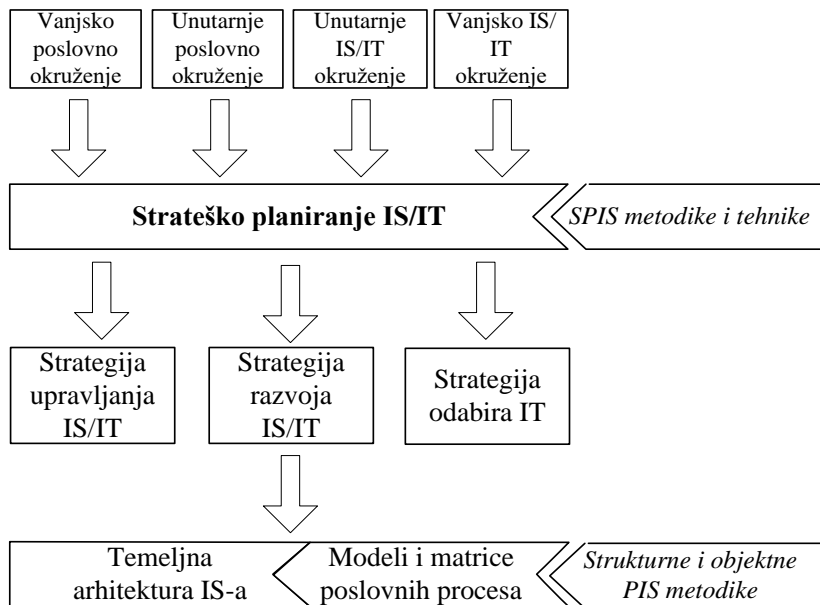
- unutarnje poslovno okruženje (trenutačnu organizacijsku strategiju, poslovne činjenice, raspoložive resurse, trenutačni i planirani način rukovođenja organizacijom, trenutačne i buduće poslovne tehnologije i trenutačnu poslovnu efikasnost);
- unutarnje okruženje IS-a/IT-ja (trenutačni IS i primijenjeni IT te njihov utjecaj na poslovno upravljanje, trenutačnu programsku podršku, organizacijsku sposobnost prihvaćanja novih tehnologija, znanje i iskustvo dizajnera IS-a);
- vanjsko poslovno okruženje (ekonomski sustav, tržište, industriju i konkurentsko okruženje organizacije);
- vanjsko okruženje IS-a/IT-ja (trendovi razvoja IT-ja i njihova primjena kod drugih poslovnih partnera, primjena IT-ja kod poslovnih konkurenata).

Izbor metoda predloženih SPIS-om vrši se sukladno promatranom slučaju. Gotovo da ne postoji ista grupa metoda koja bi se mogla primijeniti na sve slučajeve (Alamri, Almutiri, Ballahmar, & Zafar, 2016). Štoviše, to je kreativan proces kombiniranja

analitičkih i formalnih metoda, pri čemu je potrebno poštovati temeljne odrednice metodologije SPIS (Brumec, Dušak & Vrček, 1998):

- razlog svakog aspekta poslovnog preustroja je poboljšanje svojstava sustava,
- razvoj IS-a mora biti ciljno orijentiran,
- ciljevi trebaju biti mjerljivi.

Formalizacija procesa DASPIS-a prikazana je grafički na slici 4.5.



Slika 4.5 Formalizacija SPIS postupka (Brumec J. , *Strategic Planing of IS, 1998*)

Izlazni rezultat procesa SPIS-a su tri usko povezane strategije na temelju kojih se strukturnim i/ili objektivnim metodama projektiranja informacijskih sustava (PIS) trebaju izraditi modeli i matrice poslovnih procesa. Te tri strategije su:

- strategija upravljanja IS-om/IT-jem – uključuje opće principe i organizacijske modele upravljanja razvojnim procesom IS-a te primjene IT-ja kao temeljnih pravila organizacije;
- strategija razvoja IS-a/IT-ja – definira metode analize i oblikovanja te standarde i procedure sigurnosti i kvalitete uklapanja poslovnih funkcija u cjelokupni IS;

#### 4: Pristup i metode razvoja informacijskih sustava

- strategija odabira IT-ja – postavlja standarde procedura odabira, kupnje, najma, implementacije, primjene i održavanja IT-ja.

U tablici 4.1., pored svakog problemskog koraka i predloženih metodologija i tehnika za njihovo rješavanje, prikazan je stupac sa zahtijevanim ulaznim podacima i očekivanim rezultatima te korisnost svake od predloženih metodologija i tehnika. Rezultate procesa SPIS-a potrebno je dokumentirati putem sljedećih modela (Vidačić & Brumec, 2013):

- modeli poslovnih procesa postojeće i preustrojene organizacije,
- modeli poslovnih podataka,
- modeli programa i procedura,
- modeli tehničkih resursa,
- plan aktivnosti razvoja novog IS-a.

Tablica 4.2 Pregled problemskih koraka DASPIS-a te metoda i tehnika za njihovo rješavanje (Hell, Vidačić & Brumec, 2013)

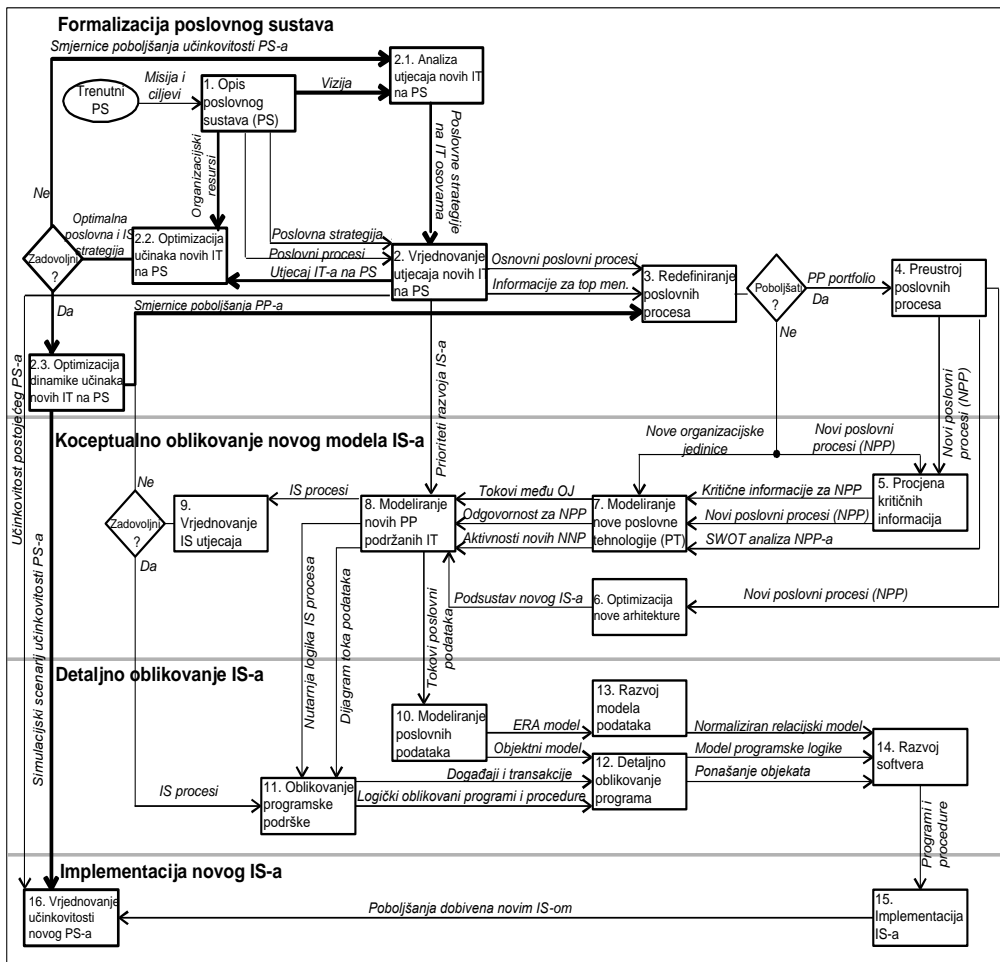
Problemski koraci DASPIS-a	Metode i tehnike (§ -strateške, # -strukturalne, x -objektne)	Ulaz i izlaz procesa u koraku <i>Ulaz / Izlazi</i>	Korisnost Vrlo snažno Upotreblji vo
1. Opis poslovnog sustava (PS)	Intervjuiranje	<i>Misija i ciljevi postojećeg PS-a</i> / Poslovna strategija; Poslovni proces (PP); Organizacijski resursi	
2.1. Analiza utjecaja novih IT-jeva na poslovni sustav	§ pSWOTa	<i>Vizija</i> / Poslovne strategije na IT osnovama;	V
2. Vrednovanje utjecaja novog IT-ja na PS	§ <i>Balanced Scorecard</i> § <i>Balanced Scorecard</i> zasnovan na matričnoj algebri § matrica BCG § model 5F	<i>PP</i> / Učinak postojećeg PS-a <i>Poslovne strategije na IT osnovama</i> /Utjecaji IT-ja na PS <i>Poslovna strategija</i> / razvojni prioriteta IS-a <i>Poslovna strategija</i> / Informacije za top menadžment	V V S U
	§ model <i>Value-chain</i>	<i>PP</i> / Osnovni i potporni poslovni procesi (PPP)	V
2.2. Optimizacija učinaka novih IT-jeva na PS	§ Linearno programiranje	<i>Utjecaji IT-ja na PS; Organizacijski resursi</i> / Optimalne IT i poslovne strategije	V
2.3. Optimizacija dinamike implementacije novog IS-a	§ Sustavska dinamika	<i>Optimalne IT i poslovne strategije</i> / Simulacijski scenarij učinkovitosti PS-a; Smjernice poboljšanja poslovnih procesa	V
3. Redefiniranje poslovnih procesa	# dekompozicija BSP # Analiza životnog ciklusa resursa	<i>PPP</i> / Nove organizacijske jedinice (OJ) <i>Temeljni resursi sustava</i> / Portfolio poslovnih procesa	S S
4. Preustroj poslovnog sustava	§ BPR	<i>Portfolio poslovnih procesa</i> / Novi poslovni procesi (NPP)	S
	§ SWOT	<i>Portfolio poslovnih procesa</i> / SWOT analiza NPP-a	V



5. Procjena kritičnih informacija	§ analiza CFS (Rockart) # analiza <i>Ends-Means</i>	<b>NPP</b> / Kritične informacije NPP-a <b>NPP</b> / Informacije za postizanje djelotvornosti rada i učinkovitosti poslovanja	S U
6. Optimizacija nove arhitekture IS-a	# Matrica poslovne tehnologije # Analiza osjetljivosti, Genetički algoritmi	<b>NPP</b> / Veze poslovnih procesa <b>Veze poslovnih procesa/ Klasteri</b> ; Podsustav novog IS-a	V S
7. Modeliranje nove Poslovne tehnologije (PT)	# Dijagram radnih tijekova # Organizacijski dijagram # Akcijski dijagram	<b>NPP</b> / Odgovornost za NPP <b>Novi OJ-ovi</b> / Tijekovi među novim OJ-ima <b>NPP</b> / Aktivnosti NPP-a	V S U
8. Modeliranje novih poslovnih procesa poduprti IT-jem	# Dijagram tijeka podataka (DTP) # Akcijski dijagram (AD)	<b>NPP</b> / NPP poduprti IT-jem (procesima IS-a); Tijekovi podataka; Poslovni podatci <b>proces IS-a</b> / Unutarnja logika procesa IS-a	V S U
9. Vrednovanje Učinaka IS-a	# Simulacijsko modeliranje	<b>proces IS-a</b> / Upute za poboljšanje PP-a	U
10. Modeliranje poslovnih podataka	# model ERA ⌘ Objektni model	<b>Poslovni podatci</b> / model ERA <b>Poslovni podatci</b> / Objektni modeli	V S
11. Oblikovanje programske potpore	# dijagram HIPO ⌘ dijagram <i>Transition</i>	<b>proces IS-a</b> / Logički oblik programa <b>Tokovi podataka</b> / Događaji i transakcije	V S
12. Detaljno oblikovanje programa i procedura	# Akcijski dijagram ⌘ Scenariji objekata	<b>Logičko oblikovane programske procedure</b> / Logički modeli programa <b>Model objekata; Događaji</b> / Ponašanje objekata	S S
13. Razvoj modela podataka	# Relacijski model; Normalizacija	<b>model ERA</b> / Relacijski model	V
14. Razvoj softvera	# alati CASE i 4GL ⌘ alati OO-CASE	<b>Logički model programa; Relacijski model</b> / Programi i procedure <b>Ponašanje objekata</b> / procedure OO-a	S S
15. Implementacija IS-a	Studija slučaja; Poslovne igre	<b>Programi i procedure</b> / Učinak novog IS-a	S
16. Vrednovanje učinaka novog PS-a	§ <i>Balanced scorecard</i> § <i>Balanced Scorecard</i> zasnovan na matričnoj algebri	<b>Učinkovitost starog PS-a, Svojstva novog IS-a</b> / Mjerenje uspjelosti <b>Simulacijski model učinkovitosti PS-a; Svojstva novog IS-a</b> / Ostvareni učinci novog IS-a na PS	V

Polazište metodologije SPIS je analiza usklađenosti trenutnog stanja poslovnog sustava i definicije misije i glavnih ciljeva. Problemski koraci te metodologije i tehnike SPIS-a prikazani su u tablici 4.2. Drugim korakom (*Vrednovanje utjecaja novog IT-ja na poslovni sustav*) određuju se ona područja u kojima IT može najviše doprinijeti učinkovitosti poslovnih procesa. Mjerenje performansi poslovnog sustava, kao i utjecaja IT-ja na njihovo poboljšanje, mora biti izvedeno u skladu s poslovnim ciljevima. Deveti korak – *Vrednovanje efekata IS-a*, prema (Brumec, Dušak & Vrček, 1998) – centralni je korak u SPIS-u. Vrednovanje utjecaja novog IS-a na buduće poslovne procese obavlja se simulacijama. Provođenjem različitih scenarija uz zadane mjerljive parametre odabire se najpovoljniji scenarij.

Na temelju prethodne tablice na slici 4.6 prikazan je metamodel metodologije SPIS.



Slika 4.6 Metamodel metodologije DASPIS (Vidačić &amp; Brumec, 2013)

Kao što se može vidjeti na slici, metamodel metodologije SPIS može se podijeliti na četiri povezane cjeline:

- formalizacija poslovnog sustava,
- konceptualno oblikovanje novog modela IS-a,
- detaljno oblikovanje IS-a,
- implementacija novog IS-a.

#### 4.6.2 Dosadašnja primjena DASPIS-a

Prethodna metodologija primijenjena je i potvrđena na različitim poslovnim i proizvodnim sustavima. Na taj način autori su potvrdili svoje rezultate istraživanja o izboru najprikladnijih metodologija za maksimiranje utjecaja novog informacijskog sustava u preustroju organizacije s obzirom na njezinu djelatnost. U nastavku slijedi nekoliko bitnijih zaključaka (Brumec & Vrček, Strategic Planing of Information Systems (SPIS)- A Survey of Metodology, 2002):

Jedan od šest bitnijih projekata, prema mišljenju gore navedenih autora SPIS-a, proveden je u farmaceutskoj tvrtki srednje veličine. Cilj projekta je bio, prema želji rukovodstva, korištenje informacijskih tehnologija u svrhu optimiranja dobavnog lanca. Uspjeh provođenja projekta inicirao je širenje poslova na distribucijski lanac.

Vrlo velika tvornica stakla, koja je dio međunarodne staklarske tvrtke, sadrži kombinaciju kontinuiranih i diskretnih, vrlo specifičnih procesa proizvodnje standardiziranih proizvoda. Rukovodstvo ove organizacije željelo je implementaciju novog IS-a u svrhu optimiranja i nadgledanja cijene proizvodnje i potpore kontrolnih funkcija. Nakon studije SPIS organizacija je započela s razvojem novog sustava ERP-a, baziranog na funkcionalnoj vezi između standardnih programskih paketa i svojih vlastitih rješenja.

Velika međunarodna tvornica hrane također je primjer kontinuirane i standardizirane proizvodnje. Cilj ove tvornice je bio efikasno organizirati proizvodnju uvođenjem sustava ERP-a. Na osnovi rezultata studije SPIS donesena je odluka o kupnji kritičnog programskog modula.

SPIS se koristio i kod neprofitnih organizacija. Primjer takve ustanove je nacionalni ured za zapošljavanje (HZZ). Cilj HZZ-a bio je iskoristiti moderne IKT-e, konkretno internet, kako bi se približio korisnicima te ubrzao i poboljšao pouzdanost usluge. Drugi cilj je bio optimiranje administrativnih poslova koji su okupirali administraciju, ostavljajući tako negativan imidž poduzeća. Projekt se provodi točno prema preporukama i općim rješenjima SPIS-a.

Državna administracija je također neprofitna organizacija u kojoj je primijenjen SPIS. Ovdje je također cilj bio optimiranje suvišnih administrativnih poslova s namjerom postizanja brže reakcije na različite zahtjeve javnosti. Nakon studije SPIS projekt je odgođen jer je SPIS ukazao na neočekivanu količinu potrebnih promjena.

Hrvatske željeznice (HŽ) su imale za cilj radikalne dugotrajne procese promjena – transformaciju iz državne neprofitne organizacije u profitno orijentiranu transportnu

tvrtku. Želja je bila koristiti informatičke tehnologije kako bi transformaciju učinili lakšom i efikasnijom. Studija SPIS je prihvaćena, ali je implementacija odgođena zbog nedostatka financijskih sredstava. Ovime je dodatno ukazano na široko područje primjene metodologije SPIS.

Provedenim istraživanjima, sukladno postavljenim ciljevima i hipotezama rada, obuhvaćeni su koraci SWOT analize i vrednovanje utjecaja IS-a, koji spadaju u kontekst cjeline konceptualnog modeliranja. Kao što je prije navedeno, primjena metodologije SPIS rezultira strateškim planom razvoja IS-a, koji odgovara budućem razdoblju razvoja organizacije. Zbog promjenjivosti okolnosti u tom budućem razdoblju, realizaciju strateškog plana razvoja IS-a potrebno je nadzirati te vrednovati ostvaren utjecaj IS-a prateći učinke poslovnog sustava. U tu svrhu istražena je mogućnost povezivanja SWOT analize i BSC-a s ciljem razvoja prognostičkih modela zasnovanih na metodologiji sustavske dinamike. Korištenje takvih prognostičkih modela trebalo bi povećati vrijednost informacija o budućim utjecajima IS-a na učinkovitost organizacije.

## 4.7 Literatura

1. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile Software Development Methods: Review and Analysis. *VTT publication 478*, 107.
2. Alamri, S., Almutiri, N., Ballahmar, H., & Zafar, A. (2016). Strategic Information System Planning: A Case Study of a Service Delivery Company. *International Advanced Research Journal in Science, Engineering and Technology Vol. 3*, Issue 5, 78-84.
3. Aziz Butt, S., Piñeres-Espitia, G., Ariza-Colpas, P., & Tariq, M. I. (2022). Project management issues while using agile methodology doi:10.1007/978-3-030-94238-0\_12
4. Barry, B., & Wilfred, J. H. (2000). *Spiral Development: Experience, Principles and Refinements*. Pittsburgh: SEI Joint Program Office.
5. Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, str. 61-72.
6. Boehm, B. W., Defense, T. R. W. & Group, S. (1987). A Spiral Model of Software Development and Enhancement'.
7. Brumec, J. (1997). A Contribution to IS General Taxonomy, *Zbornik radova FOI-a, god.21, br. 1.* (22). 1997.
8. Brumec, J. (1998). Strategic Planing of IS. *Journal of Information and Organisational Science*, 23(2), str. 11-26.
9. Brumec, J., & Vrček, N. (2001). Procjena učinaka korištenja IT metodom Balanced Scorecard. *Zbornik radova CASE 13*, pp 175-183, CASE d.o.o. Rijeka 2001.
10. Brumec, J., & Vrček, N. (2002). Strategic Planing of Information Systems (SPIS)- A Survey of Methodology. *CIT*, Vol. 10, No. 3, str.: 225-232.
11. Brumec, J., Dušak, V. & Vrček, N. (1998). Structured and Object-Oriented Methods in a Complex Is Project, *Zbornik radova : journal of information and organizational sciences*, 22(2), pp. 119–136.
12. Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring – A systematic literature review. *The Journal of Systems and Software* 110, 85-100.
13. Collabnet. (2017). *What Is Agile Methodology?* Dostupno na <https://resources.collab.net/agile-101/agile-methodologies>
14. Curcio, K., Navarro, T., Malucelli, A., & Reinehr, S. (2018). Requirements Engineering: a systematic mapping study in agile software development. *Journal of Systems and Software*. Elsevier.
15. Davis, G. B. (1982). Strategies for information requirements determinatio. IBM SYST, VOL 2, 1.
16. Dunder, I., Juričić, V. & Pogarčić, I. (2012). Matrica poslovne tehnologije kao temelj razvoja informacijskog sustava za grafičko-nakladničku djelatnost, in *Proceedings of the 16th International Conference on Printing, Design and Graphic Communications*, pp. 95–107.

17. Fichman, R. G., & Kemerer, C. F. (1993). *Adoption of Software Engineering Process Innovations: The Case of Object Orientation*. Mitsloan.
18. Garača, Ž. (2008). *Poslovni informacijski sustavi*. Split: Ekonomski fakultet.
19. Heikkilä, V. T., Damian, D. L., & Paasivaara, M. (2015). A Mapping Study on Requirements Engineering in Agile Software Development u *Proceedings - 41st Euromicro Conference on Software Engineering and Advanced Applications SEAA*.
20. Hell, M., Vidačić, S. & Brumec, J. (2013.) Dynamic aspects of the strategic planning of information systems, *Journal of Information and Organizational Sciences*, 37(1), pp. 23–43.
21. Hill, T., & Westbrook, R. (1997). SWOT Analysis: It's Time for a Product Recall. *Long Range Planning*, Vol. 30, No. 1, pp. 46 to 52.
22. IBM. (2014). *IBM Rational Unified Process (RUP)*, Dostupno na: <http://www-01.ibm.com/software/rational/rup/>
23. Javanmard, M. & Alian, M. (2015). Comparison between Agile and Traditional software development, *Science Journal*, 36(3), pp. 43–49. Available at: <http://dergi.cumhuriyet.edu.tr/cumusciy@2015%0Ahttp://dergi.cumhuriyet.edu.tr/cumusciy>.
24. Kroll, P., & Kruchten, P. (2003). *Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison Wesley.
25. Maguire, M. (2013). Using human factors standards to support user experience and agile design. Universal Access in *Human-Computer Interaction*. Design Methods, Tools, and Interaction Techniques for e-Inclusion. UAHCI.
26. Manger, R. (2016). *Softversko inženjerstvo*. Zagreb: Element.
27. Martin, J., & Odell, J. (1997). *Object-Oriented Methods: A Foundation*, UML Edition (2nd Edition). Prentice Hall.
28. Molnár, Bálint & Benczúr, András & Béleczi, András. (2016). Formal approach to modeling of modern information systems. *IJISPM - International Journal of Information Systems and Project Management*. 4. 69-89. 10.12821/ijispm040404.
29. Muller, G. (2013). *System Modeling and Analysis: a Practical Approach*. Kongsberg: Buskerud University College.
30. Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of Migrating to Agile Methodologies. *COMMUNICATIONS OF THE ACM*, 73-78.
31. PMI. (2017). *A Guide to the Project Management Body of Knowledge (Pmbok Guide)*. USA: Project Management Institute.
32. Radovan, M. (1991). *Projektiranje informacijskih sustava*. Zagreb: Informator.
33. SBS. (2018). *What is spiral model?* Dostupno na: <http://www.selectbs.com/analysis-and-design/what-is-the-spiral-model>
34. Schmuller, J. (2004). *Teach Yourself UML in 24 Hours*, Third Edition. Indianapolis: Sams Publishing.

35. Schön, E. M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. *Computer Standards and Interfaces. Elsevier*, Vol. 49, , 79–91.
36. Soares, M. d., Vranckenb, J., & Verbraeckb, A. (2010). User requirements modeling and analysis of software-intensive systems. *The Journal of Systems and Software*.
37. Sommerville, I. (2011). *Software engineering*. Boston: Pearson Education, Inc.
38. VersionOne. (2018). *12 Annual State of Agile Report*. COLLAB.NET.





## 5 Utvrđivanje i modeliranje zahtjeva

U prethodnom poglavlju prikazani su pristupi i metode razvoja IS-a/IT-ja. Svaki od njih sadrži neki oblik analize zahtjeva. To znači da je, nakon što je utvrđena ključna uloga IS-a u kontekstu poslovnog sustava, potrebno pristupiti analizi i modeliranju zahtjeva. To znači da se za svaki utvrđeni cilj analiziraju i modeliraju korisnički zahtjevi. Faza utvrđivanja zahtjeva obuhvaća zahtjeve koji se opisuju prirodnim jezikom. Formalno modeliranje zahtjeva korištenje jedinstvenog jezika za modeliranje (npr. *Unified Modeling Language – UML*) i vodi prema fazi specifikacije zahtjeva. Prezentiranje prikupljenih zahtjeva na visokom stupnju apstraktnosti zove se modeliranje poslovnih zahtjeva (*engl. business requirement modelling*). Minimum koji trebaju odrediti vizualni modeli visoke razine je okvir željenih promjena u sustavu. To znači da je potrebno identificirati poslovne slučajeve i uspostaviti najvažnije poslovne klase.

### 5.1 Prikupljanje zahtjeva

U ovoj knjizi analiza i modeliranje zahtjeva su zasnovani na objektno orijentiranom pristupu. Osnovni razlog za korištenje tog pristupa je "lakoća" njegove razumljivosti i primjenjivosti. Objekti se izvode iz poslovnog sustava, umjesto da se temelje na tehnološkim postupcima, koji se zasnivaju na informatičkim pojmovima. Na taj način analiza zahtjeva postaje razumljivija većem broju dionika, a time i učinkovitija. S druge strane, svi sudionici razvoja IS-a koriste isti koncept i notaciju u skladu s fazama razvoja, čime je olakšan način komuniciranja dionika razvoja IS-a.

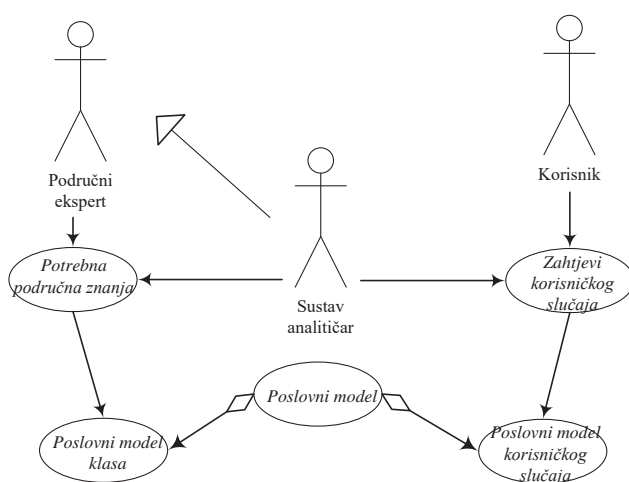
Nakon razumijevanja poslovnog sustava te utvrđivanja i dokumentiranja poslovnih zahtjeva, potrebno je utvrditi što softverska poboljšanja trebaju omogućiti krajnjem korisniku. Sukladno fazama životnog ciklusa IS-a to znači da ulazimo u fazu prikupljanja i analize zahtjeva. Utvrđivanjem onoga što softver treba (ili ne treba) omogućiti krajnjem korisniku detaljno se usmjeravaju daljnje aktivnosti u razvoju IS-a/IT-ja. Bez potpunog razumijevanja zahtjeva krajnjih korisnika pojačava se rizik investiranja u nove IS-ove/IT-jeve, tj. dovodi u pitanje ostvarenje željenih učinaka.

#### 5.1.1 Organizacija prikupljanja zahtjeva

Za organizaciju prikupljanja i za samo prikupljanje potrebno je manje tehničkih, a više socijalnih, komunikacijskih i menadžerskih vještina. Zahtjevima se definiraju poslovi i ograničenja koje sustav mora zadovoljavati. Iskazana ograničenja mogu biti klasificirana u kategorije sukladno pravilima sustava (sigurnost, propisan izgled i sl.). Zahtjevi se prikupljaju od korisnika i vlasnika razmatranog (pod)sustava. Postoje

mnoge tehnike koje je moguće primijeniti prilikom prikupljanja zahtjeva. Prikupljeni zahtjevi moraju se pažljivo analizirati kako bi se eliminirala ponavljanja i kontradikcije. Zbog toga je potrebno da ih korisnici revidiraju. Prikupljene zahtjeve od strane korisnika potrebno je zapisati, klasificirati i numerirati u obliku dokumenata (engl. *requirements document*). Dokumenti moraju biti strukturirani u skladu s definiranim obrascima. Dokumentacija zahtjeva može uključivati i dijagramske tehnike, kao što su UML dijagrami, koji će u ovom poglavlju biti posebno analizirani.

Analitičar sustava, dionik čija je uloga već prije objašnjena, je ključna osoba u prikupljanju zahtjeva. On mora istražiti zahtjeve promatranog podsustava konzultirajući se s korisnicima i ekspertima. U nekim slučajevima analitičar sustava može posjedovati dovoljno područnih znanja te mu zbog toga nije potrebna konzultacija s ekspertom. Općenitu situaciju moguće je prikazati kao na slici 5.1. Pritom strelica između analitičara sustava i područnog eksperta označuje vezu generalizacije. Istraživanje zahtjeva podrazumijeva područna ekspertna znanja. Ona obuhvaćaju široki skup poslovnih pravila koja se primjenjuju u mnogim organizacijama. S druge strane, korisnik postavlja zahtjeve koji su karakteristični, pa možda čak i jedinstveni za promatranu organizaciju. Zadatak analitičara sustava je uskladiti ta dva skupa zahtjeva u poslovni model. Kao što se može vidjeti na slici 5.1., poslovni model sadrži dva tipa modela: (1) poslovni model klasâ i (2) poslovni model slučaja korištenja. Poslovni model klasâ je model visoke razine (ne detaljno razrađen) klasnog dijagrama koji sadrži objekte sustava i odnose među njima. Poslovni model slučaja korištenja je također model visoke razine dijagrama slučaja korištenja pomoću kojeg se identificiraju osnovni funkcionalni blokovi sustava.



Slika 5.1 Analiza zahtjeva: dijagram slučaja korištenja

Tradicionalne metode i tehnike prikupljanja zahtjeva koje koristi analitičar sustava uključuju: (1) intervjuiranje, (2) anketiranje, (3) promatranje i (4) izučavanje poslovnih dokumenata. O složenosti posla ovisi i složenost primjene svake od navedenih metoda. Što je veća kvaliteta prikupljanja zahtjeva, manji je rizik projekta.

Bez obzira na metodu ili tehniku koja se koristi, analitičaru sustava mora biti jasno da prilikom prikupljanja informacija troši vrijeme korisnika. Bez obzira na motivaciju korisnika za unapređenjem, učestalo kontaktiranje uznemiruje i negativno utječe na kvalitetu informacija koje korisnik pruža. To znači da je u što manjem broju susreta potrebno prikupiti što više potrebnih informacija. U svakom sljedećem susretu potrebno je korisniku pokazati razinu ostvarenog napretka. Upravo zato je potrebno pažljivo pristupiti te jasno procijeniti potrebno vrijeme kako bi se napravio plan prikupljanja. Na taj način se postiže ozbiljnost samog procesa prikupljanja informacija, što će utjecati i na ozbiljnost pristupa krajnjih korisnika. Navedeno je ključna stvar za dobivanje kvalitetnih informacija.

### 5.1.2 Intervjuiranje korisnika i područnih eksperata

Intervju je osnovna tehnika za utvrđivanje činjenica i prikupljanje potrebnih informacija. Većina intervjuja se provodi s korisnicima, iako je u ovom slučaju moguće i intervjuiranje eksperta. Intervjuiranjem se dobiva uvid u modele slučajeva korištenja koji se odmah mogu oblikovati i dokumentirati u obliku različitih dijagrama. Dijagramom se lakše prezentira shvaćeno stanje i dobiva potvrda prikupljenog koju daje intervjuirani korisnik.

Intervjuiranje (područnog) eksperta često rezultira transferom znanja koje je potrebno analitičaru sustava, što znači da je analitičaru sustava lakše prikupiti zahtjeve od eksperta jer on kvalitetnije objasni i definira ono što je potrebno od „običnog“ korisnika. Čest je slučaj da taj korisnik ima nejasnu sliku onoga što želi postići, tj. nejasnu sliku svojih zahtjeva. Zbog toga može doći do njegovog opiranja ili pak neće biti u mogućnosti formulirati zahtjeve u razumljivoj terminologiji. Također se može dogoditi da ponudi zahtjeve koji nadilaze planirane resurse.

Zbog navedenog je intervjuiranje (običnog) korisnika često puta značajno teže nego što je to intervjuiranje eksperata. Postoje dva osnovna oblika intervjuiranja:

- strukturirani (formalni) i
- nestrukturirani (neformalni).

Strukturirani intervju potrebno je pripremiti unaprijed. On ima jasan raspored prethodno utvrđenih pitanja koja mogu biti koncipirana tako da se zaokruži ponuđeni

odgovor ili da se proizvoljno odgovori. Ako se provodi intervju s više osoba o istom problemu, u slučaju samostalnog pisanja odgovora ispitanika mogu se pojaviti problemi klasifikacije odgovora (primjerice, svaki ispitanik daje različiti odgovor). Iz tog razloga je potrebno oblik intervjuiranja prilagoditi određenoj situaciji. Tako je, primjerice, ako ispitujemo zadovoljstvo veće skupine krajnjih korisnika, prikladno unaprijed pripremiti moguće odgovore, odnosno provesti strukturirani oblik intervjua.

Rezultate strukturiranog intervjua moguće je nadopuniti nestrukturiranim intervjuom. Nestrukturirani intervju je sličniji informativnom razgovoru u kojem se mogu pojasniti određene nedoumice, npr. neočekivani odgovori iz strukturiranog intervjua.

### 5.1.3 Anketiranje

Anketiranje je efikasan način za prikupljanje informacija od kupaca. Ono se uobičajeno koristi kao dodatak intervjuima, ali ne umjesto njih. Prednost anketa je u tome što ispitanik ima vremena razmisliti prije nego odgovori na pitanje. Međutim, nedostatak je u tome što ispitanik ne može dobiti objašnjenje pitanja, što može dovesti do krivih odgovora. Kad je anketiranje u pitanju, najčešći tipovi pitanja s ponuđenim odgovorima su:

- pitanja s više izbora odgovora (u njima ispitanik mora svoje mišljenje iskazati odabirom postavljenih tvrdnji),
- pitanja u svrhu ocjenjivanja (takvim pitanjima ispituje se mišljenje ispitanika o izjavi),
- pitanja u svrhu rangiranja (odgovaranjem na pitanje korisnik/ispitanik ocjenjuje brojem ili postotkom ponuđenu tvrdnju).

Dobro oblikovan upitnik mora biti jednostavan za odgovaranje. Broj pitanja je potrebno uskladiti s potrebnim vremenom za ispunjavanje cijelog upitnika.

### 5.1.4 Promatranje

Postoje situacije u kojima analitičar sustava nije u mogućnosti saznati sve potrebne informacije pomoću intervjua i anketiranja. To se može dogoditi kada korisnik možda nije u mogućnosti jasno definirati ili objasniti traženi slučaj korištenja ili ga možda jednostavno dovoljno ne poznaje. U takvim slučajevima promatranjem se mogu utvrditi željene informacije. Uostalom, promatranje i je najsigurniji način da se utvrdi trenutačno stanje. Promatranje se pojavljuje u dva oblika:

- pasivno promatranje, u kojem analitičar sustava promatra poslovne aktivnosti bez uplitanja ili prekidanja rada (u nekim slučajevima to se može zamijeniti videokamerama, čije se snimke naknadno proučavaju);
- aktivno promatranje, u kojem analitičar sustava sudjeluje u izvođenju aktivnosti i postaje sastavni dio poslovnoga slučaja.

U svrhu reprezentativnosti uzorka promatranje mora biti provedeno u razdvojenim trenutcima. To znači da je promatranje istog poslovnog slučaja potrebno ponoviti nakon nekog vremena. Osnovna poteškoća kod promatranja je to da se zaposlenici drukčije ponašaju kad ih se promatra. Zbog toga mogu promaknuti ključni elementi koje je potrebno unaprijediti.

### 5.1.5 Izučavanje dokumentacije i IS-a

Izučavanje dokumentacije i softverskih sustava omogućuje da se na jednom mjestu saznaju i slučajevi korištenja i potrebna ekspertna znanja. Zahtjevi slučaja korištenja se istražuju studijama postojeće organizacijske dokumentacije te obrascima i izvješćima unutar sustava. Pritom se najznačajnije informacije za dobivanje slike o stanju sustava mogu dobiti iz zapisa o defektima i promjenama zahtjeva postojećeg sustava. Neki od organizacijskih dokumenata mogu biti: (1) poslovni obrasci, (2) radne procedure, (3) opis poslova, (4) pravila, (5) poslovni planovi itd.

### 5.1.6 Usklađivanje i validacija zahtjeva

Opsežnije prikupljanje zahtjeva može rezultirati konfliktnim rezultatima. Naime, čest je slučaj da prikupljeni zahtjevi budu proturječni ili se međusobno preklapaju. Neki pak mogu biti dvosmisleni ili pretjerani. Zbog toga je potrebno provesti njihovo usklađivanje i validaciju – tek ih je nakon toga moguće dokumentirati.

Proces validacije i usklađivanja zahtjeva sastavni je dio procesa dokumentiranja zahtjeva. Usklađivanje i validaciju moguće je provoditi paralelno tijekom prikupljanja zahtjeva. Odmah nakon prikupljanja zahtjevi se podvrgavaju promatranju. Ovakva dinamika karakteristična je za moderne metode. Međutim, detaljno i sveobuhvatno usklađivanje zahtjeva moguće je jedino kada su prikupljeni svi zahtjevi. Usklađivanje zahtjeva obično se radi na nacrtu dokumenta. To znači da se usporedba i eventualne promjene rade na osnovi liste zahtjeva u nacrtu dokumenta, tj. izbacuju se sporni, a dodaju novi zahtjevi. Za razliku od uspoređivanja, validacijom se ulazi u srž dokumenta. To znači da svaki zahtjev za validaciju mora biti jasno definiran i klasificiran.

Izbor IS/IT projekata je utvrđen aktivnostima u fazi planiranja. Međutim, njihova detaljna razrada provodi se prilikom faze analiziranja zahtjeva. Detaljnom razradom

zahtjeva ostvaruje se definiranje granica sustava. Dijagramska tehnika kojom se utvrđuju granice sustava je kontekstni dijagram. Postoje mnogi razlozi zašto se prikupljeni zahtjevi mogu naći izvan planiranih granica. Jedan od razloga je teškoća implementacije zahtjeva u informacijski sustav u planiranim okvirima resursa, zbog čega se taj zahtjev izostavlja. S druge strane, zahtjevi mogu imati niski prioritet, što također stavlja zahtjev na čekanje, odnosno odgađa za neku kasniju fazu informatizacije.

Na osnovi jasno definiranih i numeriranih zahtjeva moguće je formirati matricu ovisnosti zahtjeva (engl.: *requirements dependency matrix*). Matrica prikazuje zahtjeve sortirane po redcima i stupcima. Gornji dijagonalni dio matrice, uključujući dijagonalu, se ne ispunjava. Donji dijagonalni dio matrice se ispunjava na način da se u polja zapisuje neispravan odnos onih zahtjeva ukoliko postoji, kao na slici 5.2. Prema slici 5.2, vidljivo je da su zahtjevi Z2 i Z1 te Z4 i Z3 u konfliktu, dok se Z4 i Z2 preklapaju. Konfliktni zahtjevi moraju biti diskutirani s korisnicima i preformulirani kako bi se izbjegla konfliktnost. Oni koji se preklapaju također trebaju biti preispitani i reducirani.

Zahtjev	Z1	Z2	Z3	Z4
Z1				
Z2	konflikt			
Z3				
Z4		preklapanje	konflikt	

*Slika 5.2 Matrica ovisnosti zahtjeva*

Nakon što su konflikti i preklapanja otklonjeni i nakon što je formiran skup zahtjeva, potrebno je provesti analizu rizika i određivanje prioriteta. Analiza rizika identificira zahtjeve koji su teško izvedivi, dok je određivanje prioriteta nešto što je potrebno kako bi se postigao veći učinak projekta kada se uračunaju kašnjenja. Tipične vrste rizika su:

- tehnički rizik uzrokovan implementacijom tehničkih rješenja,
- poslovni rizik uzrokovan negativnim učincima prilikom odaziva cijelog sustava na implementirana informatička rješenja,
- rizik sigurnosti uzrokovan implementacijom zahtjevâ kojima se smanjuje sigurnost sustava,
- rizik integriteta baze podataka koji se javlja kada je konzistentnost podataka u zahtjevima teško validirati,
- rizik razvojnog procesa kada zahtjevi traže korištenje nekonvencionalnih razvojnih metoda,

Prioritete razvoja prvenstveno utvrđuju korisnici. Nakon dodavanja faktora rizika, uz dogovor s korisnicima se ponovno revidiraju prioritete.

Zahtjevi trebaju biti povezani, stoga je njima potrebno upravljati. Upravljanje zahtjevima sastavni je dio upravljanja projektima te može biti podijeljeno na tri glavne faze:

- identifikacija, klasifikacija, organiziranje i dokumentiranje zahtjeva,
- promjena zahtjeva,
- utvrđivanje slijeda zahtjeva.

Korisničko prikupljanje zahtjeva često započinje običnim opisivanjem. Na taj način gomilaju se velike količine opisa koje je potrebno formalizirati u konkretne dokumente. Kako bi se postigla preglednost i uspješno upravljalo takvim dokumentima, potrebno ih je numerirati nekakvom identifikacijskom shemom. Identifikacijska shema može sadržavati klasifikaciju dokumenata u skup povezanih dokumenata. Postoji nekoliko tehnika identifikacije i klasifikacije dokumenata.

- Jedinstveni identifikator. To je obično dodijeljeni redosljedni broj obrađenog dokumenta. Dodjeljivanje identifikatora može biti generirano automatski, pomoću nekog alata CASE, ili jednostavno ručno.
- Redosljedni broj s hijerarhijom dokumenata. Dobiva se razmatranjem pozicije dokumenta (npr. poglavlja i odjeljci u knjigama).
- Redosljedni broj s redosljednom kategorijom, koja se dobije s brojem u nekoj od kategorija zahtjeva. Zahtjevi prema kategoriji mogu biti funkcijski, podatkovni, sigurnosni itd.

Tehnika identificiranja s najmanje pogrešaka je svakako automatski generator identifikatora baze podataka u koju se unose dokumenti.

Zahtjevi mogu biti strukturirani po sistemu roditelj-dijete, koji je sličan hijerarhijskoj strukturi. Zahtjevi koji se nalaze na višoj razini strukture (roditelj) sastavljeni su od zahtjeva na nižoj razini (djeca), tj. zahtjev na nižoj razini predstavlja podzahtjev zahtjeva na višoj razini. To znači da je za ispunjenje više razine potrebno prvo realizirati zahtjeve na nižoj razini. Na taj način se hijerarhijskim povezivanjem uvode dodatni nivoi klasifikacije zahtjeva. Iako se hijerarhijsko povezivanje može koristiti i prilikom numeracije zahtjeva, ne znači da je i neophodno.

Hijerarhija zahtjeva omogućuje viši nivo apstrakcije zahtjeva. Na taj način moguće je krenuti od aktivnosti koja je definirana planom i koja predstavlja visoku apstrakciju zahtjeva i spustiti se do najnižih razina apstrakcija. To znači da je moguće spustiti se na razinu konkretnih koraka projekta koje je potrebno realizirati kako bi se postigao

planirani cilj (koji je, prema strateškoj mapi, rezultat promatrane aktivnosti na apstraktnoj razini).

Zahtjevi se tijekom razvojnog ciklusa mogu mijenjati, izbacivati ili zamijeniti novima – zbog toga je neophodno upravljanje promjenama zahtjeva. Promjene mogu biti uzrokovane ljudskom pogreškom, ali često se rade zbog promjene internih pravila poslovanja. Bez obzira na razlog, upravljanje promjenama nalaže da je svaki zahtjev za promjenom potrebno dokumentirati. Budući da je promjena zahtjeva trošak, potrebno je napraviti poslovni slučaj promijenjenog zahtjeva. Prihvaćena promjena zahtjeva, koja je procijenjena kao tehnički izvediva, ima utjecaj na ostatak projekta. Jednom prihvaćena promjena se uvodi u projekt i implementira u softver.

### 5.2 Osnove objektno orijentiranog pristupa

Osnovni koncept objektno orijentiranog pristupa polazi od načina kako se stvari događaju. Uzevši to u obzir, njegova primjena olakšava komunikaciju među dionicima razvoja IS-a i sami način njegove implementacije. U početku (1968.) je objektni pristup bio namijenjen za rad djece na osobnim računalima, što ga je činilo izuzetno jednostavnim. Objekti su lakši za razumijevanje korisnicima jer polaze od poslovanja. Primjerice, prilikom razvoja novog IS-a, koristimo termine kao što su narudžbenica, bankovni račun i sl. Na taj način se puno lakše uočavaju i razumiju moguća unapređenja u svrhu ostvarenja strateškim planom postavljenih ciljeva. Nadalje, podatci i operacije ne razvijaju se i pohranjuju odvojeno, kao što je slučaj kod strukturnog koncepta. Kod objektno orijentiranog koncepta podatci i operacije se drže u lako upravljivim paketima. Na taj način smanjuje se kompleksnost koda, koji je manje osjetljiv na promjene u zahtjevima korisnika. S druge strane, kôd softvera postaje ponovno upotrebljiv. Objektno orijentirana tehnologija se temelji na čvrstim inženjerskim temeljima, čije elemente nazivamo objektni model razvoja ili objektni model. Objektni model obuhvaća principe **apstrakcije**, **enkapsulacije (učahurivanja)**, **modularnosti**, **hijerarhije**, **tipizacije** i **konzistentnosti**.

Niti jedan od navedenih pojmova nije nov, ali svi objektni modeli primjenom navedenog ostvaruju sinergiju (Booch i dr., 2007.). Prethodno korištene monolitne strukture softvera zamjenjuju se dijelovima, što omogućuje njihovo ponovno korištenje u rješavanju novih postavljenih zahtjeva unapređenja poslovanja. Promjena zahtjeva u tom slučaju, pojednostavljeno rečeno, preslagivanje kockica u novi oblik. Primjerice, jednom programirana tražilica imena kupaca jednostavno se implementira u bilo kojem koraku različitih procedura koje to zahtijevaju, odnosno nije je potrebno ponovno programirati. Ovo svojstvo posebno je interesantno u radu s klasama, o čemu će više riječi biti u nastavku.



### 5.2.1 Pojam objekta

Primjena objektnog pristupa zahtijeva utvrđivanje objekata koji se nalaze unutar nekog sustava. Objekt možemo definirati kao bilo koji samostalan pojam koji je moguće zamisliti i jako jednostavno opisati. Tako objekt može biti kupac, račun, zaposlenik, proizvod itd. Bitno je znati da svaki objekt ima svoje atribute, koji određuju svojstva nekog objekta. Primjerice, objekt *automobil* ima atribute proizvođač, marka automobila, tip automobila, boja, snaga motora, potrošnja goriva itd. Isto tako, objekt *osoba* ima atribute ime, prezime, spol, dob, OIB, datum rođenja itd. Isto tako, objekti imaju i svoj oblik ponašanja, pa se tako automobil može kretati, osobe mogu govoriti, nekakav proizvod može biti lomljiv, račun u banci se može otvoriti ili zatvoriti i sl.

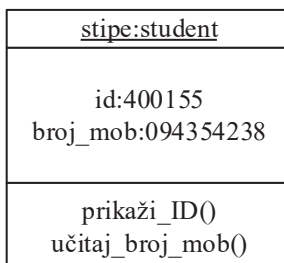
Prema prethodno navedenom, objekti se moraju pronaći u točno određenoj problemskoj domeni koja je uzeta u razmatranje. Bilo koji dio poslovnog svijeta može biti uzet u obzir kao potencijalni objekt. Ipak, katkada se može pojaviti određeni problem jer objekt može imati isto ime, ali se može nalaziti u različitim problemskim domenama. Objekt se može razlikovati prema dva ključna faktora:

- vrsti informacija koje se pridružuju pojedinom objektu,
- namjeni za koju se upotrebljavaju unaprijed uočeni objekti.

Na slici 5.3. može se vidjeti prikaz objekata korištenjem UML-a. U prvi okvir (redak) se upisuju nazivi objekata, dok se u drugi upisuju nazivi atributa. U trećem okviru se upisuju operacije kojima se definiraju ponašanja objekata – npr. kad će se prikazati *id stipe*, koji je student. Zagrada se nastoji opisati da je za izvršenje neke operacije potreban određeni parametar, kojim je postavljeno uvjetovano pokretanje same operacije. Potrebno je navesti kako su atributi skriveni unutar objekata te se mogu vidjeti samo korištenjem operacija.

Prema slici 5.3., iz objekta *stipe*, pokretanjem operacije navedene u trećem okviru (npr. `učitaj_broj_mob()`), koriste se vrijednosti atributa drugog okvira (npr. 094354238). Na taj način, koristeći skup različitih podataka (npr. `broj_mob`) i algoritama (npr. `učitavaj_broj_mob()`) koji operiraju tim podatcima, objekt postaje aktivan i može surađivati s drugim objektima.

Procedure koje se odvijaju nazivamo metodama (npr. znati kako učitati broj). Metode se pozivaju na osnovi poruke koja joj je poslana od drugog objekta. Bitno je navesti da je ime poruke i operacije isto. Na temelju parametara koje sadrži sama poruka, operacija će vratiti vrijednost pošiljatelju.



Slika 5.3 Prikaz objekta prema UML notaciji

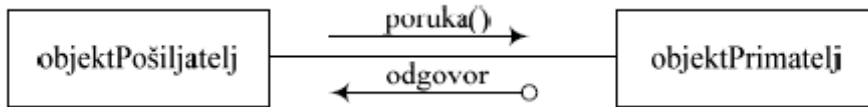
### 5.2.2 Suradnja objekata

Primjerice, da bi se ostvario poziv prema objektu *stipe*, potrebno je napraviti određenu proceduru, a to je unijeti broj mobilnog uređaja te zatim stisnuti tipku za pozivanje. Zbog toga, prema zahtjevu nekog drugog objekta, objekt *stipe* pokreće operaciju „učitaj\_broj\_mob()“, na temelju koje taj drugi objekt pokreće biranje i ostvaruje poziv. Suradnja objekata ostvaruje se slanjem poruka jedno drugom. Poruka poziva operaciju, koja pristupa vrijednosti atributa. Kako bi se to ostvarilo, operacije objekta moraju biti vidljive drugim objektima. Za njih kažemo da su javno vidljive. U skromnim objektno orijentiranim sustavima većina operacija je javno vidljiva, ali većina atributa privatno vidljiva. To znači da objekti jedne klase mogu jedino zahtijevati operaciju za prikaz vrijednosti atributa. Tada kažemo da su operacije učajurile (engl. *encapsulate*) attribute, uz napomenu da se to odnosi na različite klase jer jedan objekt nije u mogućnosti ništa sakriti od drugog objekta u istoj klasi.

To znači da je operacija „učitaj\_broj\_mob()“ vidljiva drugim objektima u istoj klasi. Vidljivost operacija označuje da objekt poznaje operaciju. Tu je bitno razlikovati dva oblika operacija: prvi oblik je operacija koja treba iskoristiti vrijednost unutar objekta i takva se operacija naziva objektnim dosegom (engl. *instance scope*), dok je drugi oblik operacija koja treba izračunati nešto na temelju atributa drugih objekata. U slučaju kad se radi o objektima iste klase, tada se takav oblik naziva klasnim dosegom (engl. *class scope*). Primjerice, ako operacija iščitava vrijednost atributa cijene kave, tada se može govoriti o objektnom dosegu, a ako mora iščitati prosječnu vrijednost svih narudžbi, onda se govori o klasnom dosegu (zato jer treba izračunati srednju vrijednost svih objekata vrsta kave).

Kako smo prije spomenuli, objekti međusobno surađuju. Na taj način ostvaruje se povezivanje objekta u svrhu postizanja jedne složenije funkcionalne cjeline. Jedan objekt jeste cjelina za sebe, ali ne može obaviti sve poslove i operacije. Neke su radnje dodijeljene drugom objektu. Iz tog razloga se u svrhu odrađivanja složenijih poslova, tj. kompletiranja cijelog procesa, ostvaruje suradnja među objektima. Naravno, da bi

objekti surađivali, moraju uspostaviti komunikaciju, a to rade porukama. Na slici se može vidjeti jednostavan prikaz suradnje između dva objekta: objekt pošiljatelj šalje poruku objektu primatelju, koji objektu pošiljatelju natrag šalje odgovor (slika 5.4).



*Slika 5.4 Primjer suradnje među objektima*

Poruke i odgovori u komunikaciji između dva ili više objekata mogu sadržavati jedan ili više parametara, koji se nazivaju argumentima. Parametar je u ovom slučaju potreban kako bi se mogla pokrenuti operacija – i to specifična operacija, na strani samog primatelja poruke. Jasno je da objekt primatelj u sebi može sadržavati više operacija te jedan od parametara može odrediti primatelju koja će se operacija pokrenuti (prisjetite se primjera crtanja likova). Na temelju pokrenute operacije primatelj poruke će moći pročitati te promijeniti vrijednost nekog od svojih atributa.

Na ovako opisanom principu se izvršavaju jako složeni zadatci unutar određenog informacijskog sustava. Važno je znati da više objekata u složenim sustavima može komunicirati s više drugih objekata te je bitno da svaki objekt ima određeni identifikator.

On se objektu dodjeljuje prilikom njegova definiranja, ručno ili automatski, u određenom programskom alatu. Taj identifikator je zapravo jedinstveni broj koji objekt posjeduje tijekom svojeg postojanja, a postoje kratkoročni i dugoročni identifikatori. Naravno, razlika je u njihovom vremenskom trajanju jer neki objekti mogu postojati onoliko koliko je potrebno da se neki radni zadatak izvrši. Tako neki objekti mogu nastati i nestati nakon što se obavi zadatak, dok neki objekti mogu trajati bez obzira na to je li zadatak obavljen ili ne. U tom slučaju služe samo kako bi se zadatak mogao uopće i provesti.

### 5.2.3 Klasa

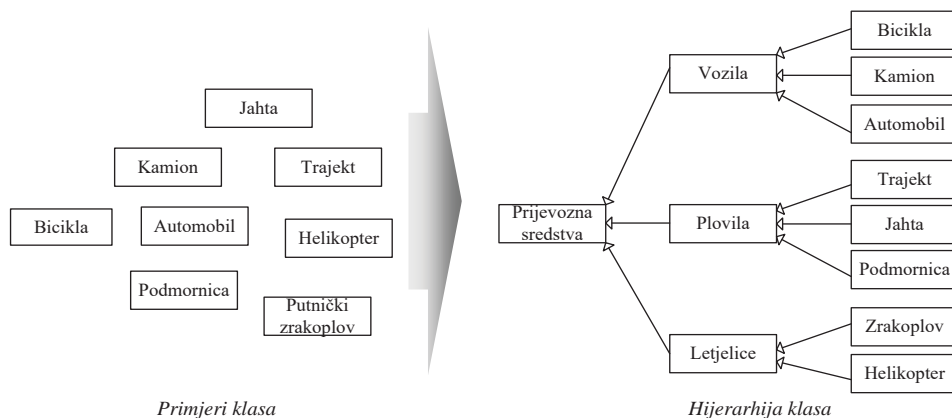
Za razliku od objekata, klasa je skup objekata koji u sebi imaju jednake atribute i operacije. Klasa se može ukratko opisati kao obrazac ili apstrakcija određene vrste objekta. Ovo nam govori da se objekt može kreirati na temelju takvog obrasca u kojem mu se mogu dodijeliti vrijednosti atributa koje odgovaraju tipu atributa definiranih na razini klase. Na takav način svaki objekt može pozvati operaciju, koja je isto definirana klasom kojoj objekt pripada.

Analogno odnosu klase i objekta moguće je definirati i atribute kod klase. Atribut je određen tipom i vrijednošću. Klasom je definiran tip vrijednosti, dok je vrijednošću određen konkretan objekt. Tip atributa može biti primitivni ili druga klasa.

Klasifikacija objekata se vrši na osnovi nekih zajedničkih svojstava. Objektno orijentirani pristup podrazumijeva kako je taj postupak potpuno prirodan. S programerskog stajališta, to omogućuje dijeljenje definicija elemenata među povezanim objektima, zbog čega ih nije potrebno ponavljati. Na slici 5.5. prikazani su primjeri klasa koji se potom prevode u hijerarhiju.

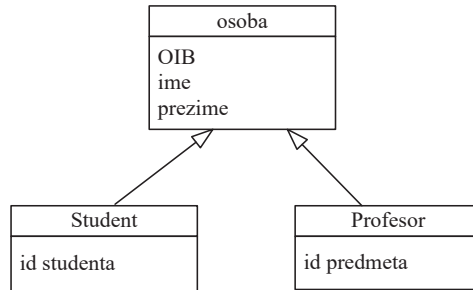
Na taj način dobivaju se više generičke klase (natklase) i više specijalizirane klase (potklase). Kao i kod svakog stabla u hijerarhijskim terminima, postoje klase roditelji i klase djeca. Potklasa je vrsta natklase. Veza između potklasa i natklasa naziva se generalizacija. Generalizacijom se postiže izostavljanje nepotrebnog ponavljanja definiranih svojstava.

Atributi i operacije koji su definirani u natklasama mogu biti ponovno iskorišteni. To znači da potklase nasljeđuju atribute i operacije od nadređene klase.



Slika 5.5 Primjer klasa i hijerarhije klasa

Tako npr. student i profesor predstavljaju potklase klase osoba. Na taj način nasljeđuju atribute klase osoba (OIB, ime i prezime i sl.) te su upotpunjeni s dodatnim atributima. Navedeno je prikazano u nastavku (slika 5.6).



Slika 5.6 Prikaz potklasa

Na ovaj način potklasa može naslijediti dio ili svu strukturu i ponašanje druge klase. Potklasa može nadopuniti strukturu i ponašanje koje nasljeđuje. Na taj način može zamijeniti ili izmijeniti naslijeđeno ponašanje, ali ne i njezinu strukturu. Pretpostavimo, nadalje, da su zadane tri vrste klase: *pravokutnik*, *krug* i *trokut*. Neka spomenute tri klase pripadaju natklasi *oblik*. Natklasa *oblik* predstavlja zajedničke osobine (atributi, npr. boja) i radnje (operacije, npr. promjena boje) svih elemenata sadržanih u navedene tri potklase. Radnja (metoda) *promjene boje*, definirana na razini klase *oblik*, sastoji se od *promjene vrijednosti boje* instance i ponovnog iscrtavanja oblika u novoj boji. Nakon promjene boje (atributa natklase *oblik*), pojednostavljeno rečeno, radnja crtanja (metoda natklase *oblik*) šalje poruku objektu u jednoj od klasa (*pravokutnik*, *krug* ili *trokut*), čime poziva objekt (instancu klase) da nacrtam sebe jer svaka potklasa zna kako se to radi.

Drugim riječima, metoda promjene boje može biti pozvana za bilo koji oblik. Međutim, svaki oblik se crta drukčije. Zbog toga metoda za crtanje šalje poruku objektu (poopćeno klasi), koji u sebi ima integriranu metodu za crtanje upravo onih objekata koji se nalaze u pripadajućoj klasi. To znači da za različite vrijednosti ista metoda može različito reagirati.

Ovakvo svojstvo zovemo polimorfizam ili jednostavno *moć poprimanja više oblika*. Prema prethodnom vidljivo je da se polimorfizam kontrolira nasljeđem. To znači da varijabla, koja u sebi može sadržavati poziv na objekt natklase, može sadržavati i poziv na objekt koji pripada bilo kojoj od njezinih potklasa.

Definirane klase mogu se koristiti u različitim dijelovima sustava. Tako se npr. jednom definirana klasa kupac u podsustavu prodaje, sa svim svojim atributima i operacijama, može koristiti i u podsustavu marketinga. Ovakva vrsta ponovnog korištenja je osnova objektno orijentiranog pristupa.

### 5.2.4 Povezivanje dijelova u cjelinu

Osnove objektnog modeliranja prilikom modeliranja zahtjeva su neophodne jer se povezivanjem prethodno spomenutih objekata dobiva sustav kojim se ostvaruje neka funkcija (cilj promatranog procesa). Niti jedan objekt ne funkcionira sam za sebe. Svi su oni povezani s drugim objektima, a povezivanje objekata omogućuje navigaciju sustavom i utvrđivanje dodatnih informacija i ponašanja. Kod objektno orijentiranog modeliranja postoje dva tipa povezivanja objekata, odnosno dva tipa veza ili relacija: (1) asocijacija i (2) agregacija.

Iako je često teško uočiti razliku između ta dva tipa veze, postoji osnovna ideja prema kojoj asocijacija predstavlja slabi tip veze, dok agregacija predstavlja jaki tip veze. Kod slabog tipa veze (asocijacija) objekt može biti dio neke familije objekata, ali je potpuno neovisan o njoj. Kao primjer asocijacije možemo promotriti putnika koji se vozi u autobusu. On je u slaboj vezi sa svim ostalim putnicima jer se svi voze u istom autobusu i idu u istom smjeru. Međutim, kada neki putnik izađe iz autobusa, ta slaba veza se prekida. Asocijaciju je moguće shvatiti kao *pridruživanje*. S druge strane, jaki tip veze (agregacija) znači udruživanje objekata kako bi se stvorio veći objekt. Kao i svaka druga knjiga, tako i ova sadrži poglavlja, a ona se sastoje od odjeljaka i paragrafa. To znači da su objekti knjiga, poglavlje i odjeljak u jakoj vezi (agregacija). Kao što se može vidjeti, razlika između agregacije i asocijacije je vrlo osjetljive prirode prilikom modeliranja sustava.

Osnove objektnog modeliranja poslužit će kao temelj i za razumijevanje modeliranja podataka (poglavlje 7), u kojem će biti prikazani modeli uz pomoć entiteta i veza. Objektno orijentirani model podataka je sličan modelu entiteti-veze, o čemu će više govora biti u sljedećim poglavljima (poglavlje [7.4](#)).

Prikupljanje zahtjeva i analiza se rade po istim principima. Međutim, objektni model je bogatiji jer sadrži operacije (metode) pomoću kojih se opisuje procesna komponenta informacijskog sustava.

## 5.3 Unificirani jezik modeliranja

### 5.3.1 Uvod u UML 2

OMG (engl. *Object Management Group*) je računalni konzorcij koji razvija standarde za različiti spektar tehnologija (OMG, 2019). Od najpopularnijih standarda koji se tiču sadržaja ove knjige, ističu se (1) BPMN i (2) UML. Nastavno na prethodno spomenuti objektni pristup, važno je posebno istaknuti UML, koji je zapravo kombinacija različitih objektno orijentiranih notacija. UML ima za cilj arhitektima sustava, programerima i inženjerima osigurati alat za analizu, dizajn i implementaciju softvera, kao i alat za modeliranje poslovnih, ali i drugih procesa. Inicijalna verzija UML 1 je

izvorno sadržavala tri vodeće objektno orijentirane metode (Booch, OMT, i OOSE), a uključivala je primjere najboljih praksi objektno orijentiranog programiranja, dizajna jezika modeliranja te deskriptivnih jezika arhitekture. U odnosu na UML 1, ova revizija UML-a (2.5.1) iz 2017. poboljšana je znatno preciznijim definicijama njegovih apstraktnih pravila sintakse i semantike, modularnijom jezičnom strukturom i znatno poboljšanom sposobnošću za modeliranje sustava velikih razmjera.<sup>1</sup>

Dakle, UML je standardizirani jezik za *specifikaciju, vizualizaciju, oblikovanje arhitekture, izradu, simulaciju i testiranje te dokumentiranje modela*. Standardiziranim notacijama (označavanjem) omogućuje jednoznačnu komunikaciju među programerima, projektantima, analitičarima sustava i korisnicima te ostalim dionicima razvoja IS-a. Kao takav, vrlo je važan u razvoju IS-a.

Svojim grafičkim tehnikama i različitim pogledima na sustav omogućuje poslovnom analitičaru cjeloviti pogled na sustav – kako na njegovu strukturu, tako i na ponašanje sustava. Kao i svaki proizvod i UML ima svoj razvoj, koji je ovdje numeriran verzijama. Mi ćemo se bazirati na verziji 2.5. UML specifikacija definira dvije glavne vrste UML dijagrama: (1) strukturni dijagrami (dijagrami strukture) i (2) dijagrami ponašanja.

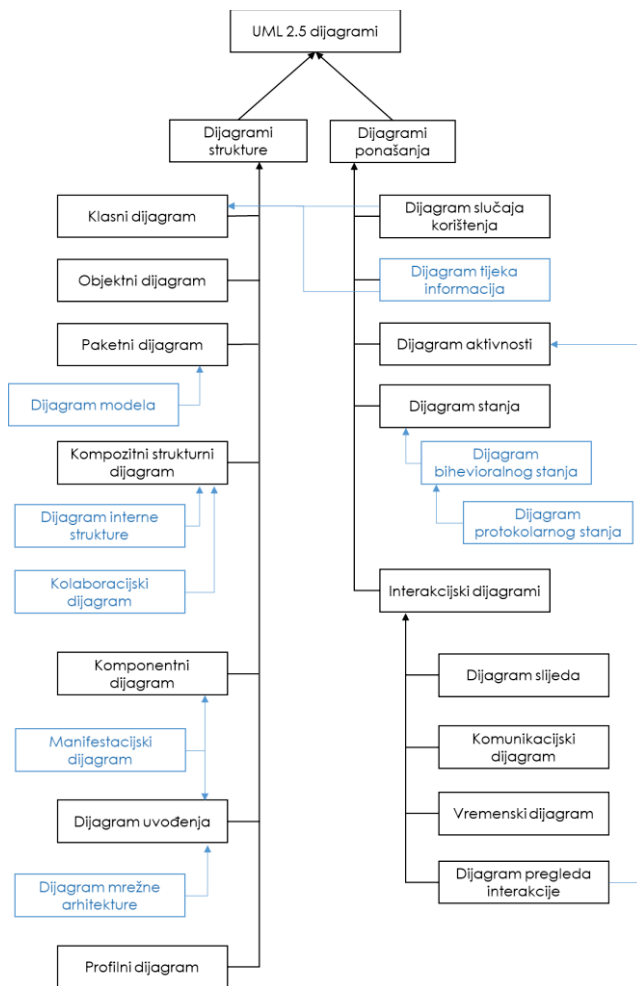
Strukturni dijagrami prikazuju statičku strukturu sustava i njegovih dijelova na različitim razinama apstrakcije i implementacije, kao i njihovu međusobnu povezanost. Apstraktni pojmovi izvode se kao i pojmovi objekt-klasa, o čemu je bilo govora u prethodnim poglavljima. Na taj način elementi u dijagramu strukture predstavljaju smislene pojmove sustava, a mogu uključivati apstraktne, realne i implementacijske koncepte.

Dijagrami ponašanja pokazuju dinamičko ponašanje objekata u sustavu, tj. opis niza promjena u sustavu, bez obzira radi li se o načinu korištenja, slijedu događaja unutar promatranog procesa ili sl.

Dijagrami UML 2.5 mogu se kategorizirati hijerarhijski, kao što je prikazano na slici 5.7 u nastavku (UML-Diagrams, 2018). Napomena – stavke prikazane plavom bojom nisu dio službene taksonomije dijagrama UML 2.5.

---

<sup>1</sup> Više detalja može se pronaći na <https://www.omg.org/spec/UML/2.5.1/PDF>



Slika 5.7 UML dijagrami(UML-Diagrams, 2018.)

Sukladno navedenom izvoru, na slici 5.7. su navedeni hrvatski nazivi svakog od dijagrama, koje ćemo dalje koristiti u ovoj knjizi. Normirani UML ima 14 dijagrama (notacija), a sve ono što je ovdje označeno plavom bojom su vrste problema koji se mogu rješavati pomoću nekog dijagrama.

### 5.3.2 Osnovni dijagrami UML-a

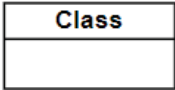
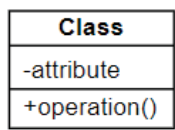


U sljedećem tekstu objasniti ćemo ulogu svakog od nama relevantnih dijagrama. Neke dijagrame nećemo objašnjavati jer zahtijevaju više informatičkog i programerskog znanja nego što se očekuje od čitatelja ove knjige. Krenut ćemo sa strukturnim dijagramima te nastaviti s dijagramima ponašanja. Nastavno na prethodno poglavlje, počinjemo s klasnim dijagramom.



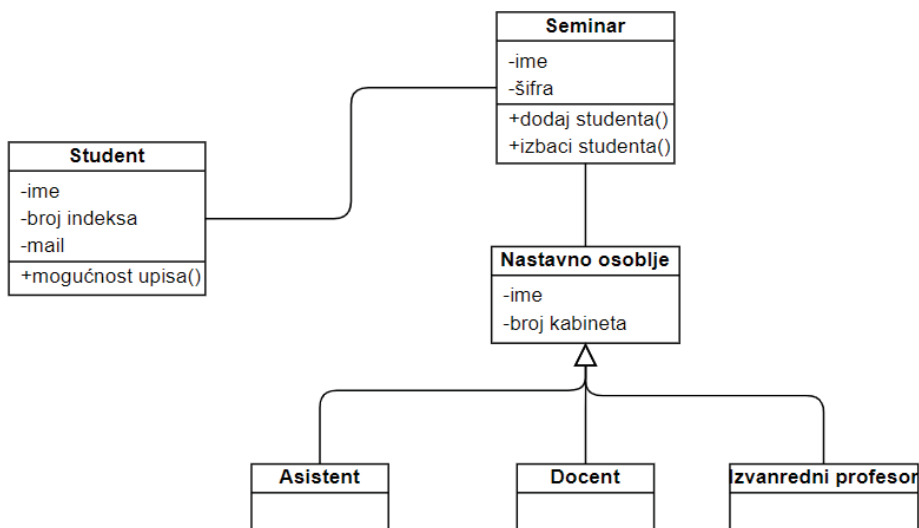
### 5.3.2.1 Klasni dijagram (engl. class diagram)

Prethodno je pojašnjeno što su to klase i objekti, a u nastavku će biti opisan način na koji se navedeni pojmovi označuju u dijagramu. Klase se najčešće crtaju u dijagramu klasa, kako navodi UML. Kako bi se razlikovalo crtanje klasa i objekata, pravilo je da se nazivi klasa ne podcrtavaju, a nazivi objekata podcrtavaju. Također, nazivi klasa započinju velikim slovom, a nazivi objekata malim slovom te se zahtijeva da nazivi objekata budu iznimno kratki te napisani u jednini. Kod klase je isto tako moguće definirati i atribute, a oni su određeni tipom i vrijednošću. Dok je kod klase definiran tip vrijednosti, samoj vrijednosti je definiran određeni objekt. Notacija koja se koristi u dijagramu klasa prikazana je u tablici 5.1.

Tablica 5.1 Notacija dijagrama klasa

Notacija	Značenje
	Klasa (engl. <i>class</i> )
	Klasa s prikazanim atributima i operacijama
	Veza generalizacije
	Veza asocijacije

Na slici 5.8 je primjer klasnog dijagrama na kojem imamo sljedeće klase: *Student*, *Seminar*, *Nastavno osoblje*, *Asistent*, *Docent* i *Izvanredni profesor*. Primjere atributa navest ćemo na primjeru klase *Student*, koja ima četiri atributa: ime, mail, broj indeksa te prosječna ocjena. Klasa *Student* ima operaciju *mogućnost upisa*, budući da svaki student može upisati *Seminar*. *Seminar* je druga klasa, koja je povezana s klasom *Student* tipom veze *asocijacija*. Još jedan tip veze koji je vidljiv na slici je veza *generalizacije*. To je veza među (nad)klasom *Nastavno osoblje* i klasama *Asistent*, *Docent* te *Izvanredni profesor*.



Slika 5.8 Primjer klasnog dijagrama (izrađeno pomoću alata Draw.io)

### 5.3.2.2 Objektne dijagram (engl. object diagram)

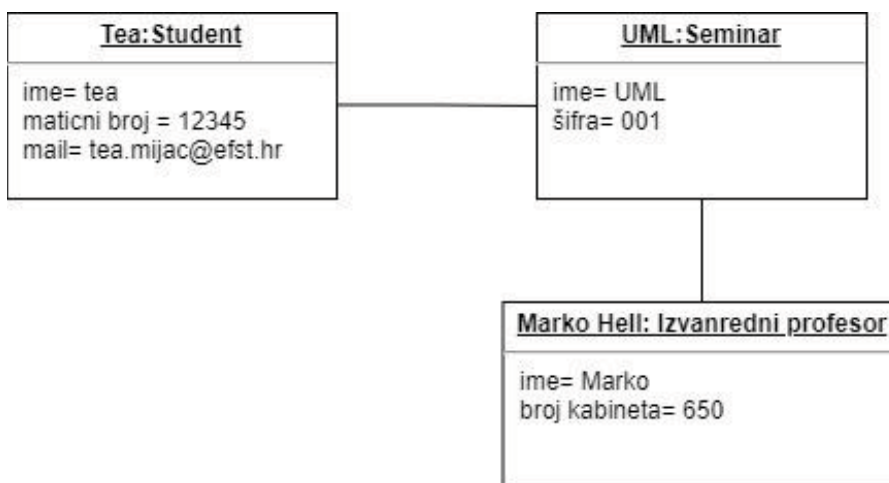
Objektni dijagram prikazuje skup objekata i njihove veze (relacije) u sustavu. Objektne dijagrami predstavljaju statičku snimku stvari koje se nalaze u dijagramu klase. Oni se bave statičkim pogledom na dizajn i procese sustava, kao i dijagrami klase, ali iz perspektive pravih prototipova, tj. pravih objekata u sustavu. Notacija koja se koristi u objektnom dijagramu prikazana je u tablici 5.2.

Tablica 5.2 Notacija objektnog dijagrama

Notacija	Značenje
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p><b>Object: Type</b></p> <p>field1 = value1 field2 = value2 field3 = value3</p> </div>	Objekt
<div style="border: 1px solid black; width: 100px; height: 10px; margin: 0 auto;"></div>	Veza asocijacije

Objektni dijagrami nemaju važnost kao što je imaju dijagrami klase, ali su korisni za pojašnjavanje kompleksnih dijagrama klase, prvenstveno prikazujući mogući primjer dijagrama klase i prikaz suradnje između skupova objekata. Na slici 5.9 prikazan je

primjer objektnog dijagrama. Ako uzmemo u obzir klasni dijagram koji smo prikazali na slici 5.8, vidimo da je objektni dijagram jako sličan klasnom, s tim da atributi klase sada poprimaju vrijednosti. Primjerice, klasa *Student* sada postaje *Student: Tea*. Atribut *ime* je poprimio vrijednost *Tea*, atribut *mail* je poprimio vrijednost: *tea.mijac@efst.hr*, a matični broj je poprimio vrijednost *12345*. Kako je klasa *Student* u vezi s klasom *Seminar*, tako i student *Tea* ovdje upisuje seminar pod nazivom *UML*. Atributi klase *seminar*, kada je riječ o objektu, sadrže attribute koji poprimaju vrijednosti, pa tako atribut *ime* poprima naziv *UML*, a atribut *šifra* poprima vrijednost *001*. Isto je i s trećim objektom.




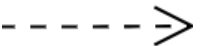
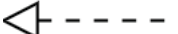


Slika 5.9 Primjer objektnog dijagrama (alat Visual Paradigm)

### 5.3.2.3 Paketni dijagram (engl. package diagram)

Paketi, odnosno paketni dijagrami, su kolekcije logički povezanih elemenata UML-a. Omogućuju organiziranje elemenata modela u grupe, čineći tako dijagrame UML-a jednostavnijima i lakšima za razumijevanje.

Paketni dijagram organizira elemente sustava u povezane grupe kako bi se smanjila ovisnost među njima. Ovisnost definira odnos u kojem će promjene jednog paketa utjecati na drugi paket. Uvoz je oblik ovisnosti koja jednom paketu dopušta pristup sadržaju drugog paketa. Notacija koja se koristi u objektnom dijagramu prikazana je u tablici 5.3.

Tablica 5.3 Notacija paketnog dijagrama

Notacija	Značenje
	Paket
	Ovisnost (dviije su vrste ovisnosti – ulaz i pristup)
	Realizacija
	Sadržanost
	Generalizacija

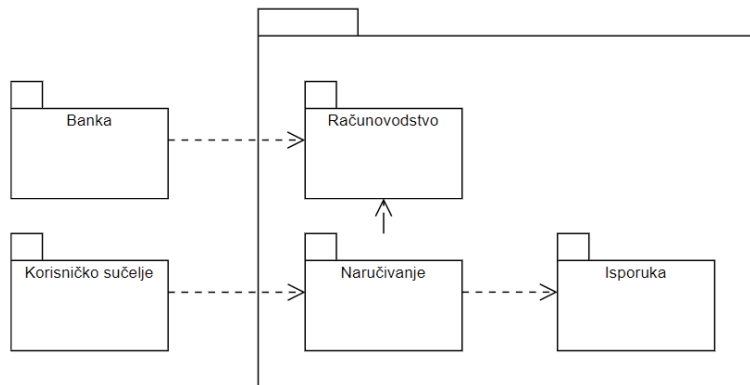
Svrha paketa je omogućiti hijerarhijsku organizaciju elemenata u UML dijagramu. Paketi su prikazani kao datoteke i mogu se koristiti na bilo kojem UML dijagramu, iako su najčešće korišteni na dijagramima slučajeva korištenja i klasnim dijagramima, budući da ti modeli imaju tendenciju rasta. Paket može sadržavati druge pakete, objekte, razrede, komponente itd.

Markeri vidljivosti označuju tko može pristupiti informacijama koje su sadržane unutar paketa:

- - *private* – atribut ili operacija nisu dostupni ni na što izvan paketa,
- + *public* – atribut ili operacije mogu pregledati i drugi paketi,
- # *protected* – atribut ili operacija vidljivi su samo paketima koji se nasljeđuju.

Jedan od tipičnih primjera paketnog dijagrama prikazan je na slici 5.10 u nastavku. Dijagram sadrži pet paketa: *Banka*, *Korisničko sučelje*, *Računovodstvo*, *Naručivanje* i *Isporuka*. Iz dijagrama se može vidjeti da su *Računovodstvo*, *Naručivanje* i *Isporuka* grupirani odvojeno od *Banke* i *Korisničkog sučelja*.

Primjerice, veze ovisnosti omogućuju da *Naručivanje* pristupi sadržaju koji se nalazi u *Korisničkom sučelju*, a da onda paket *Isporuke* pristupi sadržaju *Naručivanja*.


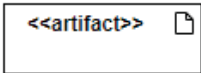
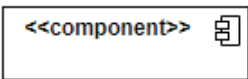



Slika 5.10 Primjer paketnog dijagrama (Lucidchart)

#### 5.3.2.4 Dijagram uvođenja (engl. deployment diagram)

Dijagram raspodjele pokazuje na kojoj komponenti IT-ja se nalazi i/ili izvršava koja softverska komponenta. On statičkim pogledom na pokretanje arhitekture sustava prikazuje konfiguraciju izvršnih procesnih čvorova i komponenata koje se izvode na njima. Povezani su s dijagramima komponenata tako da čvor tipično obuhvaća jednu ili više komponenata. Dijagram uvođenja je zapravo graf čvorova povezanih komunikacijom. Notacija koja se koristi u objektnom dijagramu prikazana je u tablici 5.4.

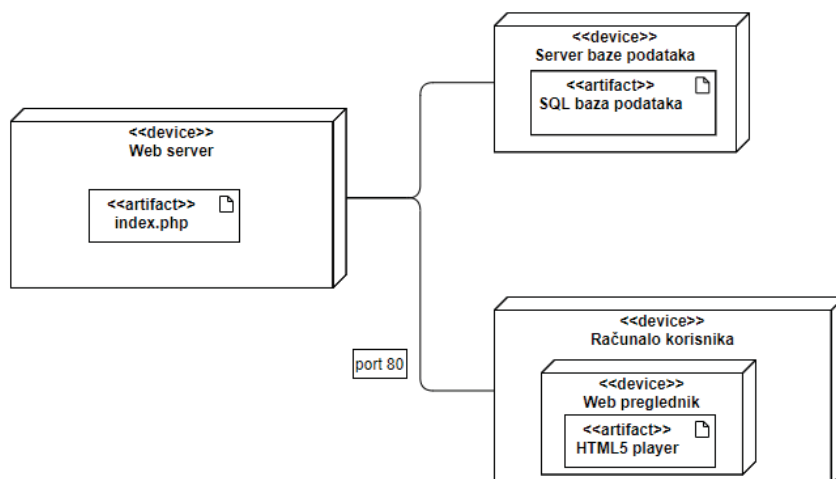
Tablica 5.4 Notacija dijagrama uvođenja

Notacija	Značenje
	Uređaj (engl. <i>device</i> )
	Artefakt (engl. <i>artifact</i> )
	Komponenta (engl. <i>component</i> )
	Sučelje (engl. <i>interface</i> )

	Ovisnost
	Kompozicija
	Korištenje (engl. <i>use</i> )
	Agregacija

Čvor se definira kao fizički objekt koji prikazuje izvor procesa, a ima svoju memoriju i sposobnost procesiranja. Prikazuje uređaje, ljudske izvore i mehaničko procesiranje (alokacijom izvršnih komponenti i objekata prikazuje se koja se programska jedinica izvodi na kojem uređaju).

Dijagram uvođenja prikazuje fizičku arhitekturu elemenata procesa u realnom vremenu. Komponente koje ne postoje u realnom vremenu ne prikazuju se na dijagramu komponenata. Na slici u nastavku (5.11) je prikazan dijagram uvođenja. *Web server*, *Server baze podataka* i *Računalo korisnika* predstavljaju okoline u kojima se izvršavaju određeni artefakti, primjerice *index.php*, *HTML5* ili baza podataka *SQL*. Veza između te dvije okoline predstavlja put komunikacije – protokol (npr. port 80).






Slika 5.11 Dijagram uvođenja (internetski alat Lucidchart)

### 5.3.2.5 Kompozitni strukturni dijagram (engl. composite structure diagram)

Strukturni dijagrami služe, kao što im to samo ime kaže, za predstavljanje strukture, i to kod prikaza arhitekture sustava. Koriste se kako bi se modelirale „stvari“ od kojih se model sastoji, skupa s fizičkim elementima (Systems, 2007).

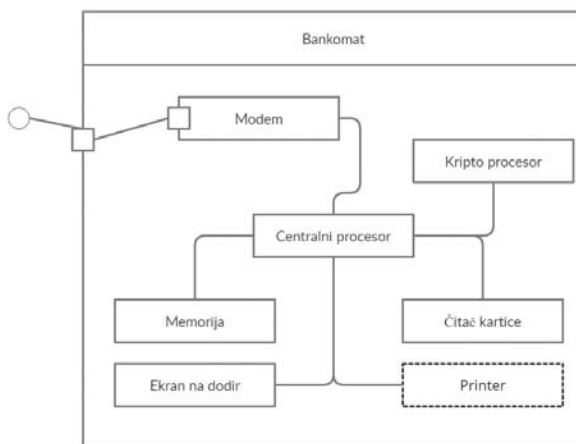
Notacija koja se koristi u objektnom dijagramu prikazana je u tablici 5.5.

Tablica 5.5 Notacija dijagrama raspodjele

Notacija	Značenje
	Element modela
	Vanjski element modela
	Port (ulaz)

Isto tako, ovi dijagrami se koriste za modeliranje veza i ovisnosti među elementima. Kompozitni strukturni dijagrami koriste se za istraživanje instanci međusobno povezanih slučajeva koji surađuju komunikacijskom vezom, stoga strukturni dijagrami predstavljaju prirodan način prikazivanja i razdvajanja komponenti.

U nastavku (Slika 5.12) je prikazan primjer modela bankomata koji se sastoji od više elemenata (fizičkih). Drugim riječima, sastoji se od centralnog procesora, koji je uz pomoć modema povezan na internet (port), memoriju, kriptoprocessor, čitač kartice, ekran na dodir te printer (koji predstavlja vanjski element modela).


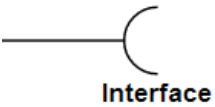
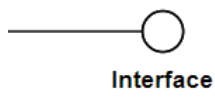



Slika 5.12 Primjer kompozitnog strukturnog dijagrama (izrađeno pomoću internetskog alata Creately)

### 5.3.2.6 Dijagram komponenti (engl. component diagram)

Dijagram komponenti prikazuje organizaciju i ovisnosti između skupova komponenata. Dijagrami komponenti bave se statičkim pogledom na implementaciju sustava i povezani su s dijagramima klasa, tako da su komponente tipično mapirane na jednu ili više klasa, sučelja ili suradnji. Notacija koja se koristi u objektnom dijagramu prikazana je u tablici 5.6.

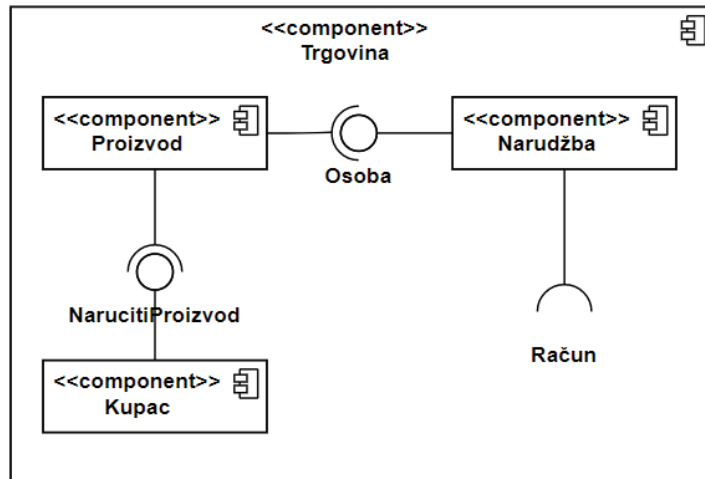
Tablica 5.6 Notacija dijagrama komponenti

Notacija	Značenje
	Komponenta (engl. <i>component</i> )
	Korištenje sučelja (engl. <i>interface</i> )
	Sučelje
	Realizacija



Osnovni cilj dijagrama je prikaz strukturnih veza među komponentama sustava.

Dijagram komponenti prikazan je na slici u nastavku (slika 5.13).

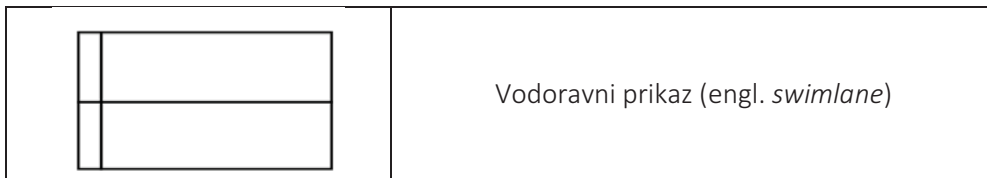


Slika 5.13 Dijagram komponenti (izrađeno pomoću alata Lucidchart)

Dijagram aktivnosti je pojednostavljen prikaz onoga što se događa za vrijeme operacije ili procesa (Booch et al.,2001). Pokazuje korake u operaciji ili procesu, koji su nazvani aktivnosti, ali i točke odluke (engl. *decision points*) i grane (ogranke). Koristi se npr. za prikazivanje poslovnog procesa ili operacije. Dijagram aktivnosti nadopunjuje dijagram stanja, odnosno predstavlja njegov dodatak. Dijagram stanja prikazuje stanje objekta i aktivnosti kao strelice koje povezuju stanje, dok dijagram aktivnosti fokus stavlja baš na aktivnosti. Notacija koja se koristi u objektnom dijagramu prikazana je u tablici 5.7.

Tablica 5.7 Notacija dijagrama aktivnosti

Notacija	Značenje
	Aktivnost (engl. <i>activity</i> )
	Početak
	Kraj

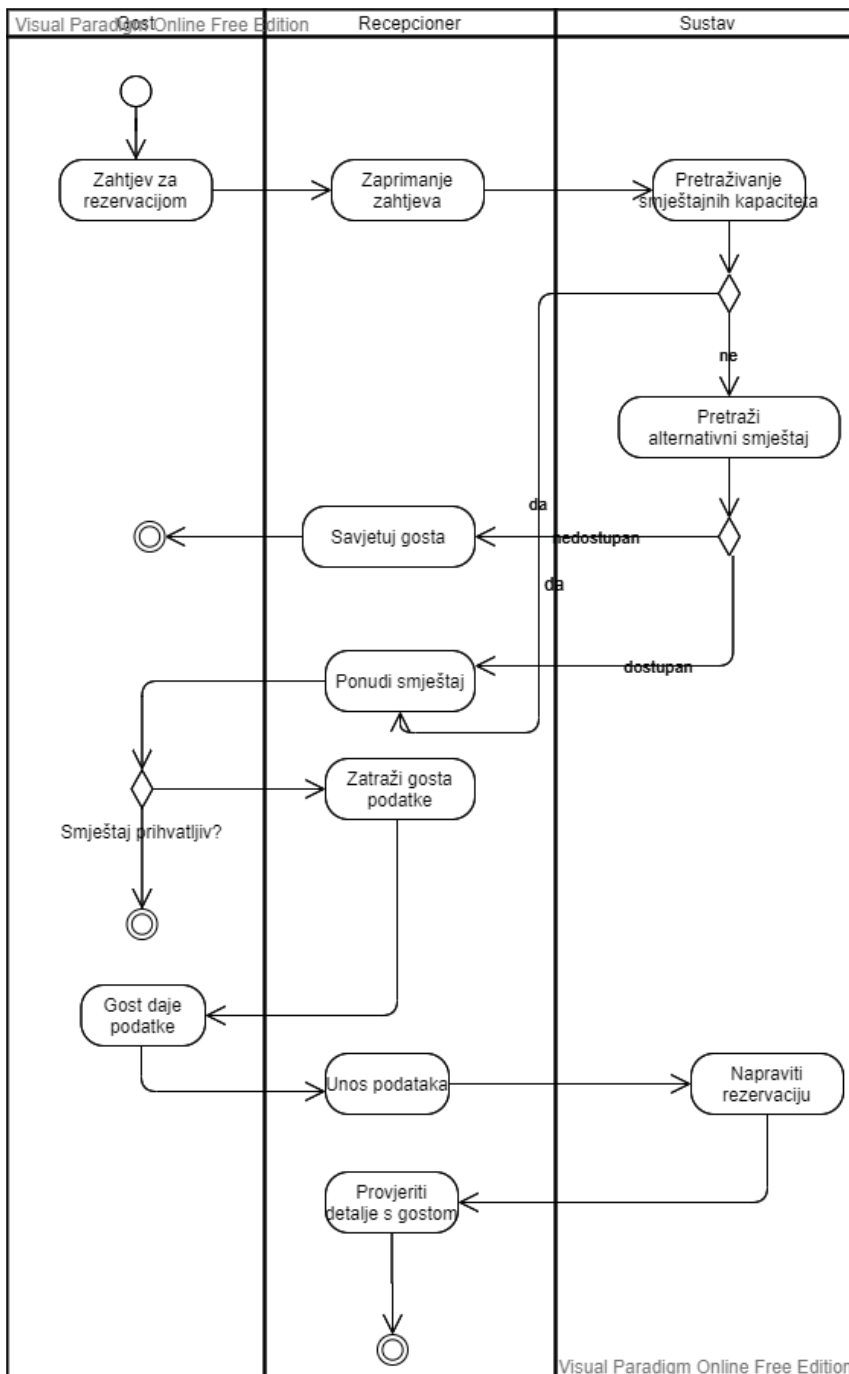


Svaka aktivnost je predstavljena zaobljenim pravokutnikom. Kad se završi odvijanje jedne aktivnosti, automatski se prelazi na odvijanje druge aktivnosti. Strelice predstavljaju kretanje od jedne aktivnosti do sljedeće. Kao i dijagram stanja, dijagram aktivnosti ima početnu točku prikazanu kao "bikovo oko" (engl. *bull's eye*), odnosno kružnice s manjim ispunjenim kružićem unutar njega. Budući da je bitna karakteristika UML-a njegova univerzalnost, pojedine dijagramske tehnike se mogu koristiti u različitim koracima razvoja programskog sustava.

Tako se dijagrami aktivnosti mogu koristiti za prikaz tijeka poslovnih procesa na početku procesa razvoja programskog sustava, ali se mogu koristiti i na nižim razinama (npr. na razini dizajna), gdje se koriste za prikaz tijeka po pojedinim klasama podataka.

Mogu se koristiti i za prikaz tijeka između metoda i klasa objekata. U dijagramima akcija mogu se koristiti i tzv. staze, kojima se označuje pripadnost pojedine aktivnosti određenom objektu. Staze (engl. *swimlane*) grupiraju aktivnosti unutar stupaca koji predstavljaju objekte. Samim time određuje se odgovornost objekta u izvršavanju aktivnosti, tako da je, primjerice, recepcionar odgovoran za zaprimanje zahtjeva, savjetovanje gosta, ponudu smještaja, traženje podataka gosta, unos podataka i provjeru detalja s gostom (Slika 5.14).

Simbol grananja preusmjerava, ovisno o ispunjenju uvjeta, na određenu aktivnost. Ako je traženi smještaj dostupan, recepcionar će ponuditi gostu smještaj, a ako nije, sustav će pretražiti alternativni smještaj. Početna točka se označuje crnim krugom, a završna kružnicom s manjim ispunjenim kružićem unutar nje.



Slika 5.14 Primjer dijagrama aktivnosti

### 5.3.1.1. Dijagram slučaja korištenja (engl. use case diagram)

Dijagram slučaja korištenja je dijagram koji prikazuje skup: (1) slučaja korištenja, (2) korisnika (engl. *actors*) i (3) njihove veze (engl. *relationship*). Pomoću dijagrama slučaja grafički se dokumentiraju prikupljeni zahtjevi. Korisnost ovih dijagrama očituje se u vizualizaciji slučajeva korištenja, čime se pojednostavljuje komunikacija između analitičara i korisnika. U dijagramima slučaja korištenja simbol za jedan slučaj korištenja je elipsa. Simbol sudionika je figura, a veze se označavaju linijama. Cjelokupni skup slučajeva korištenja okružuje se pravokutnikom, koji predstavlja granicu sustava. Kako bi se dijagram slučaja korištenja pojednostavio, prilikom crtanja se koriste još neki tipovi veza.

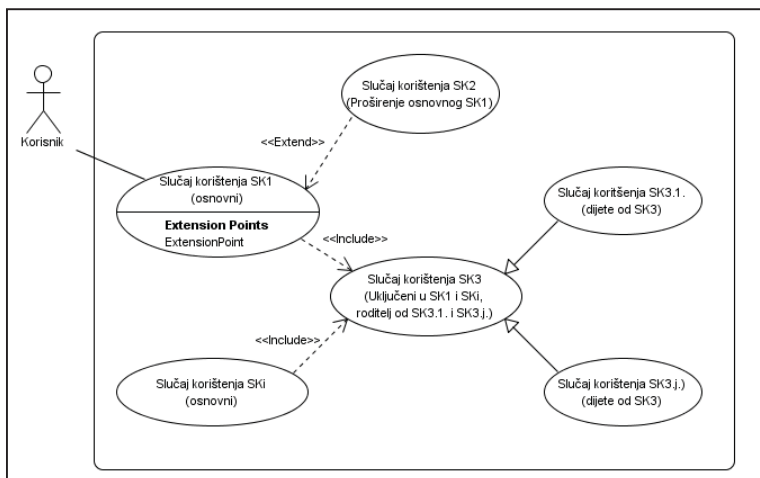
Tipovi veze mogu biti:

- uključenje (engl. *include*),
- proširenje (engl. *extend*),
- generalizacija.

Uključenje je prikazano linijom na kojoj piše *include*. Ovakav tip veze koristi se kada je potrebno modelirati višekratnu funkcionalnost slučaja korištenja. Tako, primjerice, svaki slučaj korištenja koji zahtijeva verifikaciju korisnika može biti povezan vezom uključenja. Na taj način verifikacija korisnika je obilježena kao sastavni dio svakog osnovnog slučaja korištenja koji zahtijeva verifikaciju. Uključeni slučajevi korištenja ne egzistiraju samostalno nego se proširenje prikazuje linijom na kojoj piše *extend*. Veza proširenja označuje da je ponašanje osnovnog slučaja korištenja povezano s ponašanjem drugog slučaja korištenja (svojeg proširenja).

Proširenje se može koristiti i u slučajevima modeliranja uvjetnih tijekova, tj. slučajeva korištenja koji se ostvaruju pod određenim uvjetima. Primjerice, novi korisnik nekog internetskog servisa, za razliku od postojećeg korisnika, mora ispuniti informativni obrazac. Veza generalizacije slučaja korištenja ima isti smisao kao i generalizacija objekata i klasa. Navedeno znači da veza generalizacije označuje nasljeđe značenja i ponašanje drugoga slučaja korištenja.

Generalizacija je prikazana istom linijom koja prikazuje naslijeđenost unutar klase. Primjer dijagrama slučaja korištenja prikazan je na slici 5.15.



Slika 5.15 Primjer dijagrama slučaja korištenja

Korisnik inicijalizira slučaj korištenja. Kao i na slici, elipsa predstavlja slučaj korištenja, a čovječuljak predstavlja korisnika. Naziv slučaja korisnika se nalazi ili unutar elipse ili ispod nje. Linija koja povezuje korisnika sa slučajem korištenja predstavlja komunikaciju među njima. Izrada dijagrama slučaja korištenja započinje intervjuiranjem korisnika u fazi analize. Intervjui s korisnicima rezultiraju dijagramima korištenja slučaja visoke razine, tj. poslovnog slučaja korištenja (engl. *business use case*), koji prikazuju što je sve potrebno za funkcioniranje sustava. Poslovni model slučaja korištenja nije ništa drugo nego poslovni slučaj korištenja, samo na višoj razini apstrakcije. Pomoću poslovnih modela slučaja korištenja identificiraju se poslovni procesi koje je, prema utvrđenim aktivnostima, potrebno unaprijediti. Fokus poslovnih slučajeva korištenja je sama arhitektura poslovnih procesa. Dijagrami trebaju osigurati željeni pogled na sustav, tj. kako bi ponašanje sustava trebalo izgledati (stanje *to be*).

Verbalni opis svakog poslovnog slučaja mora biti kratak, poslovno orijentiran i fokusiran na glavne tijekove, tj. aktivnosti u sustavu. Međutim, verbalni opisi programerima nisu dovoljni za jasno definiranje što sustav treba raditi. Poslovni modeli slučaja u kontekstu analize i specifikacije zahtjeva više su okrenuti poslovanju. U ovoj fazi oni predstavljaju identificirane korisničke slučajeve koji uključuju njihove potprocese i alternativne procese. Sudionici u poslovnim slučajevima korištenja generiraju događaje kojima se pokreću procesi unutar poslovnog slučaja. Komunikacijska linija između sudionika i slučaja korištenja je tijek događaja, odnosno odgovor slučaja korištenja sudioniku. Oni mogu biti i unutar i izvan opsega poslovnog slučaja. Prilikom specifikacije sustava i njegovog okruženja potrebno je opisati o kojim se suradnicima radi.

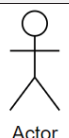
Želimo li kreirati modele slučaja korištenja, potrebno je razraditi sve dijagrame korištenja slučaja visoke razine. Krajnji dijagrami korištenja slučaja su važni za daljnje dizajniranje i za sam razvoj sustava. Jedna od prednosti korištenja dijagrama slučaja korištenja je ta što prikazuje granicu između sustava i vanjskog svijeta. Korisnici su tipično izvan sustava, dok su slučajevi korištenja unutar njega. Iz tog razloga se koristimo pravokutnikom (u kojem se nalazi i ime sustava) kako bismo prikazali granice sustava. Pravokutnik okružuje slučajeve korištenja koji se nalaze unutar granica sustava. Budući da UML nema prikladnu vizualnu grafičku tehniku za prikazivanje opsega sustava, granice sustava se izvode iz dijagrama slučaja korištenja. Međutim, postavlja se pitanje kako utvrditi opseg sustava koji promatramo. Često se događa da prilikom uvažavanja promjena zahtjeva nenamjerno povećavamo opseg posla, dok resursi ostaju ograničeni. Odgovor na ovo pitanje nije jednostavan jer je svaki podsustav dio nekog većeg sustava.


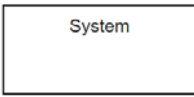
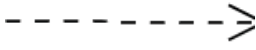

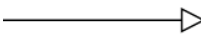
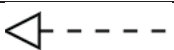
Sustavi komuniciraju razmjenjujući informacije i međusobno pozivajući servise, tako da se prethodno pitanje može preformulirati. Potrebno je zapitati se implementiraju li se zahtjevi u promatrani sustav ili se odgovornost za ostvarenje zahtijevane funkcionalnosti nalazi u nekom drugom sustavu, što znači proširenje opsega? Zato je potrebno promotriti kontekst promatranog sustava. Potrebno je poznavati elemente okruženja: (1) okolne sustave s kojima je promatrani sustav u interakciji, (2) organizaciju, (3) ljudske resurse i ostalu hardversku podršku od koje se očekuje određena informacija itd.

U poslovnim sustavima se prethodni zahtjevi opisuju tijekom podataka. Uočeni sustav dobiva ulazne podatke, na njima obavi potrebne radnje te ih "izbacuje" u obliku izlaznih podataka. Prema tome, opseg sustava može se odrediti na osnovi ulazno/izlaznih tijekomova podataka u promatrani sustav. To znači da bilo koji zahtjev koji ne može biti podržan temeljnim procesom (radnjom nad ulaznim podacima) prelazi opseg sustava i pripada negdje drugdje.

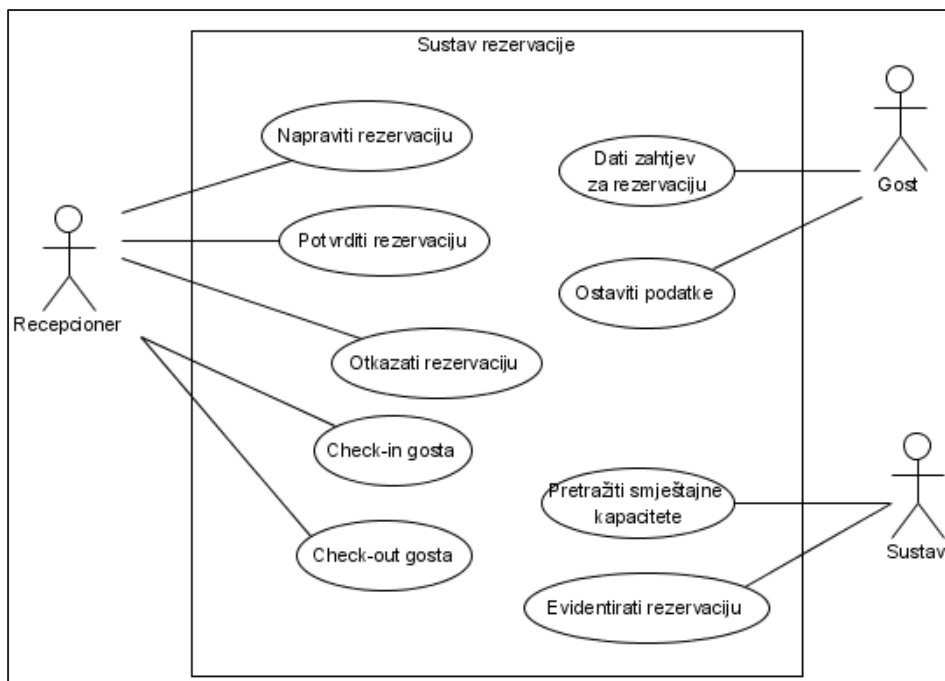
U tom kontekstu dijagram slučaja korištenja je iznimno koristan jer na jednostavan način daje prikaz funkcionalnosti cijelog programskog sustava. Notacija koja se koristi u objektnom dijagramu prikazana je u tablici 5.7.

*Tablica 5.8 Notacija dijagrama slučaja korištenja*

Notacija	Značenje
 <p>Actor</p>	Sudionik

 <p>UseCase</p>	Slučaj korištenja (engl. <i>use case</i> )
 <p>System</p>	Sustav
	Ovisnost
	Asocijacija
	Generalizacije
	Realizacija

Dijagram slučaja korištenja na primjeru sustava rezervacije prikazan je na slici u nastavku (5.16).



Slika 5.16 Slučaj korištenja na primjeru recepcije

5.3.1.2. *Dijagram stanja (engl. state diagram)*

Dijagram stanja služi za opisivanje ponašanja sustava. To pretpostavlja da mora postojati konačan broj stanja. Dijagram stanja modelira događaje koji predstavljaju „okidač“ za prelazak iz jednog stanja u drugo, a promjena se događa unutar klase. Svaki od spomenutih događaja ima pripadajuće aktivnosti koje rade promjene u klasnim instancama.

Dijagram stanja može se koristiti i za modeliranje konkurentnih aktivnosti stanja na način da se kreiraju paralelna *podstanja* (*podstanja* se tada nalaze unutar *superstanja*). Neke od osnovnih točaka su: (1) dijagram stanja modeliran na razini klase, a ne na razini instanci; (2) prikazuju se događaji koji uzrokuju promjenu stanja klase u drugo stanje te iste klase (za razliku od dijagrama slijeda, koji pokazuje interakcije među klasama); (3) opseg dijagrama stanja je čitav životni ciklus pojedine klase; (4) nekoliko dijagrama slijeda je potrebno koristiti kako bi se deriviralo kompletno stanje klase.

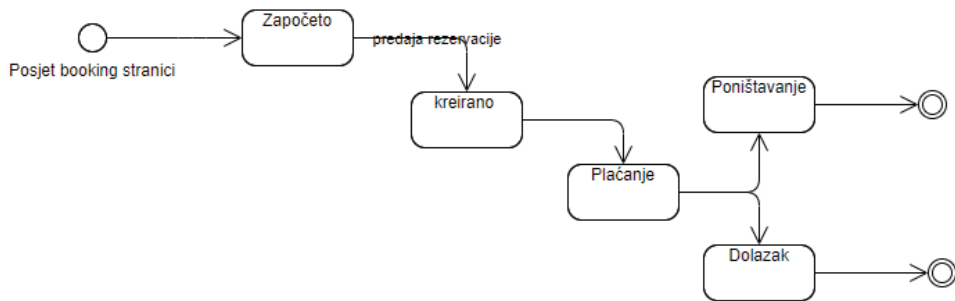
U nastavku je tablično prikazana notacija za dijagram stanja.

Tablica 5.9 Notacija dijagrama stanja

Notacija	Značenje
	Početak
	Kraj
	Stanje (engl. <i>state</i> )
	Tranzicija

Primjer dijagrama stanja u alatu *Visual Paradigm* se nalazi na slici u nastavku.





Slika 5.17 Dijagram stanja

### 5.3.1.3. Dijagram slijeda (engl. sequence diagram)

Dijagrami slijeda prikazuju interakciju među objektima u vremenskom slijedu. Važnost ove vrste dijagrama je u tome što se prikazom interakcije objekata u vremenskom slijedu može relativno lako utvrditi koji objekti i interakcije su potrebni da bi se ostvarila funkcionalnost definirana tijekom događaja. **Objekti** su poslagani od vrha prema dolje, s lijeva nadesno. Uređeni su na način da pojednostave dijagram i poslagani tako da tvore dijagram. U nastavku svakog objekta nalazi se iscrtana linija koja predstavlja vremenski slijed. Uz vremenski slijed nalazi se okomiti pravokutnik nazvan događaj. Objekt može slati poruke sam sebi, tj. putujući od jednog vremenskog slijeda i završavajući na svojem vlastitom vremenskom slijedu.

Poruka može biti jednostavna, usklađena i neusklađena (Martin, 2000.).

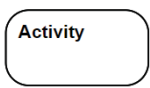


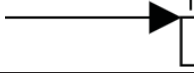
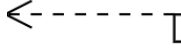
- Jednostavna poruka – prenosivost kontrole s jednog objekta na drugi.
- Usklađena poruka – ako objekt šalje usklađenu poruku, čeka na odgovor te poruke prije nego što nastavi s radom.
- Neusklađena poruka – ako objekt šalje neusklađenu poruku, on ne čeka na odgovor prije nego što nastavi s radom.

Dijagram predstavlja vrijeme u vertikalnom smjeru. Vrijeme započinje na vrhu i teče prema dnu. Poruka koja je bliža vrhu događa se ranije nego poruka koja je bliža dnu. Prema tome, dimenzije dijagrama su sljedeće:

- “lijevo – desno” je osnova objekta
- “vrh – dno” pokazuje protok vremena.

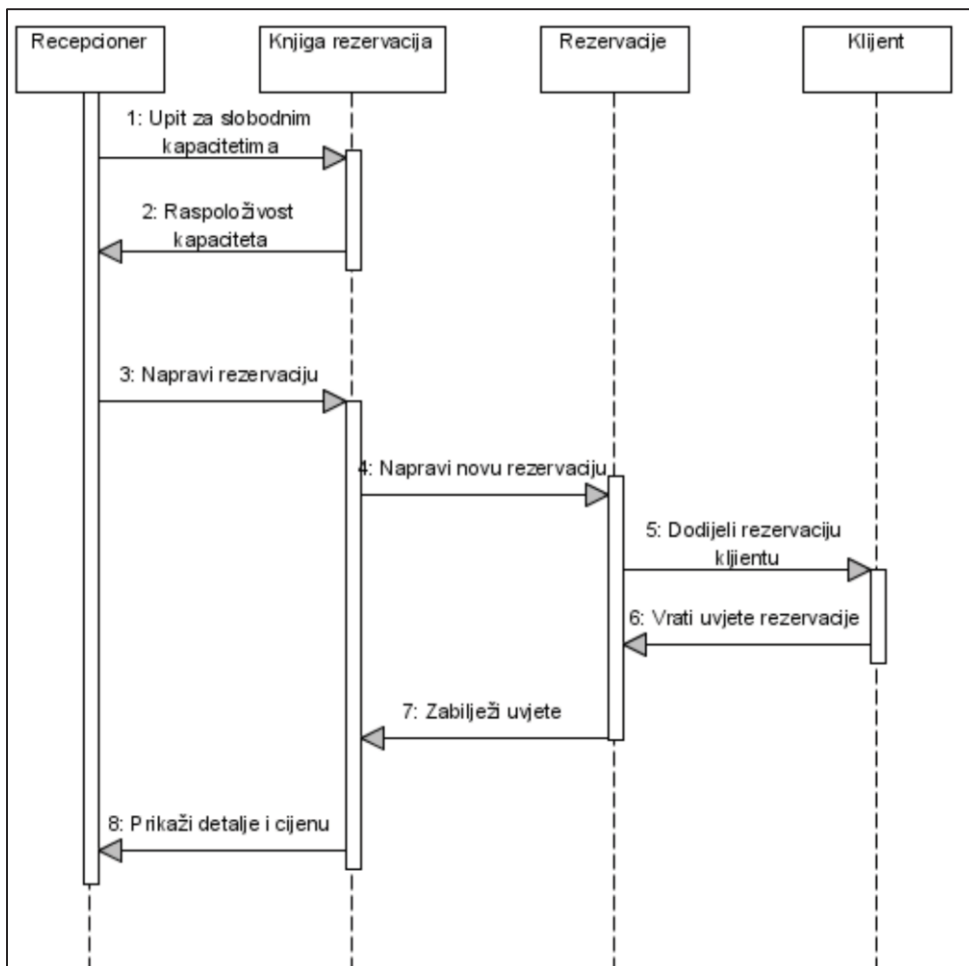
Notacija koja se koristi za izradu dijagrama slijeda prikazana je tablično u nastavku (Tablica 5.1).

Tablica 5.10 Notacija dijagrama slijeda

Notacija	Značenje
	Aktivnost
	Sudionik
	Vremenska linija (engl. <i>lifeline</i> )
	Poruka
	Odgovor na poruku

Dijagrami slijeda koriste se za vizualizaciju i istraživanje interakcije među objektima. Prikazuju vremenski slijed poruka koje se razmjenjuju između objekata. Objekti su prikazani u obliku pravokutnika te su poslagani od vrha prema dolje, s lijeva nadesno. U dijagramu koji je prikazan u nastavku (slika 5.18) objekti su recepcionar, knjiga rezervacija, rezervacija i klijent. Poruke koje se razmjenjuju između objekata su prikazane strelicom koja prikazuje smjer tijeka. U nastavku objekta se nalazi iscrtana linija koja predstavlja vremenski slijed.

Vrijeme teče od gore prema dolje, tako da se poruka *upit za slobodnim kapacitetima* vremenski šalje kao prva, a *prikazivanje detalja i cijene* kao zadnja. Vertikalnim pravokutnikom je prikazano područje djelovanja objekta, odnosno vrijeme u kojem je objekt aktivan. U nastavku je primjer dijagrama aktivnosti za rezervaciju smještaja.



Slika 5.18 Primjer dijagrama slijeda kod rezervacije

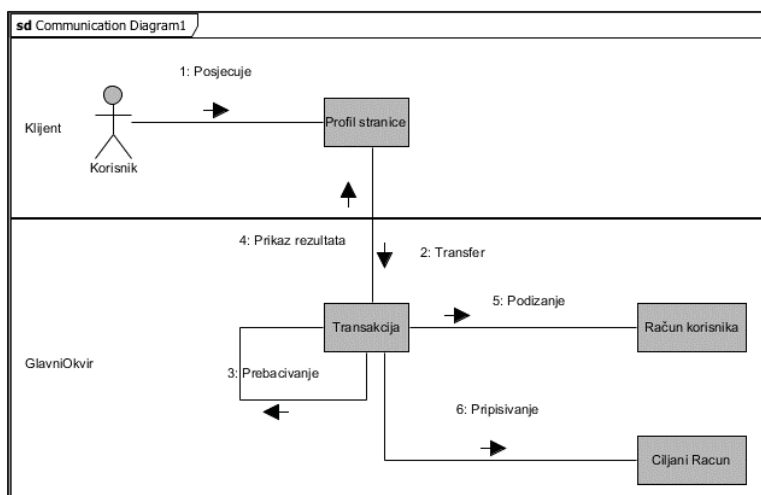
#### 5.3.1.4. Dijagram pregleda interakcije (engl. interaction overview diagram)

Dijagram pregleda interakcije je oblik dijagrama aktivnosti u kojem čvorovi predstavljaju interakcijske dijagrame. Interakcijski dijagrami mogu uključivati: (1) slijed, (2) pregled interakcije i (3) vremenske dijagrame (SpaxSystems, 2014). Većina notacija (označavanja) za dijagram pregleda interakcije je ista kao i za dijagrame aktivnosti. Primjerice, početni i završni čvor, čvor odluke, spajanja i čvorovi za pridruživanje su isti. No dijagram pregleda interakcije ima i dva nova elementa: (1) interakcija pojava i (2) interakcija elemenata.

5.3.1.5. *Komunikacijski dijagram (engl. communication diagram)*

Slično kao i kod dijagrama slijeda, svrha komunikacijskog dijagrama je modeliranje dinamičkog ponašanja slučajeva korištenja. U usporedbi s dijagramom slijeda, može se reći da je komunikacijski dijagram više fokusiran na modeliranje kolaboracije (suradnje) među objektima nego na vremenski slijed (VP, 2014).

Komunikacijski dijagram prikazan je na slici u nastavku.

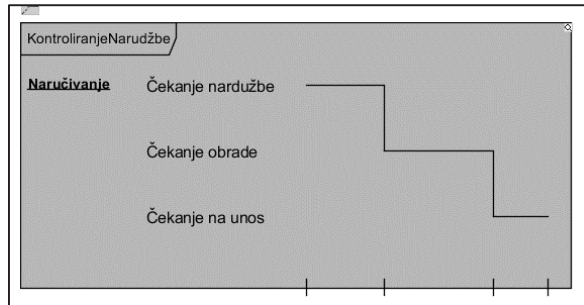


Slika 5.19 Komunikacijski dijagram

5.3.1.6. *Vremenski dijagram (engl. timing diagram)*

Vremenski dijagram pokazuje ponašanje objekata u zadanom vremenskom periodu. To je poseban oblik dijagrama slijeda. Elementi koji su dodani u dijagramu slijeda nisu pridodani odgovarajućem vremenskom dijagramu, ali elementi koji su dodani u vremenski dijagram se automatski dodaju u dijagram slijeda. Razlika između dijagrama slijeda i vremenskog dijagrama je u osima koje prikazuju vremensku dimenziju. Promatranjem vremenske dimenzije može se uočiti kako vrijeme prolazi slijeva nadesno, dakle vertikalno (VP, 2014).

Osnovna svrha je vrijeme, odnosno promjena stanja u određenom vremenu. Svrha vremenskih dijagrama može biti da prouče i modeliraju vremenska ograničenja i trajanje, kao što je prikazano na primjeru u nastavku (slika 5.20).



Slika 5.20 Vremenski dijagram

## 5.4. Izabrani alati za UML modeliranje

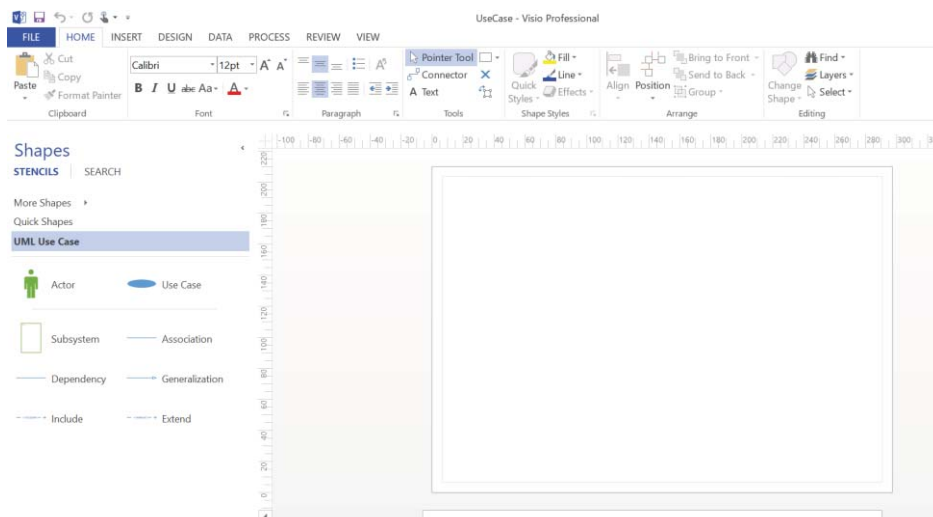
Na tržištu postoji niz alata koji omogućuju UML modeliranje. Ispod se nalaze opisi za odabrane alate koji su korišteni prilikom izrade UML dijagrama u ovoj knjizi. Uglavnom je riječ o internetskim alatima koji imaju besplatnu probnu verziju, ali po čijem isteku su isključivo komercijalni alati.

### 5.4.1. Visio

Alat *Visio* je nastao još 1990-ih, odnosno prva verzija je izašla 1992., a osam godina kasnije kupio ga je Microsoft (LucidSoftware, 2018). Pomoću alata *Visio* mogu se dobiti korisni dijagrami svih vrsta, a sadrži veliku biblioteku oblika/simbola koji se koriste u desetcima vrsta dijagrama. Visio pruža i podršku za izradu nekih od UML dijagrama (Purchase, McGill, Colpoys, & Carrington, 2001). U kontekstu UML-a, tek verzije *Visio 2013* i *Visio 2016* podržavaju UML 2.0 i UML 2.5 (Contini, 2018). Dijagrami koje je moguće pripremiti pomoću ovog alata su:

- 1) klasni dijagram,
- 2) dijagram slijeda,
- 3) dijagram slučaja korištenja,
- 4) dijagram aktivnosti,
- 5) dijagram stanja,
- 6) dijagram komponenti,
- 7) komunikacijski dijagram,
- 8) dijagram uvođenja.

Na slici u nastavku prikazan je izgled sučelja za alat *Visio 2013*.



Slika 5.21 Sučelje alata Visio za izradu UML dijagrama

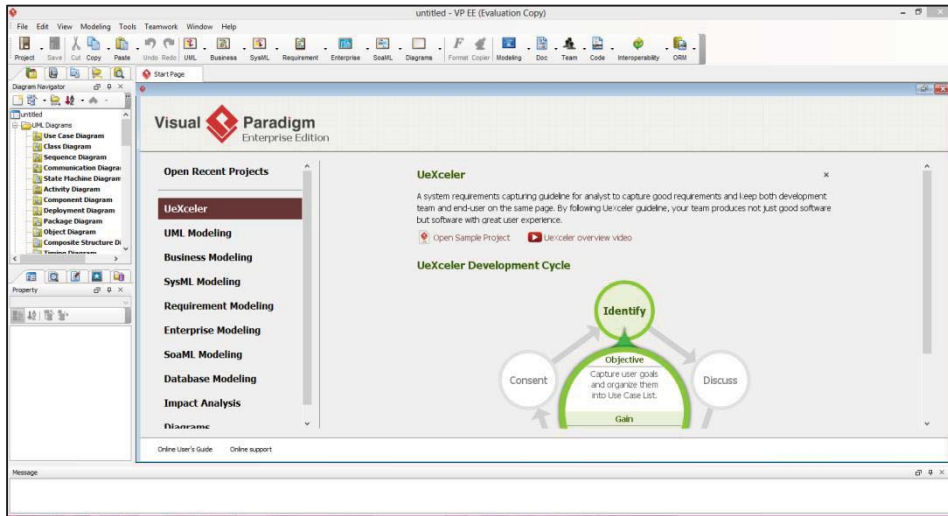
### 5.4.2. Visual Paradigm

*Visual Paradigm* je alat za dizajniranje softvera koji se pojavio još 2002. Podržava UML standard modeliranja, kao i ostale standarde (BPMN, DFD, ERD i sl). *Visual Paradigm* je alat koji radi na svim platformama, a korisnici su ga ocijenili najjednostavnijim za korištenje. U novije vrijeme postoji i internetska inačica alata *Visual Paradigm*.

To je samo manji dio funkcionalnosti alata *Visual Paradigm*. On ima više od 100 metoda i tehnika za sve faze razvoja poslovnog sustava, informacijskog sustava, aplikacija i podataka (odnosi se na verziju 16). Najvažnije od svega je što su sve one povezane, koliko god je to logički moguće. Dakle, nikako ga se ne može staviti samo u kontekst UML-a, već bi ga trebalo referencirati na mnogo mjesta, a posebno opisati njegovu filozofiju.

*Visual Paradigm* podržava UML 2.X s velikim nizom dijagrama, kao što su ranije navedeni: (1) klasni dijagram, (2) dijagram slučajeva korištenja, (3) dijagram slijeda, (4) komunikacijski dijagram, (5) dijagram stanja, (6) dijagram aktivnosti, (7) dijagram komponenti, (8) dijagram uvođenja, (9) paketni dijagram, (10) objektni dijagram, (11) strukturni dijagram, (12) vremenski dijagram te (13) dijagram pregleda interakcije. Modeli izrađeni u različitim alatima, kao što je *Rational Rose*, mogu se bez problema učitati u *Visual Paradigm*.

Na slici u nastavku prikazan je izgled sučelja alata *Visual Paradigm*.



Slika 5.22 *Visual Paradigm*

### 5.4.3. *Rational Rose*

*Rational Rose* je IBM-ov komercijalni alat za razvoj softvera baziran na UML standardu. Arhitekti, dizajneri, analitičari, dizajneri baze podataka i programeri koriste ovaj alat u svrhu izrađivanja vizualnih modela. Alat integrira dizajn i razvoj aplikacije. Neke od osnovnih karakteristika alata su (1) pružanje podrške za UML i (2) integracija s ostalim IBM-ovim alatima. Podržava rad na operativnim sustavima Windows i Linux.

*Rational Rose* omogućuje modeliranje slijedećih, prije spomenutih, UML dijagrama: (1) dijagram aktivnosti, (2) klasnih dijagram, (3) dijagram komponenti, (4) dijagram slučaja korištenja, (5) dijagram uvođenja, (6) dijagram stanja te (7) komunikacijski dijagram.

### 5.4.4. *Aris UML*

*Aris UML* je također jedan od komercijalnih alata koji ističe da svi dionici mogu koristiti standardni jezik za modeliranje kako bi procese unaprijedili na razinu aplikacije. Kao osnovne prednosti navode se: (1) integracija poslovnih procesa s projektima razvoja sustava koji se temelje na UML-u, (2) visoka kvaliteta softvera, (3) konzistentnost prilikom razvoja softvera te (4) podrška u vidu svih 2.5 UML dijagrama.

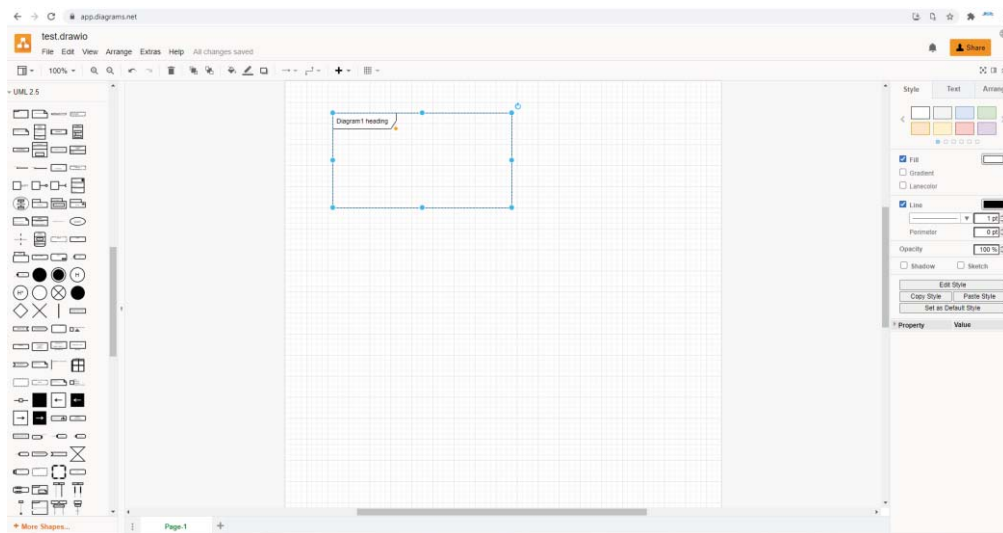
Osim 13 UML 2.x, dijagrama koje je moguće izraditi i u alatu *Visual Paradigm*, ovaj alat omogućuje modeliranje dodatna dva dijagrama (AG, 2013): (1) profilni dijagram te (2) dijagrama stanja protokola.

### 5.4.5. Ostali internetski alati

Pretraživanjem interneta moguće je pronaći brojne internetske alate i alate za desktop koji se koriste za UML modeliranje. Neki od njih dostupni su uz naknadu, a neki od njih imaju i opciju besplatnog korištenja. Jedan od alata koji je u potpunosti besplatan je *Draw.io*.

Riječ je o internetskom alatu koji podržava izradu brojnih UML dijagrama spomenutih u poglavlju 5.3.2. Konkretno, alat *Draw.io* podržava verziju UML 2.5 (<https://app.diagrams.net/>).

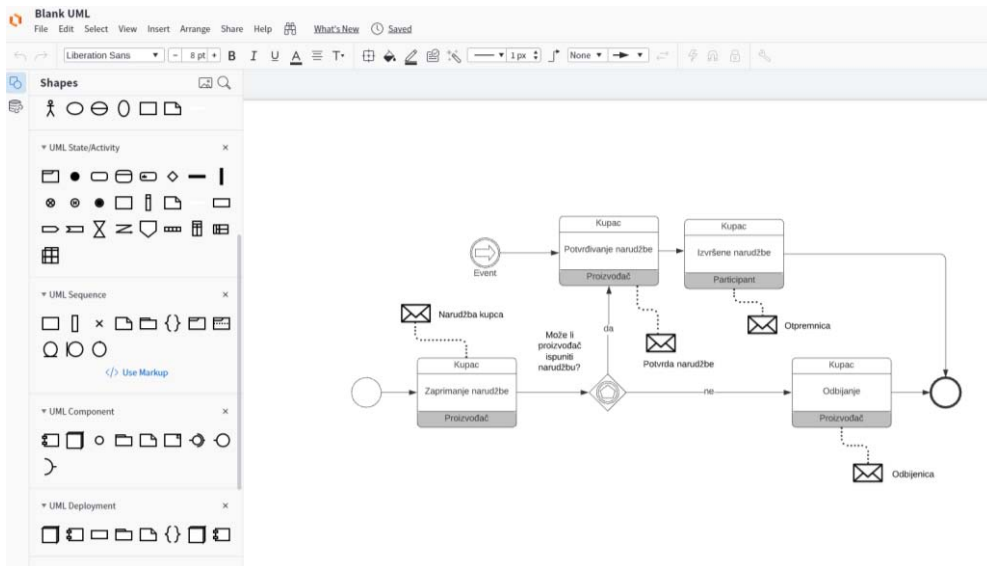
Prikaz sučelja je u nastavku.



Slika 5.23 Prikaz internetskog sučelja alata *Draw.io*

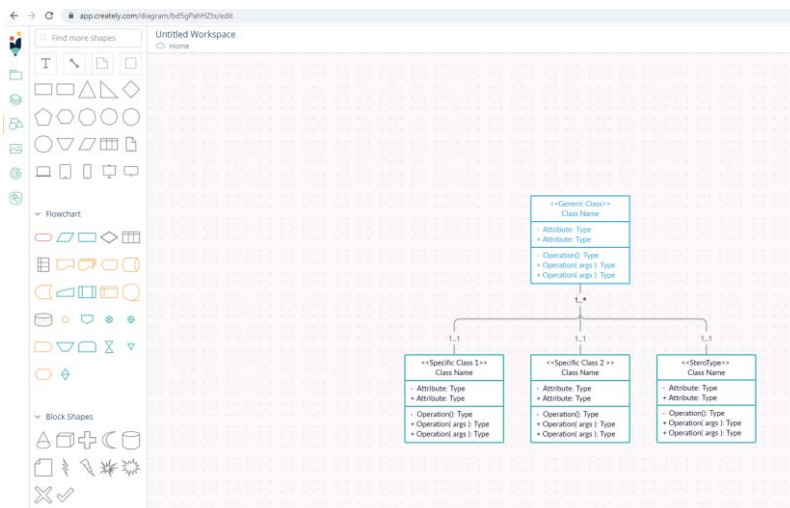
Jedan od internetskih alata koji također podržava veliki broj UML dijagrama je i *Lucidchart* (<https://www.lucidchart.com/>). U nastavku je prikazano sučelje alata.





Slika 5.24 Prikaz internetskog alata Lucidchart

Kao što je spomenuto, svaki od dosad navedenih alata podržava izradu samo pojedinih vrsta UML dijagrama. Zbog navedenog ograničenja, u prethodnim primjerima korišten je još jedan od internetskih alata, a to je *Creately* (<https://creately.com/>). Sučelje je također prikazano u nastavku teksta.



Slika 5.25 Prikaz internetskog alata Creately

Naravno, uz spomenute alate postoje još brojni drugi, a njihovo korištenje uvjetovano je potrebama modelara.

## 5.5. Literatura

1. AG, S. (2013). *ARIS UML Designer*. Software AG .
2. Booch, G., Maskimchuk, R., Engle, M., Young, B., Conallen, J., & Houston, K. (2007). *Object-oriented analysis and design with applications*. Perason Education.
3. Brooke, J. (1986). *Usability*, dostupno na: <http://www.usabilitynet.org/trump/documents/Suschapt.doc>.
4. Contini, E. (2018). *Drawing UML 2.5 diagrams with Visio 2016* (even with the Standard edition). Dohvaćeno iz <https://enzocontini.blog>: <https://enzocontini.blog/2018/07/10/drawing-uml-diagrams-with-visio-2016/>
5. Harrati, N., Bouchrika, I., Tari, A., & Ladjailia, A. (2016). Exploring user satisfaction for e-learning systems via usage-based metrics and system usability scale analysis. *Computers in Human Behavior* (61), 463–471.
6. LucidSoftware. (2018). *All About Microsoft Visio® for Diagrams*. Dohvaćeno iz <https://www.lucidchart.com>: <https://www.lucidchart.com/pages/what-is-microsoft-visio>
7. McLellan, S., Muddimer, A., & Peres, S. C. (2012). The Effect of Experience on System Usability Scale Ratings . *Journal of Usability Studies*, 56-67.
8. Purchase, H. C., McGill, M., Colpoys, L., & Carrington, D. (2001). Graph drawing aesthetics and the comprehension of UML class diagrams: an empirical study. *Research and Practice in Information Technology*, Vol. 9.
9. SpaxSystems (2014). *UML 2 Interaction Overview Diagram*. Dohvaćeno iz SpaxSystems: [http://www.sparxsystems.com/resources/uml2\\_tutorial/uml2\\_interactionoverviewdiagram.html](http://www.sparxsystems.com/resources/uml2_tutorial/uml2_interactionoverviewdiagram.html)
10. Systems, S. (2007). *Using UML part one - structural modeling diagrams*. Enterprise Architect.
11. UML-Diagrams. (2018). UML. Dostupno na: <https://www.uml-diagrams.org/uml-25-diagrams.html>
12. VP (2014). *Visual-paradigm*. Dohvaćeno iz Timing diagram: <http://www.visual-paradigm.com/VPGallery/diagrams/TimingDiagram.htmlv>

## 6 Modeliranje poslovne tehnologije

### 6.1 Procesni pogled i poslovna tehnologija

Cjelokupna naša razmatranja upotrebe IS-a/IT-ja uvijek polaze od organizacijskog sustava, tj. organizacije. Pokazali smo kako je za postizanje stabilnosti poslovanja u složenim sustavima, kao što je organizacijski sustav, potrebna velika usklađenost cijele organizacije. Generalno govoreći, cjelokupno funkcioniranje organizacije ostvaruje se izvođenjem velikog broja međusobno povezanih procesa koji se nazivaju *poslovni procesi* ili *poslovna tehnologija*. Međusobno povezani procesi ostvaruju sinergiju, odnosno njihovo zajedničko djelovanje i funkcioniranje je ono što najviše doprinosi funkcioniranju organizacije.

Pojam poslovne tehnologije podrazumijeva skup svih procesa koji su međusobno *povezani materijalnim i informacijskim tijekovima* bilo koje vrste, a izvode se radi postizanja misije i ciljeva zbog kojih ranije spomenuti organizacijski sustav (poduzeće ili ustanova) postoji i djeluje. Poslovna tehnologija koristi mehanizme kojima se organizacijama osiguravaju učinkovitije i efikasnije obrađivanje ulaza (*input*) i izlaza (*output*). Budući da poslovna tehnologija ima snažan utjecaj na sve poslovne procese, organizacija se na taj način ima priliku osnažiti u svakom pogledu. Materijalni i informacijski tijekovi, od kojih se poslovna tehnologija sastoji, obuhvaćaju širok spektar tehnologija, koji uključuju sve od samih računala pa do računalnih programa i sl. (Glenn, 2013).

Poslovna tehnologija je svakako i primjenjivanje IT-ja s ciljem ostvarivanja poslovne sposobnosti ili automatiziranja poslovanja. Poslovnom tehnologijom može se smatrati i rezultat koji se postiže konfiguriranjem, provedbom, primjenom i korištenjem IT-ja, a sve kako bi se ostvario poslovni rezultat. Ulaganje u tehnologije i poslovne tehnologije je investiranje koje se odnosi na stvaranje, korištenje i održavanje poslovne tehnologije (Fillios, 2008). Upravo se ulaganjem u pravu poslovnu tehnologiju ostvaruje konstantan napredak i uspjeh organizacije. Pritom izgradnja novog IS-a polazi od modela poslovnih procesa, uz anticipaciju potencijala suvremenog IT-ja. Iz toga se izvode specifikacije i razvija informacijski sustav koji će podržati izvođenje optimiranih, međusobno povezanih poslovnih procesa, tj. poslovne tehnologije (Brumec, 2011).

Dakle, kako bi se poslovna tehnologija efikasno primijenila, potrebno je na organizaciju gledati kao na skup međusobno povezanih procesa, odnosno ne samo kao na skup međusobno povezanih procesa. Dodatno na procese treba gledati kao na

srž organizacije (Willaert, Van den Bergh, Willems, & Deschoolmeester, 2012). Tada govorimo o procesnom pogledu, a može se naići i na izraze poput procesna orijentacija ili procesni pristup (Kaladić, 2012).

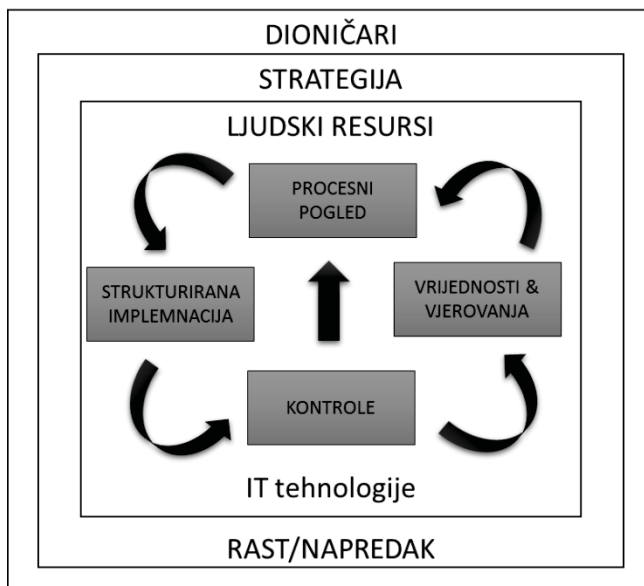
Primjenom, odnosno korištenjem procesnog pogleda ostvaruje se čitav niz prednosti, od kojih je najvažnija prednost povećanje efikasnosti (Kaladić, 2012). To je ono čemu svaka organizacija treba težiti. Kada je organizacija procesno orijentirana, to znači da osim izraženog pogleda na procese ima i karakteristiku agilne organizacije. Istraživanja su potvrdila da investiranje u postizanje procesnog pogleda ima značajan pozitivan utjecaj i na dugoročne i na kratkoročne performanse organizacije (Willaert, Van den Bergh, Willems, & Deschoolmeester, 2012).

Postoje tri dimenzije korištene u svrhu vrednovanja procesne orijentiranosti, a to su (McCormack, 2001):

1. Upravljanje procesima i mjerenje - postoje mjere koje uključuju procesne aspekte, kao što su kvaliteta rezultata, vrijeme trajanja ciklusa, trošak itd.
2. Procesni poslovi – zadaće koje su vezane za određene procese, kao i definirane uloge (npr. vlasnik procesa za razvijanje novog proizvoda).
3. Procesni pogled – iznimno detaljna dokumentacija i razumijevanje od vrha do dna, od početka do kraja procesa.

Dakle, zaključak je kako bi se svaka organizacija trebala kretati u smjeru stavljanja velikog naglaska na procese jer organizacijska paradigma polazi upravo od djelotvornosti i efikasnosti organizacijskih procesa. Promatramo li poslovnu tehnologiju kao živo biće, tada krvna zrnca predstavljaju poslovne procese. Organizacije posluju pod pritiskom različitih utjecaja iz okoline i upravo je upravljanje poslovnim procesima jedan od načina suočavanja s tim pritiscima (Willaert, Van den Bergh, Willems, & Deschoolmeester, 2012).

Cjeloviti pogled na upravljanje poslovnim procesima (engl. *business process management – BPM*) prikazan je na slici u nastavku (slika 6.1). Vidimo kako se na slici nalaze svi ključni dijelovi organizacije – počevši od dioničara, strategije, ljudskih resursa i IT-ja, dok se u samom centru nalaze poslovni procesi. Dakle, cjeloviti pogled nam još jednom ukazuje na iznimno veliku ulogu poslovnih procesa u organizaciji.



Slika 6.1 Cjelovit pogled na BPM (Willaert, Van den Bergh, Willems, & Deschoolmeester, 2012)

Zašto je uloga procesa toliko bitna? Bez obzira želi li se postići korijenita promjena ili manja izmjena u poduzeću, potrebno je započeti od poslovnih procesa. Zato što se, ako se želi postići bilo kakva korjenita promjena u poduzeću ili provesti bilo kakva izmjena, kao i u svrhu upravljanja procesima, potrebno spustiti na razinu procesa. Kada govorimo o promjenama, tada govorimo o unapređenju poslovnih procesa (engl. *business process reengineering*). Kako bi se to postiglo jako je bitno da svi sudionici u potpunosti razumiju poslovne procese.

Za potpuno razumijevanje poslovnih procesa potrebno ih je prvo otkriti, a potom i modelirati. Poznavanje onoga što se mijenja ključ je uspjeha. Krenimo stoga na upoznavanje poslovnog procesa.

Što je to poslovni proces? Proces je sve što se događa u nekoj organizaciji. Želimo li biti malo precizniji, najbolje je predstaviti genetičku definiciju poslovnog procesa. Poslovni proces je povezani skup aktivnosti i odluka koji se izvodi na vanjski poticaj radi ostvarenja nekog mjerljivog cilja organizacije, a traje određeno vrijeme i troši neke ulazne resurse pretvarajući ih u specifične proizvode ili usluge od važnosti za kupca ili korisnika (Brumec & Brumec, 2016). Na temelju ove definicije mogu se izdvojiti osnovne karakteristike procesa:

- 1) Povezanost aktivnosti i odluka – smišljeni skup radnji i odluka koje vode postizanju unaprijed definiranog cilja.
- 2) Vanjski poticaj za izvođenje – primjerice, u proizvodnim organizacijama vanjski poticaj je narudžba kupca.
- 3) Specifičnost proizvoda ili usluge – izlazni rezultat svakog procesa mora biti pojedinačno prepoznatljiv.
- 4) Važnost za kupca ili korisnika – organizacija ne bi postojala da nema smisla za korisnike, pa tako ni njezini poslovni procesi.

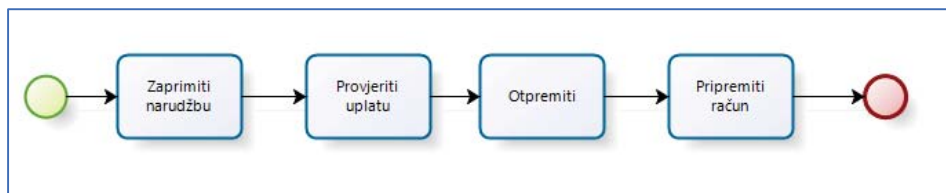
Postoje različite klasifikacije poslovnih procesa, a najčešće se dijele na ključne poslovne procese, procese podrške i upravljačke poslovne procese. Drugim riječima, razlikuju se:

- 1) Ključni procesi. To su procesi koji su karakteristični za svaku organizaciju (primjerice, gledamo li obrazovnu organizaciju, tada su njezini ključni procesi procesi studiranja, upisivanja, ocjenjivanja itd.).
- 2) Procesi podrške. To su procesi koji su manje-više isti kod organizacija, npr. procesi računovodstva, IT procesi itd.).
- 3) Upravljački procesi. To su procesi koje donose upravljačka tijela, a čiji rezultati imaju strateške utjecaje na funkcioniranje organizacije.

Kada se poslovni procesi neke organizacije identificiraju, tada slijedi modeliranje, odnosno prikazivanje tih poslovnih procesa na način da se jednoznačno opišu sva relevantna svojstva i pritom onemogućiti različita interpretacija istoga poslovnog procesa.

Poslovni procesi mogu pripadati, odnosno i pripadaju, različitim hijerarhijskim razinama. Modeliranje se radi na svim razinama, a najčešće se koristi pristup odozgo (engl. *top down*). To znači da se modeliraju „krovni“ procesi, odnosno procesi čije aktivnosti nisu detaljno razrađene, pa osobi koja promatra procese tek *spuštanjem* na razinu ispod može biti jasnije, odnosno može dobiti više informacija o detaljima procesa. Ono što je zajedničko svakom poslovnom procesu je da mora imati svoj početak i kraj, između kojih se nalazi slijed poslova ili radnih koraka (aktivnosti). Aktivnosti unutar procesa se ne odvijaju na proizvoljan način nego uvijek u definiranom slijedu (Brumec J., 2011).

Prikladan način modeliranja poslovnog procesa je njegov grafički prikaz. Primjer grafičkog prikaza procesa narudžbe slijedi na slici u nastavku (6.2).



Slika 6.2 Primjer poslovnog procesa

Kako bi se izbjegla mogućnost različite interpretacije i omogućilo računalno upravljanje izvođenjem poslovnih procesa, utvrđene su različite norme kojima se propisuje način prikazivanja i opisivanja procesa i njihovih odnosa (Aguilar-Saven, 2002). Najnovija i danas gotovo općenito korištena norma naziva se *Business Process Modeling and Notation* (BPMN), koja je u nastavku detaljno objašnjena (OMG, 2011).

## 6.2 Osnovni koncept modeliranja primjenom BPMN-a

Modeliranje poslovnih procesa je prikaz strukture i načina na koji se odvija neka djelatnost u promatranom poslovnom sustavu (organizaciji).

Analiza informacijskih sustava zahtijeva timski rad (vidi sliku dionika 1.5). S teoretskog stajališta postoje zapisane metodologije (prethodno poglavlje) te stvar izgleda jednostavno. Međutim, problem je u tome što one prikazuju samo smjernice, dok rezultat analize mora biti konkretan i razumljiv svim korisnicima. Drugim riječima, rezultat mora sadržavati sve – od opisa strukture s poslovnog aspekta do specifikacija za generiranje izvršnih procesa (Stein Dani, Freitas & Thom, 2022).

Za osobe koje se bave poslovanjem vrlo prikladan prikaz je *dijagram tijekom izvršenja poslova*. To se može potkrijepiti provedenim studijama *poslovnih analitičara* koji se bave modeliranjem poslovnih procesa (Aguilar-Saven, 2002; Underdahl, 2011). Pritom termin u množini koristimo kao sinonim za poslovnu tehnologiju, o čemu je već na početku bilo riječi. Poslovni procesi se povezuju tako da je izlaz jednog ili više procesa ulaz u jedan ili više procesa. Promatrano kao cjelina, tako povezani procesi čine poslovnu tehnologiju, odnosno način kako poslovni sustav ostvaruje svoju funkciju. Previše tehničkih zahtjeva opterećuju osobe koje se bave isključivo poslovanjem.

U posljednje vrijeme pojavile su se nove generacije jezika i alata za modeliranje poslovnih procesa. Postojeći alati se kontinuirano unapređuju, a novi poput BPMN-a razvijaju. U svakom slučaju, zajednički cilj je spustiti problem modeliranja poslovnih procesa na razinu razumljivu *običnim ljudima*.

Poslovni ljudi imaju jako dobru mogućnost vizualizacije poslovnih procesa u obliku dijagrama tijekom, no problem nastaje što se stvara jako veliki jaz između formata koji izvršava te procese i početnog dizajna procesa. Tu je važna uloga standarda, koji osigurava niz dijagrama namijenjenih za korištenje i osobama koje modeliraju procese i osobama koje upravljaju njima, a objašnjen je u nastavku. Prethodno spomenuti konzorcij OMG je razvio standard BPMN. Osnovni cilj ovog standarda je da se osigura notacija koja je lako čitljiva i koju razumiju svi poslovni korisnici (od poslovnih analitičara do razvojnih inženjera). Može se reći kako standard BPMN povezuje dizajn poslovnih procesa i implementaciju procesa. Detaljno raspisane (ažurirane verzije) specifikacije standarda moguće je pronaći na njihovim mrežnim stanicama (<https://www.omg.org/spec/BPMN>).

Dakle, BPMN pruža standardizirani vizualni mehanizam za poslovne procese koji su na taj način definirani u optimiranom jeziku za izvršavanje poslovnih procesa.

Kako biste znali nešto više o BPMN-u, potrebno ga je prvo definirati. Dakle, riječ je o notaciji (bilježenju, označavanju) koja se temelji na kombinaciji elemenata različitih prijašnjih notacija za modeliranje poslovnih procesa, kao što su XML Process Denition Language (XPDL) i Unified Modeling Notation (UML) (Dijkman, Dumas, & Ouyang, 2008; White, 2004). O UML-u je već bilo riječ u prethodnom poglavlju.

Notacija BPMN je stvorena s ciljem da je razumljiva različitim profilima ljudi (OMG, 2011) – zaposlenicima, analitičarima ili pak programerima (White S. A., 2008). Notacija ima za svrhu *spustiti* poslovne procese na razinu koja je definirana standardom BPMN-a te koja ne ovisi ni o alatu koji se koristi ni o metodologiji koja se primjenjuje (Silver, 2011).

Uvođenje ovog tipa standarda bi bilo kao da ljudi pričaju istim jezikom. Za poslovni svijet to bi značilo da bi se jednom definirani procesi mogli čitati u različitim alatima. To bi olakšalo razmjenu zaposlenika od jednog poduzeća do drugog, pripajanje poduzeća, zajedničke pothvate, kao i sama komunikaciju ili razmjenu transakcija među organizacijama. Radi se o standardu koji je svima poznat i svi ga jednako razumiju, a osnovne karakteristike su mu:

(1) jednostavnost (promatranje poslovnih procesa je jako jednostavno jer su svi strukturni elementi intuitivni) i



(2) fleksibilnost (mogućnost da svaka poslovna organizacija, primjenjujući ovu notaciju, modelira sve svoje poslovne procese, ovisno o vlastitoj potrebi).

**Opseg BPMN-a.** BPMN je ograničen na pružanje podrške samo konceptima i modeliranju koji se primjenjuju na poslovne procese. Svi ostali tipovi modeliranja su izvan opsega. Sukladno navedenome, slijede aspekti koji **ne pripadaju** opsegu:

- 1) definiranje organizacijskih modela i resursa,
- 2) modeliranje funkcijskih raččlambi,
- 3) podatkovni i informacijski modeli,
- 4) modeliranje strategije,
- 5) modeli poslovnih pravila.

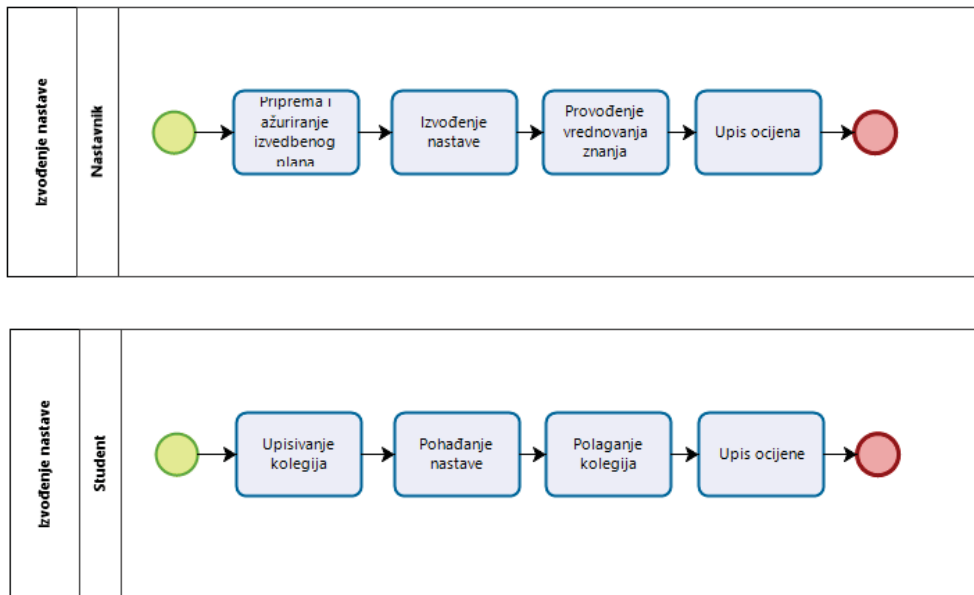
Dakle, bitno je imati na umu da, iako BPMN prikazuje tijek poruka te pridružuje podatkovne artefakte, ipak nije riječ o jeziku za prikaz podatkovnih tijekova.

Dijagrami BPMN mogu pokazivati procese različitih sudionika, što znači da na određeni proces različiti sudionici imaju **različite poglede** (iako se radi o istom procesu). Što to znači? Svaki pojedinačni sudionik ima pogled koji uključuje samo ono iz tog procesa što se direktno odnosi na njega, odnosno ima utjecaj na njega. Primjerice, aktivnosti procesa mogu biti:

- 1) unutarnje (ako su aktivnosti za sudionika unutarnje, to znači da je taj sudionik odgovoran za obavljanje navedenih aktivnosti) i
- 2) vanjske (aktivnosti za koje pojedini sudionik nije odgovoran).

Ako dijagram uključuje nastavnika i studenta kao sudionike, tada taj dijagram može imati dva pogleda. Iako dijagram prikazuje aktivnosti koje pripadaju i jednom i drugom sudioniku, prilikom izvođenja procesa svaki sudionik će imati kontrolu samo nad svojim aktivnostima.

Primjer izvođenja nastave iz perspektive nastavnika i studenta prikazan je u nastavku (slika 6.3).



Slika 6.3 Proces izvođenja nastave iz perspektive nastavnika i studenta

BPMN nema nikakve posebne grafičke mehanizme kojima bi se mogli naglasiti pogledi, stoga je taj dio prepušten modelararu.

Kako bi se olakšalo razumijevanje korištenja slijeda tijekom unutar procesa, uvodi se znakovni koncept (engl. *concept of a token*). Koncept prožima čitav slijed tijekom i prolazi kroz sve elemente procesa. Znakovni koncept je teorijski koncept koji se koristi kako bi se razumjelo ponašanje procesa koji se izvodi. Primjerice, ponašanje elemenata procesa može biti definirano opisivanjem interakcije elementa procesa sa znakom (*tokenom*) dok on prolazi strukturom procesa. Doduše, kod alata za modeliranje i alata za izvršenje procesa nije nužno implementirati niti jedan oblik znaka.

Primjerice, početni događaj generira *token* koji mora doći do kraja kada ga krajnji događaj mora koristiti (konzumirati). Put prolaska *tokena* se lako može otkriti.

### 6.3 Formalizacija poslovne tehnologije u skladu s normom BPMN 2.0

U prva tri poglavlja ove knjige pokazali smo kako utvrditi mogućnosti korištenja novih IT-jeva u svrhu strateškog razvoja i pozicioniranja poslovnog sustava. Nismo se bavili tehnologijom nego potrebnim i mogućim poslovnim unapređenjima. Razvoj IT-ja, kao što smo konstatalirali, može dovesti do potpune promjene procesa. Ono što je bitno je

da su te promjene usklađene sa zadanim ciljevima organizacije. Zašto se nismo bavili tehnologijom? Pa sama informatička tehnologija razvija se izuzetno brzo, tako da od samog usvajanja strateškog plana do njegove konkretne implementacije može doći do njezine promjene. Pogledajmo samo količinu novih (digitalnih) usluga koje je omogućio razvoj mobilnih aplikacija. Iz tog razloga je prilikom definiranja uloge IS-a/IT-ja u poslovanju potrebno imati jasnu poslovnu viziju, dok se sama tehnologija spušta na nivo detaljnije analize poslovnog sustava. Primjerice, izvedeni strateški cilj koji bi trebao doprinijeti razvoju organizacije može biti *poboljšanje komunikacije s korisnicima*. Međutim, za ostvarenje tog cilja potrebno je detaljno analizirati sam proces komunikacije s korisnicima te provesti detaljnu analizu isplativosti odabira raspoloživih IT-jeva. Sugestija je i kontaktirati eksperte promatranog područja koji poznaju trendove razvoja i poslovne prakse iz promatrane domene područja. Primjerice, svima je jasna važnost interneta u promociji usluge/proizvoda. Međutim, odgovor na pitanje koji su najbolji prodajni internetski kanali za konkretne slučajeve znaju samo vrsni marketinški stručnjaci.

U nastavku knjige bavili smo se nešto više informatičkim pristupom samom problemu. Dotaknuli smo se informatičkih termina, dijagrama i metoda kako bismo mogli percipirati što to projektanti IS-a trebaju dobiti od korisnika kako bi mogli razumjeti što se od njih traži. To znači da s jedne strane trebamo imati znanja za prikupiti zahtjeve, a s druge strane znati specificirati zahtjeve izvođačima poslova.

Upravo u tom smjeru ide i notacija BPMN, koja je podjednako razumljiva tehničkim i poslovnim stručnjacima. Notacija BPMN konsolidira najbolje ideje od svih drugih notacija i predstavlja informacijsko-komunikacijsko sredstvo za sve dioničare poslovnih procesa (Briol, 2010). U tom kontekstu, modeliranje poslovnih procesa promatra se kao dio modeliranja ukupne poslovne tehnologije. Klasifikacijom prema razini detaljnosti koja je ugrađena u normu BPMN 2.0 (OMG, 2011) definirano je da svaki praktično izveden sustav za modeliranje poslovnih procesa mora sadržavati notaciju za izradu modela na tri razine:

- Opisni model sadrži grafičke simbole i atribute koji su potrebni za modeliranje poslovnih procesa visoke razine apstrakcije. Slijed događaja i aktivnosti raspoređen je po organizacijskim jedincama i odgovornostima.
- Analitički model uključuje sve prethodno, ali detaljno razrađuje tipove događaja, skretnica i aktivnosti.

- Izvršni model omogućuje neposrednu izradu procesno orijentiranih aplikacija (POA), kao i njihovo izvođenje na specijaliziranom računalu.

Svaki od navedenih modela može se odnositi na trenutačno (*as is*) ili buduće (*to be*) stanje poslovnih procesa, a prikladan je drugi profil stručnjaka iz sfere poslovanja ili informatike. Potpuni model poslovnih procesa prema normi BPMN 2.0 ima četiri moguća dijagrama:

- dijagram poslovnog procesa ili slijedni dijagram,
- kolaboracijski dijagram,
- konverzacijski dijagram,
- koreografski dijagram.

Za izradu pojedinih dijagrama modela procesa koriste se objekti ili simboli koji imaju svoj grafički prikaz (pravokutnik, krug, romb i strelica) i funkcionalnu semantiku, o čemu će više riječi biti u nastavku.

### 6.4 Detaljno modeliranje poslovnog procesa

Detaljno modeliranje poslovnog procesa započinje izradom modela dijagrama privatnog procesa sa svim svojim objektima. Grafičko označavanje (notacija) je podijeljeno na više kategorija, kako bi korisnici lakše prepoznali objekt i razumjeli dijagram. Pet osnovnih kategorija su (OMG, 2008):

(1) Objekti tijeka (engl. *flow objects*) su glavni grafički elementi koji definiraju ponašanje poslovnog procesa.






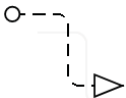

(2) Podatci (engl. *data*) su objekti koji mogu biti predstavljeni pomoću objekta podataka, podatkovnog *inputa*, podatkovnog *outputa* te skladišta podataka.

(3) Povezujući objekti (engl. *connecting objects*) mogu se podijeliti na nekoliko vrsta: (a) tijek slijeda, (b) tijek poruke, (c) asocijacije i (d) podatkovne asocijacije.






(4) Staze (engl. *swimlanes*) mogu biti *poolovi* ili linije (već prije objašnjeno).

Notacija nekih osnovnih objekata prikazana je u sljedećoj tablici. Simbolika ostalih objekata bit će spomenuta u daljnjem tekstu, prilikom detaljnijeg objašnjavanja postupka modeliranja.

Tablica 6.1 Notacija BPMN-a

Objekt	Sažeti opis	Notacija (oznaka)
Događaj	Događaj je vremenska točka koja ima neko posebno značenje za proces.	
Radni korak – Aktivnost	Aktivnost je generički izraz za rad koji neka organizacija obavlja u procesu.	
Potproces	Potproces je skup objekata tijekom povezan slijednim vezama koji se može promatrati kao zasebna cjelina.	
Skretnica (engl. <i>gateway</i> )	Skretnica je poslovno pravilo kojim se kontrolira razilaženje i spajanje tijekova u procesima i koreografiji.	
Slijedni tijek	Slijedni tijek ukazuje na redoslijed kojim se aktivnosti u procesu odvijaju (kao i u koreografiji).	
Tijek obavijesti ili poruka	Tijek obavijesti, kao što mu i samo ime kaže, ukazuje na tijek kojim obavijesti kolaju između sudionika (npr. <i>pool</i> može biti jedan od sudionika).	
Pridruživanje ili spajanje	Pridruživanje se koristi za povezivanje informacija i	

## 6: Modeliranje poslovne tehnologije

	artefakta s elementima BPMN-a.	
Polje	Polje grafički predstavlja pojedinog sudionika u kolaboraciji. Isto tako služi za odvajanje aktivnosti od ostalih polja.	
Staza	Staza se također koristi za organizaciju i kategorizaciju aktivnosti. Predstavlja poddio procesa, a ponekad i polja.	
Podatkovni objekt	Pružá podatke o tome što aktivnosti treba da bi se izvršila i/ili čime aktivnost rezultira.	
Grupa	Podrazumijeva grupiranje elemenata unutar iste kategorije. Ime kategorije predstavlja naziv grupe. Grupiranje se koristi u svrhu analiziranja ili dokumentiranja.	
Tekstualni komentar	To je mehanizam koji modelaru omogućuje da ubaci dodatni tekst za čitatelja dijagrama.	

Svaki od prethodno navedenih simbola ima svoje temeljno značenje, koje je potrebno prepoznati prilikom prikupljanja informacija te ga upotrijebiti i objasniti detaljnijom notacijom. Već prije je spomenuto da privatni proces može biti izvršni i neizvršni. Logički se nameće da je modeliranja potrebno započeti izradom neizvršnog (opisnog) te potom izvršnog (detaljnog) modela. Koristeći polja i staze analizirane u prethodnom poglavlju definiraju se granice promatranog procesa. Svaki proces započinje i završava nekim događajem. Međutim, događaji se nalaze i unutar samog procesa i predstavljaju vremenske točke koje imaju neko posebno značenje za proces, primjerice položen ispit. Kod modeliranja događaja uvijek treba paziti što je uzrok, a

što posljedica određenog događaja. To znači da događaj omogućuje izvršenje posljedične aktivnosti. Zbog toga je prilikom modeliranja događaja uvijek potrebno postaviti tri pitanja:

- *Kako se zove događaj?*
- *Što je uzrok da se taj događaj može dogoditi?*
- *Što kad se promatrani događaj dogodi?*

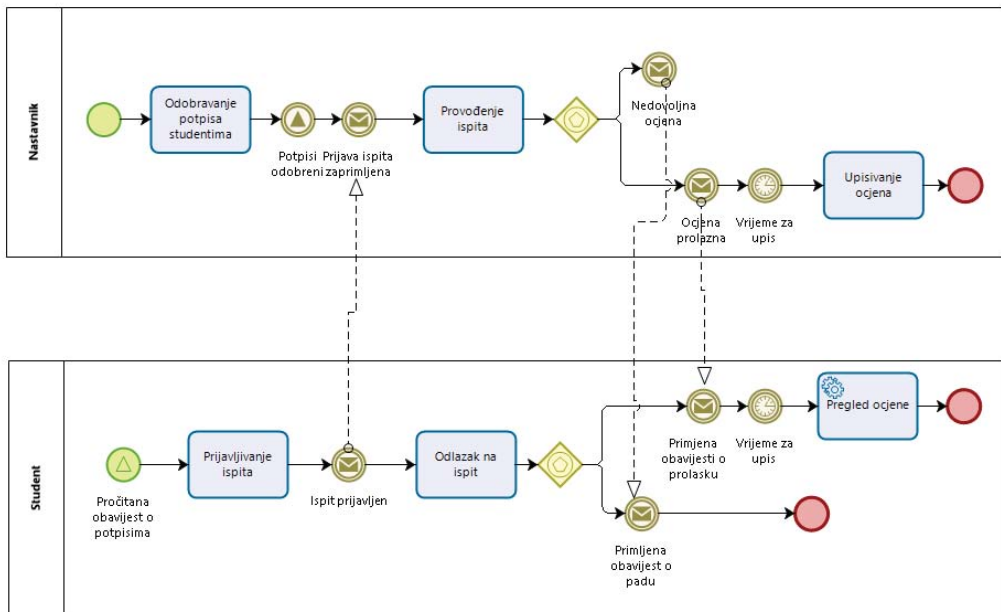
Prema notaciji BPMN 2.0, postoji 51 vrsta događaja, koji se mogu kategorizirati prema:

- položaju u procesu (početni, međudogađaj i završni);
- značenju događaja, što se definira intuitivnom oznakom unutar kruga (npr. sat za vremenski okidač, poštanska omotnica za obavijest, „X“ za prekid, „D“ za signal);
- utjecaju na izvođenje procesa (prekidajući i neprekidajući);
- smjeru djelovanja događaja (prijemni, koji registrira posljedicu nekog događaja ili aktivnosti s nekog drugog mjesta - npr. mjerač vremena – i prodajni, koji će izazvati posljedicu na nekom drugom mjestu).

Na slici u nastavku (6.4) prikazan je proces koji sadrži navedene događaje. Proces za studenta počinje primanjem signala da su potpisi dodijeljeni. Nastavno, signalni međudogađaj prima ili šalje signale – unutar navedenog procesa na slici postoji *Potpisi odobreni*.

Postoji više međudogađaja s obavijestima poput *Prijava ispita zaprimljena*. To znači da proces čeka zaprimanje prijave ispita da bi nastavio dalje s provedbom. Postoji i međudogađaj s oznakom vremenskog okidača, koji označuje da postoji kašnjenje, odnosno čekanje prije prelaska na drugu aktivnost.

Jedan takav primjer je *Vrijeme za upis*, s obzirom na to da upis ocjena dolazi s vremenskim odmakom.



Slika 6.4 Proces izvođenja nastave

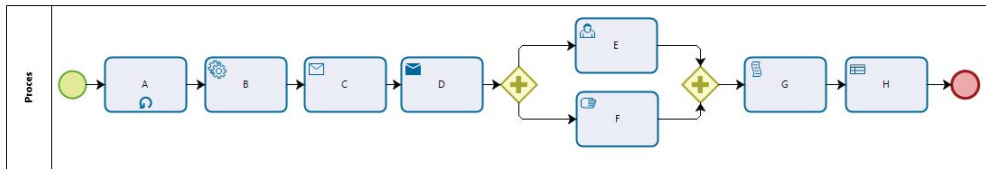
Aktivnosti su uzroci i posljedice nekog događaja. Aktivnosti predstavljaju svaki rad koji se izvodi u poslovnom procesu. Za njezino izvođenje potrebni su određeno vrijeme i resursi organizacije. Svaka aktivnost transformira ulazne tijekove u izlazne tijekove koji mogu biti materijalni, podatkovni ili logički. Strelica koja povezuje bilo koja dva objekta zove se slijednom vezom. Aktivnosti mogu biti jednostavne (predstavljaju elementarne radnje te se ne raščlanjuju nego predstavljaju radni korak) i složene (nazivaju se potprocesima, tj. mogu se razložiti na jednostavne ili promatrati kao jedna cjelina). BPMN 2.0 predviđa osam vrsta aktivnosti: (1) opća, (2) servis, (3) prijemna, (4) otpremna, (5) korisnička, (6) ručna, (7) poslovno pravilo i (8) naputak. Navedene vrste radnih koraka obvezno se navode kod analitičkog i izvršnog modela radi opisa ponašanja aktivnosti u procesno orijentiranim aplikacijama (Brumec, 2016).

Uzevši u obzir način izvođenja, BPMN 2.0 predviđa četiri vrste aktivnosti:

- ponavljajuća,
- višestruko paralelna,
- višestruko sekvencijalna,
- kompenzacijska.



Dijagram koji je prikazan na slici u nastavku (6.5) sadrži svih osam vrsta aktivnosti. Radni koraci B, C, D, E, F, G i H u gornjem lijevom kutu imaju prikazan mali simbol koji definirani o kojoj je vrsti riječ. Aktivnost A je opća aktivnost koja je ponavljajuća (simbol strelice), dok se aktivnosti E i F odvijaju paralelno.



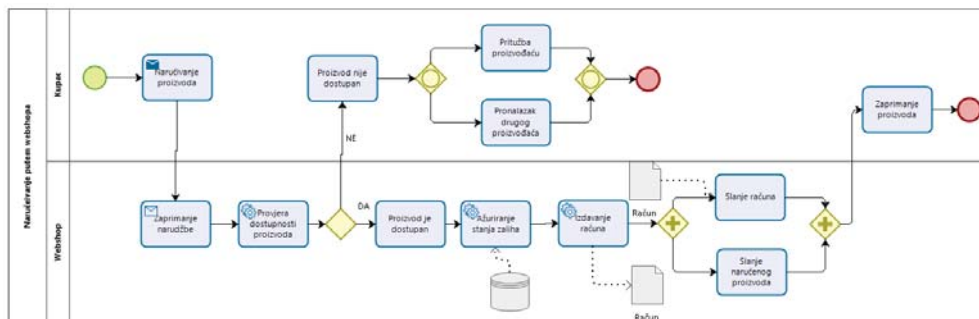
*Slika 6.5 Dijagram s različitim vrstama aktivnosti*

Sljedeći vrlo važan objekt u izradi modela je skretnica, koja formalizira poslovna pravila. Njezina osnovna svrha je upravljanje slijedom odvijanja aktivnosti. Skretnice mogu biti uvjetovane podacima ili događajima. Skretnice uvjetovane podacima upućuju slijed procesa ovisno o podatku koji donosi prethodna instanca procesa. Kod skretnica uvjetovanih događajima izlazni slijed će biti uvjetovan prethodnim događajem. Zbog toga se skretnice mogu jedino tumačiti pomoću slijeda aktivnosti koje se nalaze ispred i iza skretnice. Naime, skretnica ne dopušta izračune nego predstavlja pravilo po kojem se odvija slijed u procesu prema prethodnim izračunima, zbog čega se svi potrebni izračuni računaju u prethodnim aktivnostima (npr. student prošao ili pao ispit, ispuniti prijavnicu i uplatiti uplatnicu). Kao što se iz prethodnih primjera može vidjeti, postoji više vrsta skretnica. To su:

- ekskluzivna skretnica (koristi se za modeliranje isključivih aktivnosti),
- inkluzivna skretnica (koristi se za dodatne mogućnosti slijeda jedne aktivnosti),
- paralelna skretnica (koristi se za modeliranje više aktivnosti koje se trebaju izvršiti kako bi se slijed procesa nastavio),
- složena skretnica (koristi se za modeliranje kod kojeg se može dogoditi jedna ili više aktivnosti nakon nje, tj. nije moguće jednostavno objasniti izbor slijeda aktivnosti za izvođenje).

Slika u nastavku prikazuje tri tipa skretnica (slika 6.6). Riječ je o procesu naručivanja proizvoda u internetskoj trgovini. Prva skretnica je ekskluzivna skretnica jer proizvod može biti ili dostupan ili nedostupan – jedna aktivnost isključuje drugu. Ako je proizvod dostupan, iduća aktivnost je ažuriranje stanja zaliha. U tom dijelu procesa vidimo paralelnu skretnicu jer je riječ o dvije aktivnosti koje se odvijaju u isto vrijeme,

a to su slanje računa i slanje proizvoda. Ako proizvod nije dostupan, proces sadrži i inkluzivnu skretnicu, odnosno kupac može napraviti pritužbu i/ili pronaći novog proizvođača za željeni proizvod.



Slika 6.6 Dijagram koji sadrži više vrsta skretnica

Osnovni simbol skretnice je romb, u koji se crtaju markeri kako bi model bio razumljiviji. Više o skretnicama i njihovoj simbolici pogledati u Brumec, 2016.

Svi prethodno spomenuti objekti opisuju slijed tijekova unutar promatranog procesa. Osnovni elementi tijekova ovise o prirodi promatranog problema (proizvod, suglasnost, zahtjev, signal i sl.). Međutim, zajednički elementi svih procesa koji utječu na slijed odvijanja procesa su podaci. Zbog toga se posebna važnost pridaje podatkovnim objektima. Iako nemaju sve potrebne atribute za potpuni opis podataka, bitno ih je naznačiti kako bi se u daljnjoj specifikaciji zahtjeva mogli lako uočiti. Više govora o modelima podataka bit će u sljedećem poglavlju. Sam proces stvara ili koristi pojedinačne podatke, koje pretvara u informacije, zapisuje i čita iz baze. Prema notaciji BPMN 2.0, predviđena su četiri grafička simbola, pomoću kojih se mogu predočiti podatci koji se koriste u procesu. To su:

- Podatkovni objekti. Pravokutnik s gornjim preklapljenim desnim uglom u obliku dokumenta predstavlja općeniti i nestrukturirani skup podataka. Na slici 6.10 iznad imamo primjer „Račun“.
- Ulazni i izlazni podatci. Simbolika je ista kao kod prikaza općenitog podatkovnog objekta, samo što se unutar pravokutnika nalazi strelica koja označuje ulazni (ako je prazna) ili izlazni podatak (ako je puna). Na gornjem primjeru procesa, izlazni podatak za aktivnost „Izdavanje računa“ je podatkovni objekt „Račun“, dok taj podatkovni objekt „Račun“ postaje ulazni podatak aktivnosti „Slanje računa“.

- Spremište podataka. Predstavljeno je klasičnim simbolom baze podataka te podrazumijeva stalan zapis, neovisno o mediju na kojem je podatak pohranjen. Sa stajališta slijeda tijekova unutar promatranog procesa, skladište označuje da podatci ostaju i dalje zapisani, iako su izašli iz skladišta. Na prethodnoj slici imamo spremište podataka, odnosno bazu podataka koja predstavlja stanje na zalihama. Opisano se nalazi na slici 6.10.

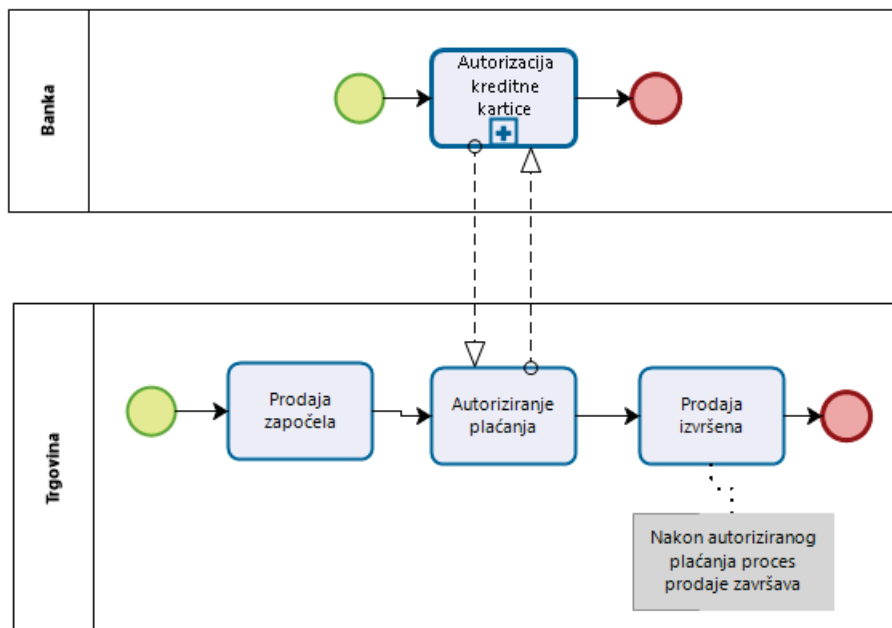
Povezivanje svih prethodnih objekata ostvaruje se objektima spajanja, na temelju čega se dobiva jedna povezana logička cjelina. U ovu grupu objekata spadaju:

- Slijedni tijek povezuje redoslijed izvršenja aktivnosti i uspostavlja se samo između objekata tijeka istog procesa. Primjerice, između pravokutnika (aktivnosti), kruga (događaja) i rombova (skretnice). Slijedni tijek može biti:
  - normalni – određuje redoslijed bezuvjetnog izvršenja,
  - uvjetni - određuje redoslijed uvjetnog izvršenja,
  - nadomjesni - određuje redoslijed uvjetnog izvršenja kad nije zadovoljen niti jedan uvjet.
- Tijekovi obavijesti su tijekovi poruka koji se koriste za modeliranje i razmjenu informacija među različitim sudionicima, tj. među različitim poslovnim procesima. Prikazuju se iscrtanom strelicom s kružićem na početku i trokutićem kao šiljkom na kraju.
- Tijekovi pridruživanja podataka označavaju se točkastom crtom s otvorenom strelicom. Za razliku od prethodnih tijekova, označuje tijek koji nema utjecaja na izvođenje procesa.

Na slici 6.7 u nastavku imamo prikazana sva tri primjera tijekova. Tako se, primjerice, slijedni tijek nalazi između početka (zeleni krug) i aktivnosti *Prodaja započeta* te između aktivnosti *Prodaja započeta* i *Autoriziranje plaćanja*. Aktivnosti *Autoriziranje plaćanja*, *Prodaja izvršena* te krug koji označuje kraj također su povezani slijednim tijekom.

Tijek obavijesti na slici je prikazan između aktivnosti dvaju različitih procesa, a to je *Autorizacija plaćanja*, koja pripada procesu trgovine, te *Autorizacija kreditne kartice*, koja pripada procesu unutar financijske institucije, odnosno banke.

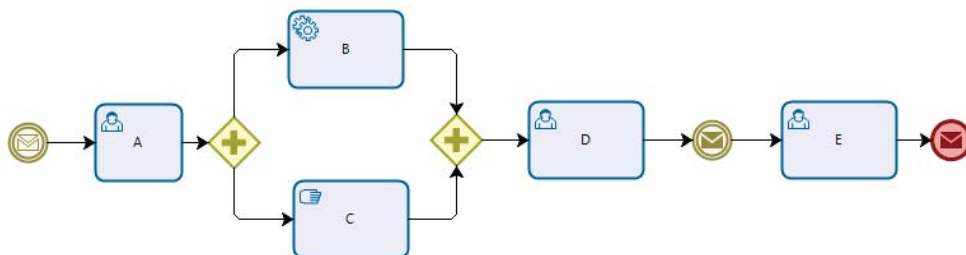
Treći tip tijeka, odnosno pridruživanje podataka, nalazi se u procesu prodaje, kada je aktivnost *Prodaja izvršena* povezana s tekstualnim komentarom, koji daje detaljnije objašnjenje osobi koja čita proces.



Slika 6.7 Slika s prikazanim tijekovima

#### 6.4.1 Dijagram poslovnog procesa ili slijedni dijagram

Dijagram poslovnog procesa ili slijedni dijagram predstavlja grafički prikaz detaljne logičke strukture modeliranog procesa (slika 6.8).



Slika 6.8 Primjer poslovnog procesa

Svaki poslovni proces ima svoj početak i kraj te se ponavlja svaki put kad treba izvršiti svoju svrhu. Sastoji se od više povezanih poslova ili radnih koraka, koji se općenito nazivaju aktivnostima. Aktivnosti se odvijaju u određenom slijedu (otud i naziv slijedni dijagram), prikazuju zaobljenim pravokutnicima i povezuju slijednom vezom, koja se prikazuje strelicom. Svaka aktivnost ima određeno vremensko trajanje i troši neke resurse. Početak i kraj predstavljaju događaje. Jednako tako, u slijednom dijagramu

postoje i međudogađaji. Općenito govoreći, događaj (engl. *event*) je vremenska točka koja ima neko posebno značenje u procesu. Više o objektima tijekom reči ćemo u nastavku ovog poglavlja. Poslovni procesi mogu biti privatni i javni.

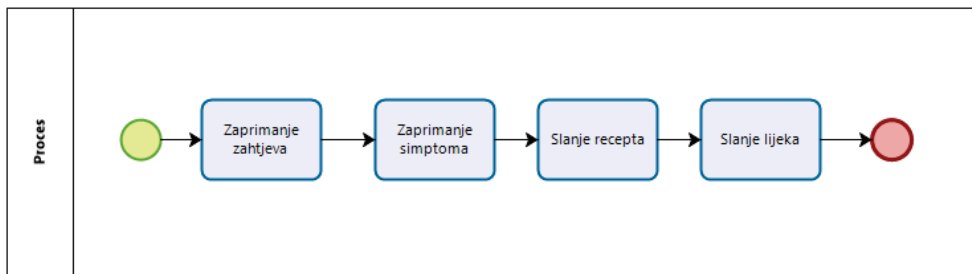
#### 6.4.2 Privatni poslovni procesi

Ovo je najčešći prikaz modela, koji se koristi za prikaz pojedinačnog i samostalnog procesa. Iako je formalno potpuno ispravan, često je nejasan i nepotpun (vidi sliku 6.9). To znači da se ne može utvrditi sama svrha procesa, rezultat njegova rada i kome je on potreban. Privatni poslovni proces je unutarnji proces neke organizacije, a može biti izvršni ili neizvršni. Ako je riječ o izvršnom procesu, to znači da je modeliran s ciljem da se izvrši prema semantici. Ako neki proces nema dovoljno detalja da bi se izvršio, tada je riječ o neizvršnom procesu. Cilj takvog procesa nije izvršenje nego dokumentiranje ponašanja procesa, tako da neće sadržavati nikakve informacije potrebne za izvršenje procesa. Kako bismo neki neizvršni model procesa pretvorili u izvršni, potrebno je ubaciti dodatne informacije, kao što su logički ili algebarski izrazi ili postupci obrade i prikaza podataka.

Preslikavanje grafičkih simbola objekata izvršnog modela poslovnog procesa u aplikacijske specifikacije i naredbe potrebno je raditi u skladu sa semantikom opisanom u normi BPMN. Usklađivanje redoslijeda odvijanja aktivnosti unutar jednog polja naziva se orkestracija.

Sada smo došli do pojma polje, koji određuje granice promatranog procesa kao cjeline. Polje je, sukladno normi BPMN 2.0 (OMG, 2011), grafički reprezentant sudionika u poslovnom procesu. Polje se u modelu procesa crta kao pravokutnik. Taj pravokutnik sadrži naslov i grafički opis procesa. U naslov polja upisuje se ime sudionika koji ima nadzor ili ovlaštenje za izvođenje svih aktivnosti koje se odvijaju u polju. Ako je riječ o konkretnoj organizaciji, onda polja mogu imati imena koja odgovaraju stvarnim jedinicama – računovodstvo, prodaja, hitni prijem i sl. (Brumec, 2011).

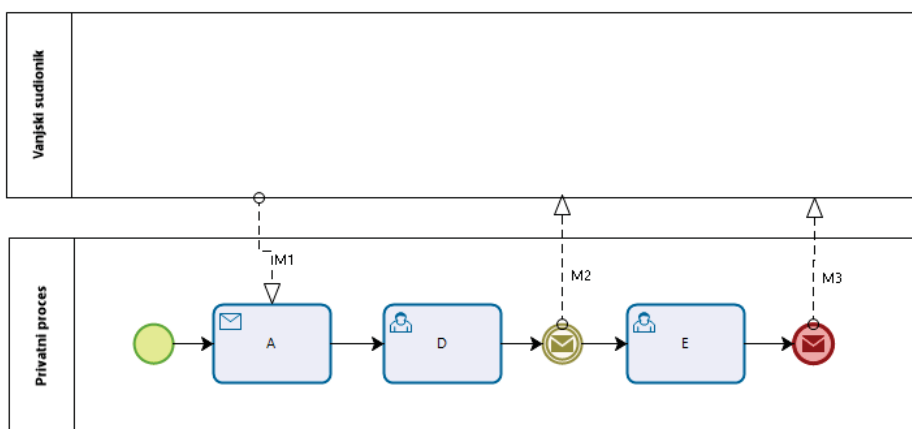
Staza (engl. *lane*) je dio polja koji se crta uz cijelu dužinu polja jer predstavlja specifičnog dionika procesa na detaljnijoj razini provedene analize. Primjerice, ako je polje predstavljalo organizacijsku jedinicu, onda će staza predstavljati radno mjesto unutar te organizacijske jedinice. Ime staze piše se uz užu stranicu pravokutnika.



Slika 6.9 Slika privatnog procesa

### 6.4.3 Javni proces

Možemo reći da u praksi ne postoje slučajevi u kojima se jednim poljem može opisati poslovna tehnologija. U suprotnom bi se postavilo pitanje potrebe za modeliranjem. Izgradnju modela poslovne tehnologije dijelimo na privatne procese, a njihovu interakciju modeliramo korištenjem javnih procesa. Javni proces predstavlja interakciju između privatnog poslovnog procesa i drugog procesa ili sudionika. Na taj način ograničavamo razmatranje na predmetni proces, iako ne poznajemo kako izgleda drugi privatni proces s kojim promatrani ostvaruje komunikaciju. Javni proces sadrži samo one aktivnosti koje se koriste u komunikaciji s drugim sudionicima uključenima u javni proces. Sve ostale unutarnje aktivnosti privatnog procesa se ne nalaze ovdje. Dakle, javni procesi prenose vanjskom svijetu tijek poruke i redoslijed poruke, sukladno interakciji. Javni procesi mogu biti modelirani odvojeno ili unutar kolaboracije (slika 6.10).



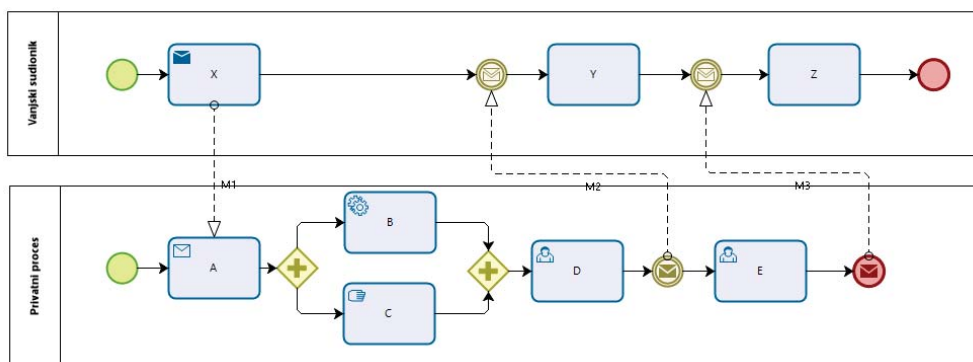
Slika 6.10 Slika javnog procesa

Na slici imamo privatni poslovni proces koji u prvoj aktivnosti prima poruku vanjskog sudionika, a nakon druge aktivnosti (D) šalje poruku tom vanjskom sudioniku. Po završetku privatnog procesa na slici se vidi još jedno slanje poruke, odnosno interakcija s vanjskom organizacijom.

Prethodno opisano podrazumijeva da javni proces predstavlja interakciju s vanjski sustavom, tj. pogled na tijekove obavijesti koje razmjenjuju dva procesa. Preporuke pretvaranja privatnog u javni proces idu u smjeru sažimanja nekih aktivnosti i ispuštanja nepotrebnih internih elemenata modeliranog procesa (Brumec, 2011).

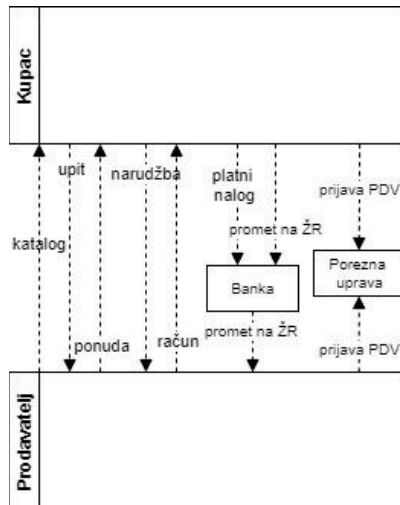
#### 6.4.4 Kolaboracijski dijagram

Kao što sam naziv dijagrama govori, kolaboracijskim dijagramom opisuje se kolaboracija, tj. suradnja više privatnih procesa (prikazanih odvojenim poljima). Potreba za ovakvim dijagramom proizlazi iz ograničenja povezivanja različitih polja (privatnih procesa) slijednim dijagramom. Možemo reći da kolaboracijskim dijagramom modeliramo nadsustav, koji se sastoji od sustava pojedinačnih privatnih procesa. Grafički ih prikazujemo privatnim dijagramima, koji su u međusobnoj interakciji, a sve u svrhu ostvarenja zadanog zajedničkog cilja. To znači da sudionici u kolaboraciji mogu biti privatni ili javni procesi. Prikaz sudionika poljima problem svodi na utvrđivanje tijekova obavijesti među njima. Primjer se nalazi na slici 6.11.



*Slika 6.11 Prikaz kolaboracijskog dijagrama između dva privatna procesa: nadogradnja na prethodnu sliku*

Uvođenjem dijagrama javnih procesa pojednostavljujemo pristup, a model svodimo na odnose među njima. Pritom se mora paziti da poredak tijekova obavijesti ukazuje na redoslijed razmjene informacija. Na slici 6.12 u nastavku prikazan je kolaboracijski dijagram na primjeru dva javna procesa (prodavatelj i kupac).



Slika 6.12 Slika kolaboracijskog dijagrama između sudionika

Postavlja se pitanje: odakle krenuti s modeliranjem, tj. koji dijagram prvo koristiti ako znamo što nam je zadani strateški cilj? Ne postoji jedinstveni odgovor. Sve ovisi o opsegu problema, kao i o iskustvu modelara. Ono što je bitno je da se modeliranje temelji na dva načela (Brumec, 2011):

- provjeriti jesu li pod pojmom „proces“ obuhvaćeni svi elementi čijim izvršavanjem se dobiva neka upotrebljiva vrijednost za korisnika,
- nedvojbeno utvrditi mogu li svi elementi koji su uključeni u jedno polje biti pod jednim vlasnikom procesa.

Sukladno prethodnom, objekte svrstavamo u ista ili u različita polja.





#### 6.4.5 Konverzacijski dijagrami

Konverzacijski dijagram je treća vrsta dijagrama. Riječ je o dijagramu koji vizualizira razmjenu poruka među više sudionika (engl. *pool*). Načelno se može reći kako je konverzacijski dijagram jednostavnija verzija kolaboracijskog dijagrama. Konverzacijski dijagram prikazuje pregled sudionika koji surađuju na određenim zadaćama. Heksagon i konverzacijska veza su dva dodatna elementa koja se javljaju u ovom dijagramu.

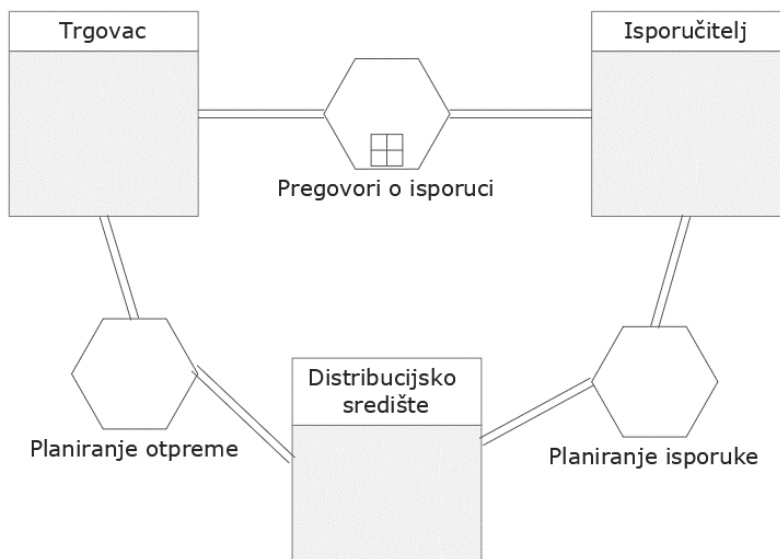
U nastavku je tablica s korištenom notacijom.



Tablica 6.2 Notacija konverzijskog dijagrama

	Komunikacijski element
	Konverzijski element <i>compound</i>
	Konverzijska veza
	Sudionik

Na slici 6.13 prikazan je konverzijski dijagram na primjeru trgovca i isporuke proizvoda. Preciznije, na slici se vidi primjer konverzijskog dijagrama koji se odvija između trgovca, distribucijskog središta, isporučitelja, konsolidatora, prijevoznika, špeditera, servisnog vozila te vozila službe.

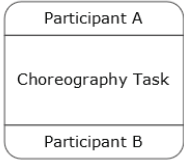



Slika 6.13 Konverzijski dijagram

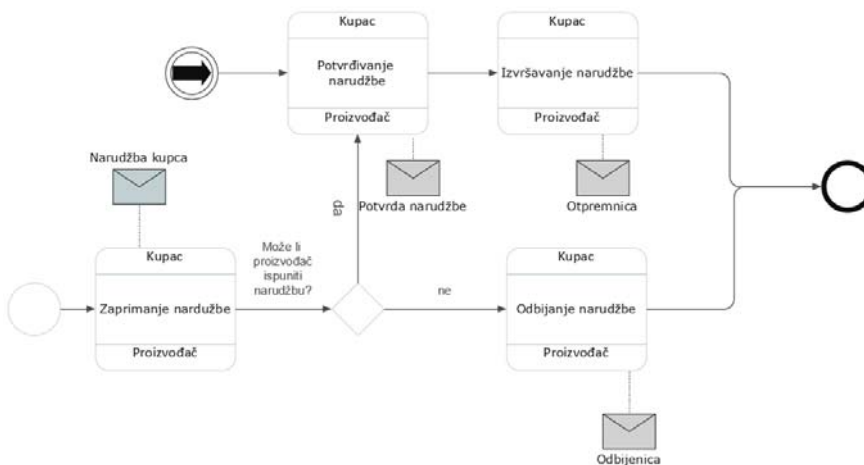
### 6.4.6 Koreografski dijagrami

Detaljnija razina konverzijskog dijagrama može se modelirati pomoću prethodno spomenutog kolaboracijskog ili pak pomoću koreografskog dijagrama. U nastavku je prikazan koreografski dijagram korištenjem notacije BPMN. Koreografski dijagram se fokusira na poruku i redoslijed primanja poruka među sudionicima procesa. Za razliku od ostalih BPMN dijagrama, koji definiraju aktivnosti među sudionicima, koreografski dijagram formalizira način na koji sudionici koordiniraju interakcije. Zbog navedenog naglasak nije na načinu na koji se odvijaju aktivnosti nego na razmjeni poruka među sudionicima. Notacija korištena u koreografskom dijagramu prikazana je u tablici 6.3.

Tablica 6.3 Notacija konverzijskog dijagrama

Notacija	Značenje
	Sudionik
	Poruka

Primjer razmjene poruke između kupca i proizvođača prikazan je na slici 6.14. Poruke su: *Narudžba kupca*, *Potvrda narudžbe*, *Otpremnica* i *Odbijenica*.



Slika 6.14 Koreografski dijagram (Izrađeno pomoću alata ConceptDraw)

## 6.5 BPMN model zrelosti organizacije

S pojmom modela zrelosti susreli smo se već i prije. Model zrelosti omogućuje organizaciji da metode i procese koje koristi usporedi s najboljom praksom, odnosno s jasno definiranim mjerama (engl. *benchmarks*). Zrelost se mjeri na način da se definira pripadnost nekoj od odgovarajućih faza.

Dobar model zrelosti osigurava niz prednosti, kao što su (Apmg, 2014):

- 1) točno definirane faze, koje nude preporuke kako unaprijediti organizaciju;
- 2) mogućnost usporedbe s drugim organizacijama ili prijašnjim stanjem vlastite organizacije;
- 3) naprednu samoprocjenu;
- 4) konzistentan niz upitnika i bodovanja;
- 5) Neovisan set definiranih mjera (engl. *benchmarks*).

Gartnerov BPM model zrelosti (engl. *business process management*) sastoji se od pet razina, koje pomažu organizaciji da definira svoju poziciju bazirajući se na kritičnim faktorima, koji su inače definirani BPM okvirom (Melenovsky & Sinur, 2006). OMG je normirao model zrelosti, a pet razina BPM modela zrelosti su (OMG, 2008):

- Razina 1 (Inicijacija). Poslovni procesi izvode se na nedosljedan, a ponekad i ad hoc način, koji ima teško predvidive rezultate;
- Razina 2 (Upravljanje). Uprava stabilizira posao unutar lokalnih radnih jedinica kako bi se osiguralo da se taj posao može provoditi na ponavljajući način koji zadovoljava primarne obveze radne grupe. Međutim, radne jedinice koje obavljaju slične poslove pritom mogu koristiti različite procedure.
- Razina 3 (Standardizacija). Zajednički standardni procesi se sintetiziraju iz najbolje prakse identificirane u radnim skupinama te se kreiraju smjernice za potporu različitim poslovnim potrebama. Standardni postupci predstavljaju ekonomiju obujma i temelj za učenje iz zajedničkih mjera i iskustva.
- Razina 4 (Predvidljivost). Sposobnosti omogućene standardnim procesima se eksploatiraju i vraćaju natrag radnim jedinicama. Učinkovitošću procesa upravlja se pomoću statističkih podataka kroz tijek rada u svrhu razumijevanja i kontrole varijacija kako bi se ishodi procesa mogli predvidjeti.
- Razina 5 (Inovacije). Proaktivne i oportunističke mjere za poboljšanje iziskuju inovacije koje trebaju prebroditi, odnosno zatvoriti praznine između trenutačne sposobnosti organizacije i sposobnosti potrebne za postizanje svojih poslovnih ciljeva.

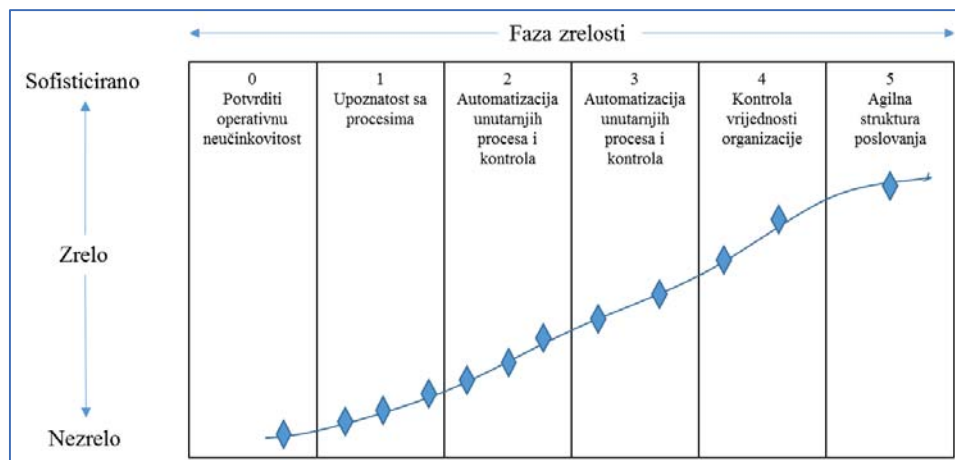
Autor je dao pregled nekoliko modela zrelosti, čije se faze nalaze prikazane u tablici 6.4 (Belaychuk, 2016).

Tablica 6.4 Faze modela zrelosti

SEI CMMI	Gartner	Forrester	ABPMP
	0. Svjesnost operativne neučinkovitosti	0. Nepostojanje	0. Ad hoc
1. Inicijacija	1. Svjesnost procesa	1. Ad hoc	1. Definiranost
2. Upravljanje	2. Automatizacija i kontrola unutar procesa	2. Ponavljanje	2. Kontrola
3. Definiranje	3. Automatizacija i kontrola među procesima	3. Definiranost	3. Arhitektura
4. Kvantitativno upravljanje	4. Procesi orijentirani ciljevima	4. Mjerenje	4. Proaktivno upravljanje
5. Optimizacija	5. Agilna poslovna struktura (optimirani procesi)	5. Optimizacija	

Krivulja koja je ugrađena u model zrelosti predstavlja količinu truda, a potom i koristi, koji pripadaju svakoj fazi. Kako se krećete prema naprednijim fazama, strmina krivulje ukazuje na to da je potrebno uložiti više truda, ali i da je povrat vrijednosti sve veći. Najbolji znak zrelosti je sljedeći: mudrost proizlazi iz ulaganja te mudrost podiže korisnosti.

Slika u nastavku (6.15) prikazuje šest faza modela zrelosti s odgovarajućom krivuljom.



Slika 6.15 Šest faza BPM modela zrelosti (Kerremans, 2008)

Osim navedenih šest faza zrelosti, druga važna dimenzija modela zrelosti su i organizacijski čimbenici, koji moraju biti uravnoteženi unutar i između faza. Postoji šest kritičnih faktora uspjeha koje organizacija mora prerasti prolaskom kroz faze – naravno, s konačnim ciljem upravljanja procesima.

Faktori uspjeha su (Melenovsky & Sinur, 2006):

1. strateško usklađivanje – kontinuirana povezanost prioriteta organizacije s poslovnim procesima, omogućujući pritom ostvarenje poslovnih ciljeva;
2. kultura i rukovođenje – zajedničke vrijednosti i stajališta koji oblikuju procesna ponašanja i stavove;
3. ljudi – individue i grupe koje kontinuirano rade poboljšanja i primjenjuju svoje procesno iskustvo i znanje;
4. uprava – relevantno i transparentno donošenje odluka;
5. metode – pristupi i tehnike koji pružaju podršku i omogućuju konzistentne procese i rezultate;
6. informatička tehnologija – softver, hardver te informacijski sustavi koji omogućuju i pružaju podršku procesnim aktivnostima.

Uspjeh upravljanja poslovnim procesima (BPM) ovisi o transparentnom i konstantnom unapređenju poslovnih procesa. To nastaje kao rezultat korištenja tehnika modeliranja poslovnih procesa, pristupa i alata. Normirana notacija za modeliranje poslovnih procesa je ranije spomenuti BPMN 2.0 (Institute, 2007).

Osim gore navedenog BPM modela zrelosti, postoji i procesni model zrelosti. Procesni model zrelosti se također sastoji od šest faza, a u nastavku će biti prikazana uloga BPMN-a u svakoj od faza procesne zrelosti. Faze su (Institute, 2007):

- Nepostojeća faza (0). Kako je karakteristika ove faze nedostak prepoznatljivih procesa, tako u ovoj fazi ne postoji potreba za modelima BPMN.
- Ad hoc faza (1). Organizacija prepoznaje neke poslovne procese, ali oni nisu standardizirani. I dalje ne postoji potreba za modelima BPMN, a procesi koji su prepoznati najčešće su modelirani na najvišem hijerarhijskom nivou – top model (Polančič, 2013).
- Ponavljajuća faza (2). Poslovni procesi u ovoj fazi imaju sposobnost da ih na sličan način izvode različiti ljudi (pojavljuju se slični i zajednički procesi). U domeni BPMN-a to znači da procesi mogu imati neke definirane podatke i da

su izvoditelji procesa sposobni definirati „svoj“ proces ako procesni analitičar to od njih zatraži (Polančič, 2013).

- Faza definiranja (3). Poslovni procesi u ovoj fazi su standardizirani i dokumentirani (najčešće procesni dijagram). U ovom trenutku definitivno je poželjno koristiti BPMN jer on pruža niz modela poslovnih procesa, npr. procesni model (engl. *process model*), kolaboracijski model (engl. *collaboration model*), koreografski model (engl. *choreography model*) te konverzacijski model (engl. *conversation model*) (Polančič, 2013).
- Faza upravljanja i mjerenja (4). Organizacija ovdje mjeri sukladnost obavljenog posla i definiranih procesa, kako bi se u slučaju neefikasnosti ili neučinkovitosti mogle poduzeti korektivne radnje. Proces bi trebali biti prikazani detaljnije – aktivnostima. U ovoj fazi bi se trebao koristiti čitav analitički set elemenata BPMN-a, tj. svi grafički elementi BPMN-a (Polančič, 2013).
- Faza optimizacije (5). U ovoj fazi svi procesi su poboljšani u skladu s dobrom praksom, a na temelju rezultata kontinuiranog unapređenja, na modelima zrelosti drugih organizacija te na temelju najbolje prakse. To bi značilo da su procesni modeli implementirani na način da su omogućene brze izmjene u modelima, kao i brza implementacija procesa u realne (trenutačne) poslove. U BPMN-u to znači da: (a) procesni modeli moraju biti izrađeni u okolini koja im omogućuje brze izmjene u samom modelu (primjerice, pristup modelima BPMN-a putem repozitorija ili oblaka); (b) promjene u procesnim modelima se moraju moći brzo evidentirati i primijeniti, što iziskuje učinkovitu komunikaciju među sudionicima procesa te prilagodljivu i integriranu IT podršku – najbolje bi bilo koristiti *Business Process Management System – BPMS* (Polančič, 2013).

Dakle, vidimo da se model zrelosti procesa sastoji od šest faza, kao i BPM model zrelosti. Poveznica svakako postoji, a zaključak je da je posljednja faza optimizacije svakako ono čemu svaka organizacija treba težiti.

Činjenica je da se napredovanjem iz prethodne u narednu fazu povećava i važnost utjecaja koju modeli BPMN, zajedno s IT podrškom, imaju na pozitivan ishod upravljanja poslovnim procesima (Polančič, 2013).

## 6.6 Osnovni pojmovi automatizacije poslovnog procesa

U prvom dijelu knjige smo analizirali važnost učinkovitog i efikasnog usklađivanja poslovanja u skladu s prilikama koje se javljaju na današnjem poprilično kompetitivnom globalnom tržištu. U prethodnim poglavljima opisani proces utvrđivanja i modeliranja zahtjeva krenuo je od poslovanja.

U ovom odjeljku opisat ćemo osnovne pojmove koji se koriste u daljnjem tijeku „informatizacije“, tj. razvoja aplikacija. Naime, sama namjena BPMN-a jest premošćivanje jaza u sporazumijevanju između poslovnih i informatičkih stručnjaka (Brumec, MPP). Na taj način poslovni stručnjaci koriste norme kojima opisuju procesni model, a softverski inženjeri upravo taj procesni model preslikavaju u izvršne aplikacije (sustave). Na tragu uspostavljanja što bolje komunikacije između poslovnih i softverskih stručnjaka razvijaju se novi dijagrami (koreografija i konverzacija) i značenja elemenata (poslovno pravilo, servisi, pozivajuće aktivnosti i globalni potproces) te ostali elementi norme.

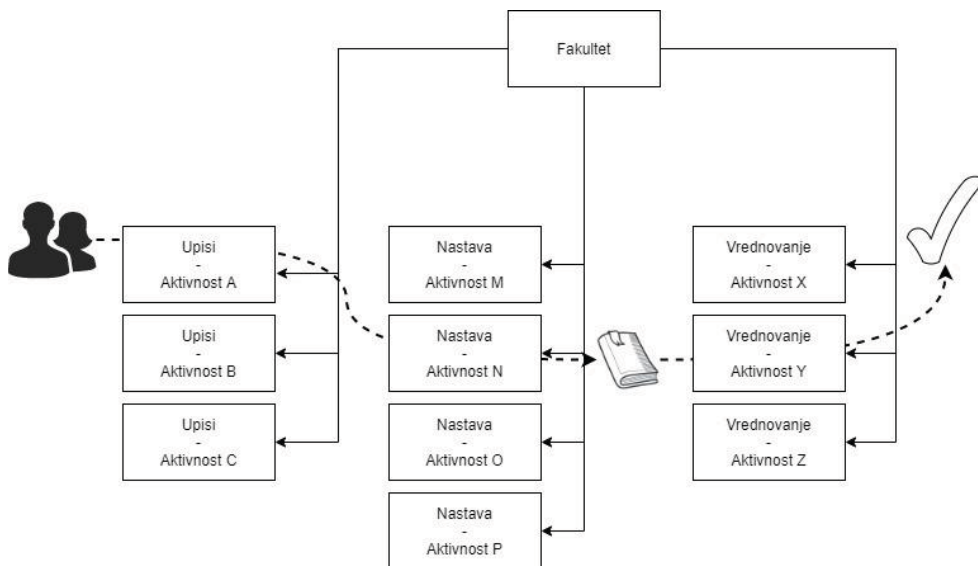
Upravo ovakva unapređenja omogućuju pretvaranje dijagrama poslovnih procesa u izvršne programe namijenjene krajnjim korisnicima. Ovakve programe zovemo procesno orijentirane aplikacije ili skraćeno POA.

### 6.6.1 Procesno orijentirane aplikacije (POA)

POA ne zamjenjuje klasične aplikacije, kao što su CRM, ERP, kadrovske ili računovodstvene ili pak internetske trgovine (Brumec, MPP). Svaka od navedenih aplikacija ima svoju funkciju za obavljanje specifičnih aktivnosti koje su sastavni dio jednog poslovnog procesa. Namjena POA-e upravo slijedi iz same definicije procesa (vidi definiciju procesa na str. 154) i ima za cilj povezivanje spomenutih klasičnih funkcijskih aplikacija u skladu s opisanim procesom.

Drugim riječima, POA je tijek aktivnosti, eksplicitan i neovisan o korisničkim i programskim sučeljima. Primjerice, aktivnosti na fakultet pripadaju različitim procesima, no moguće je da POA predstavlja aplikaciju koja omogućuje studentima upis na fakultet ili na pojedine kolegije, dok u isto vrijeme imaju pristup svim nastavnim materijalima, kao i vrednovanju pojedinih ishoda učenja.

Bez obzira što je riječ o aktivnostima koje, promatrano iz perspektive fakulteta, pripadaju različitim poslovnim procesima, aplikacija obavlja specifične aktivnosti spomenutih poslovnih procesa. Ilustrativan primjer dan je u nastavku (slika 6.16).



*Slika 6.16 Procesno orijentirane aplikacije podržavaju cjeloviti poslovni proces*

Primjer situacija u kojima treba razmotriti razvoj i korištenje POA-e može se sažeti ovako (Stiehl, 2014):

- 1) Specifični procesi tvrtke koji nisu podržani korporativnim aplikacijama.
- 2) Poslovi koji se obavljaju *end to end*, a obuhvaćaju više aplikacija, pa čak i izvan tvrtke.
- 3) Proces koji se provode uz iznimno opsežnu komunikaciju, koordinaciju i suradnju, a dosad su se obavljali razmjenu elektroničke (npr. e-poštom) i/ili papirnatu dokumentaciju.
- 4) Poslovi s velikim brojem poslovnih pravila.
- 5) Situacije u kojima treba pojednostaviti ili preložiti razne procese i/ili aktivnosti, transakcije, korisnička sučelja ili pak IT komponente.
- 6) Proces koji su kombinacija ljudskih i automatiziranih aktivnosti.
- 7) Situacije kada treba provesti „laku“ B2B integraciju poslovnih procesa, bez složenih i opsežnih podatkovnih transformacija, mapiranja i preusmjeravanja.
- 8) Samoposlužni procesi gdje je začetnik poslovnog slučaja istovremeno i stranka koja iščekuje rješenje svog problema iz prevođenja tog poslovnog slučaja, npr. izdavanje potvrde o studiranju.
- 9) Proces u kojima su promjene poslovnih pravila vrlo učestale ili se to očekuje.
- 10) Proces u kojima je izražena potreba za djelovanjem u realnom vremenu.



Procesno orijentirane aplikacije mogu biti izvedene na različite načine, ali glavno svojstvo gotovo svih komercijalnih alata je da POA izvode kao internetske aplikacije. Na taj način POA omogućuje integraciju s heterogenim sustavima korištenjem internetskog servisa. Takva integracija provodi se povezivanjem POA-e s internetskim servisima vanjskih aplikacija. Prema tome, procesno orijentirana rješenja koriste servisno orijentiranu arhitekturu. Izvedba procesno orijentirane arhitekture temelji se na pristupu odozgo (engl. *top down*), što znači da se pri projektiranju počinje s analitičkim modelom procesa, a završava izvršnim (Brumec, MPP). POA koristi servisno orijentiranu arhitekturu. Odnosno, POA su orijentirane korištenju poslovnih funkcionalnosti, a ne postojećim programskim sučeljima tih funkcionalnosti.

Poslovni stručnjaci i procesni analitičari najčešće nemaju dovoljno informatičkih znanja za prevođenje dijagrama poslovnog procesa u izvršnu aplikaciju. To je uloga programera, što implicira da je za modeliranje i simuliranje poslovnih procesa potreban interdisciplinarni tim. Iako poslovni stručnjaci i procesni analitičari modeliraju poslovni proces prema vlastitom poznavanju procesa, validacija izrađenog modela završava izradom POA-e. Usporedimo li modeliranje poslovnih procesa, simuliranje i izradu POA-e s klasičnim programiranjem, možemo konstatirati sljedeće (Brumec, MPP):

- (1) Modeliranje poslovnih procesa jest programiranje, ali na vrlo visokoj razini. Programski jezik je grafička notacija prema normi BPMN 2.0.
- (2) Naredbe programa BPMN su grafički simboli, a izvršni simbol je zapis XML za procesni pogon.
- (3) Model procesa mora biti sintaktički ispravan, kao i klasični program (ako nije, neće se moći pokrenuti simulacija poslovnog procesa i započeti izrada POA-e).

### 6.6.2 Servisno orijentirana arhitektura

Sagledavanje informacijskog sustava s aspekta realizacije ukazuje na tri komponente neophodne za funkcionalnu programsku potporu koja podržava poslovne procese (ERL, 2009):

- program kao funkcionalna cjelina koja podržava jedan segment ili cijelu poslovnu tehnologiju,
- programska arhitektura, tj. način na koji je programski sustav realiziran (u smislu načina rasporeda, realizacije i mehanizama povezivanja programskih komponenti),

- tehnološka infrastruktura kao programska i sklopovska podloga na kojoj programska arhitektura radi (nije predmet razmatranja ove knjige).

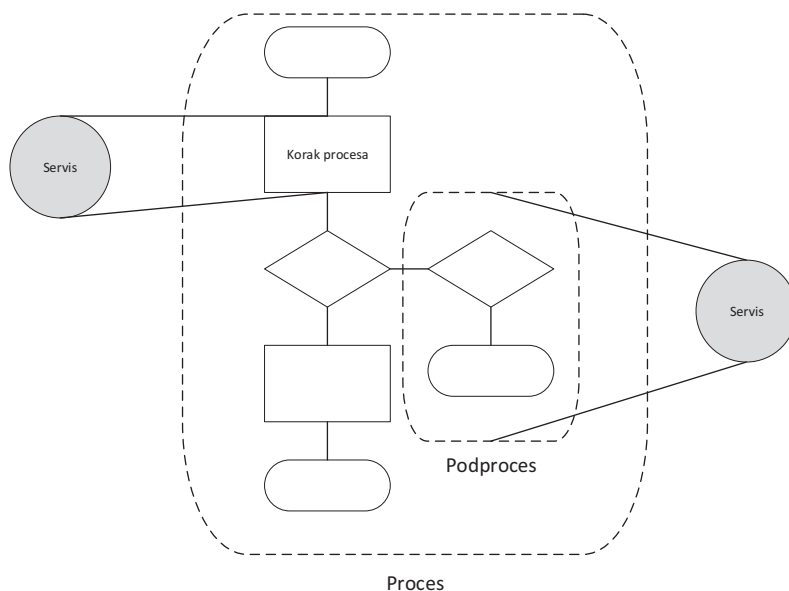
Prva točka u kontekstu programskog inženjerstva predstavlja korisničke zahtjeve iskazane na normiran način (također obrađeno u prethodnim poglavljima ove knjige). Važan trend u razvojnim paradigmatama je razvoj koji se temelji na modelima (engl. *model driven development*). Time se želi čvršće povezati grafičku prezentaciju problemske domene s realizacijom u programskom kodu. Za razliku od klasičnog razvojnog pristupa, u kojem se zapisani korisnički zahtjevi interpretiraju i ručno kodiraju u dorađenom programskom paketu, ovakvim pristupom sa stajališta programskog inženjerstva se nastoji što jače automatizirati prijelaz iz specifikacije korisničkih zahtjeva u obliku izvršnog modela procesa prema konačnom izvršnom programu (Brumec, MPP).

Programska arhitektura, tj. određivanje u kakvoj će arhitekturi biti izveden programski sustav, je važna zbog usklađenosti poslovnih procesa i programskog sustava u svrhu maksimiranja učinaka sustava u ostvarenju postavljenih strateških ciljeva i misije organizacije. Prema tome, programska arhitektura dobivena iz modela poslovnih procesa trebala bi biti izravno povezana s funkcionalnošću programskog sustava te predstavlja glavnu specifičnost servisno orijentiranih arhitektura.

Termin servisno orijentirana arhitektura, koji je osmislio Gartner, odnosi se na softver, odnosno softverski model u kojem se aplikacija sastoji od softverskih servisa i korisnika tih servisa (softverskih klijenata). SOA je okvir za izgradnju informacijskih sustava kombiniranjem različitih servisa (Komoda, 2006). Pristup SOA-e postaje standardan pristup koji se primjenjuje za IT sustave poduzeća. SOA omogućuje brži razvoj softvera, modularnost i dinamičko sastavljanje sukladno zadanim potrebama (Chen, Zhang, & Zhang, 2012). S obzirom na to da „proces“ može biti zamijenjen „servisom“, SOA može biti objašnjena i kao POA. Korist koju nudi SOA može se istinski ostvariti samo ako se primijeni u više različitih softverskih okruženja. Riječ je o izradi višekратно upotrebljivih i interoperabilnih servisa temeljenih na neovisnoj platformi koja ne uvjetuje određenog proizvođača softvera (Erl, 2005). Upravo povezivanje različitih platformi uz pomoć komunikacijske tehnologije povećava interoperabilnost među različitim sustavima (Bruning, Weißleder & Mirosław Malek, 2007).

SOA potiče individualne stavke povezane zajedničkom logikom da djeluju autonomno, ali ne nužno izolirano od drugih. Stavke povezane zajedničkom logikom poznate su kao servisi (Erl, 2005). Kontekst može biti povezan s poslovnim zadatkom, poslovnim entitetom ili bilo kojim drugim logičkim grupiranjem (grafički prikazano na slici ispod). Servis je jedinica programa koja se poziva standardiziranim procedurama,

a isto tako neovisno može izvršavati zadanu funkciju (Komoda, 2006). Navedeno je grafički prikazano na slici 6.17.



*Slika 6.17 Primjena servisa ovisno o različitim logičkim grupacijama*

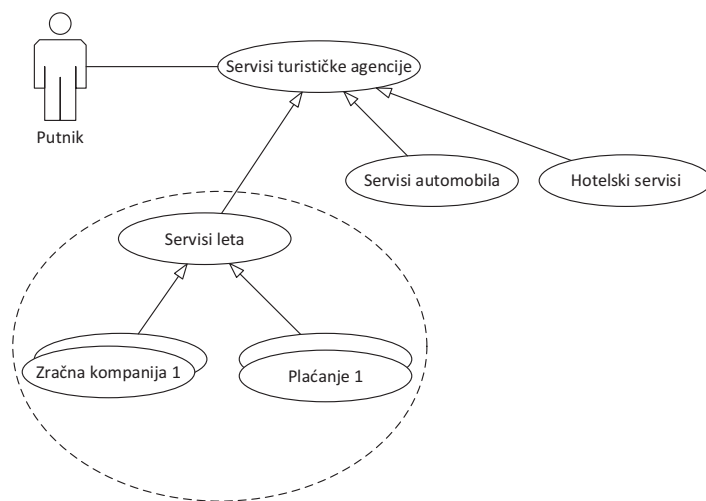
Unutar servisno orijentirane arhitekture servise mogu koristiti i drugi servisi ili softveri. Veze među servisima se baziraju na pretpostavci da servisi mogu biti u interakciji, tj. moraju biti svjesni jedni drugih. Svjesnost se realizira pomoću servisnih deskriptora (Erl, 2005). Kako bi uspostavili interakciju i postigli nešto, servisi moraju razmjenjivati informacije. Komunikacijski okvir je neophodan za održavanje njihovih labavih veza.

Neki od osnovnih principa SOA-e su (Erl, 2005):

- **Fleksibilne/labave veze.** Servisi održavaju veze koje minimaliziraju ovisnost i jedino što zahtijevaju je da budu svjesni jedni drugih.
- **Servisni ugovor (opis servisa).** Servisi se pridržavaju ugovora koji je zajednički definiran servisnim objašnjenjima i povezanim dokumentima.
- **Autonomnost.** Servisi imaju kontrolu nad sadržajem kojeg inkapsuliraju.
- **Apstrakcija.** Servisi skrivaju logiku, a jedino je sadržaj servisnog ugovora vidljiv. Koncept servisa se ponaša poput crne kutije (Akhrif, El Idrissi, & Hmina, 2018).
- **Mogućnost ponovnog korištenja.** Logika je podijeljena među uslugama upravo s ciljem ponovnog korištenja.
- **Kompozitnost.** Kolekcije servisa mogu biti koordinirane i sastavljene na način da sačinjavaju kompozitne servise.

- *Statelessness*. Servisi minimiziraju zadržavanje informacija koje se tiču neke akcije. Ne pamte stanje između dva pozivanja.
- Pretraživost. Servisi su dizajnirani na način da je do njih jednostavno doći i može ih se lako pronaći. Ideja je da postoji centralni servisni sloj koji predstavlja katalog servisa dostupan svim aplikacijama poduzeća.

Primjer turističke agencije s fokusom na rezervacijski servis zračne linije nalazi se na slici 6.18.



Slika 6.18 Prema Bruning, Weißleder & Mirosław Malek, 2007

## 6.7 Literatura

1. Aguilar-Saven, R. S. (2002). Business process modelling: Review and framework. *Production Economics*, 129-149.
2. Apmg. (2014). *What is a Maturity Model, and why use one?* Dostupno na: <http://www.apmg-international.com/en/consulting/what-maturity-model.aspx>
3. Belaychuk, A. (2016). *BPM Maturity Model: Go Deep vs. Go Wide Strategy*. Dohvaćeno iz bpm.com: <https://bpm.com/bpm-today/blogs/1056-bpm-maturity-model-go-deep-vs-go-wide-strategy>
4. Briol, P. (2010). *Distilled, Bpmn 2.0*. Lulu.com, 2010.
5. Brumec, J. (2011). *Modeliranje poslovnih procesa*. Varaždin: Koris.
6. Brumec, J. & Brumec, S. (2016). *Modeliranje poslovnih procesa*. Zagreb: Koris d.o.o.
7. Dijkman, R. M., Dumas, M., & Ouyang, C. (2008). Semantics and analysis of business process models in BPMN. *Information and Software technology* 50, str. 1281-1294.
8. Fillios, M. (2008). *Technology vs. Information Technology*, dostupno na: <http://www.cioupdate.com/insights/article.php/3742886/Business-Technology-vs-Information-Technology.htm>
9. Glenn, S. (2013). *Define business technology*: [http://www.ehow.com/about\\_6507460\\_define-business-technology.html](http://www.ehow.com/about_6507460_define-business-technology.html)
10. Institute, I. G. (2007). Cobit 4.1. ISA.
11. Kaladić, H. (2012). *Procesni pristup u poslovanju BPR*, dostupno na: <http://www.slideshare.net/setuplinks/procesni-pristup-poslovanjubpr-ppt>
12. Kerremans, M. Gartner (2008). *Assessment for Business Process Improvement Leaders: Six Phases for Successful BPM Adoption*. Gartner, Inc., 2008
13. McCormack, K. P. (2001). *Business Process Orientation: Gaining* .
14. Melenovsky, M., & Sinur, J. (2006). Having a BPM Maturity Model is Important for Long Lasting BPM Success. *Business Rules Journal*, Vol. 7, No. 12.
15. OMG. (2011). *Business Process Model and Notation (BPMN)*.
16. Polančič, G. (2013). *Process Maturity and BPMN*. London: Orbus.
17. Silver, B. (2011). *BPMN Modeling with Method and Style*. Cody-Cassidy Press.
18. Stein Dani, V., Freitas, C. M. D. S., & Thom, L. H. (2022). Recommendations for visual feedback about problems within BPMN process models. *Software and Systems Modeling*, 21(5), 2039-2065. doi:10.1007/s10270-021-00972-0
19. Underdahl, B. (2011). *Business Process Management*. IBM.
20. White, S. (2004). *Introduction to BPMN*. IBM Cooperation.
21. White, S. A. (2008). *BPMN - Modelling and Reference Guide*. Future Strategies.
22. Willaert, P., Van den Bergh, J., Willems, J., & Deschoolmeester, D. (2012). *The Process-Oriented Organisation: a Holistic View*.

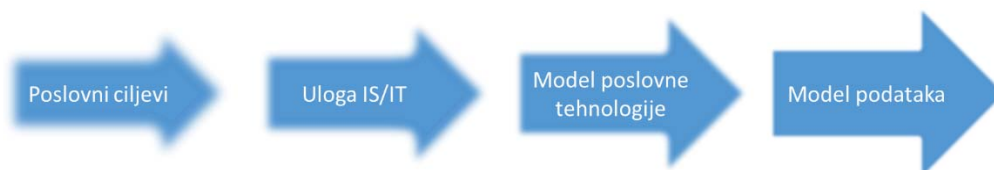


## 7 Modeliranje podataka

Bez obzira na mogućnosti koje pružaju generatori POA-e, logički ispravno i formalno korektno modeliranje podataka nužan je preduvjet za razvoj bilo kakvog informacijskog sustava ili aplikacije, pa tako i procesno orijentirane (Brumec, MPP). U petom poglavlju, kod objašnjavanja UML dijagrama navedeno je da postoje dvije vrste dijagrama – dijagrami kojima se opisuje stanje sustava i dijagrami koji opisuju dinamiku (ponašanje) sustava.

U šestom poglavlju smo opisivali način modeliranja poslovne tehnologije, tj. kako se izvršavaju poslovni procesi (ponašanje sustava). Već tada smo imali mogućnost notiranja potrebnih podataka koji se koriste za pokretanje ili u samim aktivnostima promatranog procesa.

Tako smo se približili i jednoj od posljednjih faza, nakon koje posao prepuštamo dionicima koji se bave *tehničkim izvođenjem informatičkih rješenja*. Kao što pokazuje i slika 7.1., jasnoća potrebne informatizacije nekog organizacijskog sustava sada postaje sve jasnija izvođačima IS-a.



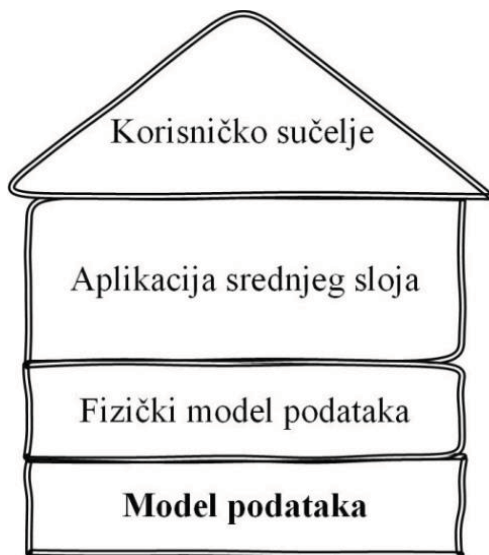
*Slika 7.1 Jasnoća specifikacija za razvoj novog IS-a*

Normalno, svaki od ovih koraka ima svoju važnost i predstavlja jedan od koraka, bez obzira na dinamiku ostvarenja plana razvoja IS-a. U narednim poglavljima detaljnije su pojašnjeni modeli podataka.

### 7.1 Model podataka – temelj IS-a

#### 7.1.1 Vrste modela podataka

Biste li izgradili kuću bez kvalitetnog temelja, ili bez temelja uopće? Naravno da ne biste. Kao što kuća bez temelja ne bi imala na čemu stajati, tako ni bilo kakav informacijski sustav bez modela podataka ne bi imao što obrađivati.



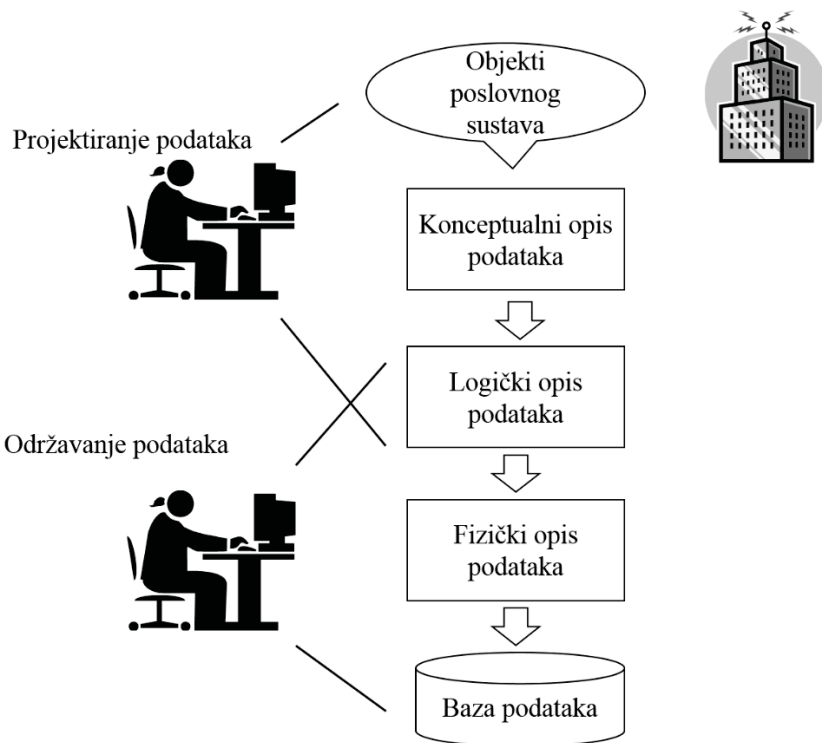
*Slika 7.2 IS kao kuća (Lurton, 2011)*

Modeliranje podataka je čin upoznavanja podatkovno orijentirane strukture. Podatkovno modeliranje predstavlja jednu od najkritičnijih faza prilikom razvoja baze podataka (Carli & Maguire, 2000). Poput ostalih modeliranja tvorevina, modeli podataka se mogu koristiti u razne svrhe – od konceptualnih modela do visokih fizičkih modela podataka. Ovisno o fazi razvoja informacijskog sustava, razlikuju se tri vrste modela podataka (Panian, 2005):

- 1) Konceptualni model predstavlja najvišu razinu apstrakcije modela, koja ne sadrži ni atribute, ni primarne ključeve, već samo entitete.
- 2) Logički model opisuje podatke na što detaljniji način, uključujući sve atribute, primarne ključeve i normalizaciju.
- 3) Fizički model predstavlja način na koji je model podataka implementiran u bazu podataka (vrsta podataka).

Slika 7.3 u nastavku prikazuje nastajanje baze podataka od konceptualnog do fizičkog modela.





Slika 7.3 Razvoj baze podataka (Vukmirović, 2013)

Kada pričamo o čitavom procesu dizajna, tada konceptualnom dizajnu prethodi faza planiranja i analize, a nakon fizičkog dizajna slijedi još i implementacija (Mamčenko, 2004).

Može se reći i da je modeliranje podataka formaliziranje i dokumentiranje postojećih procesa i događaja koji se javljaju tijekom dizajna i razvoja softvera. Modeliranje podataka je tehnika modeliranja za snimanje i „prevođenje“ kompleksnog realnog sustava u lako razumljive prikaze tijekova podataka i procesa, čime se stvara nacrt za izgradnju i/ili redizajniranje (Rouse, 2010).

Model podataka ima više uloga. Prva je da omogući dizajniranje podataka neovisno o karakteristikama implementacije. Drugim riječima, podatci se organiziraju bez izgradnje tehnologije, što bi značilo da su neovisni o hardveru i softveru. Druga uloga je prikazati podatke kako ih korisnik vidi u stvarnom svijetu (Mamčenko, 2004; Beynon-Davies, 2004). Upravo u tom smjeru (približiti korisniku shvaćanje podataka kao objekata realnog svijeta) ide i razvoj modeliranja podataka. Model podataka je formalni jezik za konceptualni prikaz i definiranje strukture podataka koje zahtijeva baza podataka, a rezultat navedenoga je shema za buduću bazu (Beynon-Davies, 2004). Strukture podataka uključuju podatke predmeta, povezanost između

podatkovnih objekata i pravila. Kao što sam naziv govori, model podataka usredotočuje se na podatke koji će biti potrebni i pitanje kako ih treba organizirati. Na taj je način analogan planovima arhitekata – nacrtu (Mamčenko, 2004).

Cilj modela podataka je da sadrži sve elemente koje baza podataka zahtijeva te da su ti elementi potpuni i točno navedeni (Mamčenko, 2004). Krajnji korisnici ga moraju moći razumjeti. U tu svrhu razvijaju se novi pristupi modeliranju i notacije.

Kao rezultat potrage za modelom koji će imati semantičko značenje realnog svijeta nastao je model podataka EVA. Naime, u svrhu što lakšeg sporazumijevanja s krajnjim korisnikom, model EVA (engl. *Entity – Relationship ER* ili *entitet – veza – atribut EVA*) koristi svakodnevnu terminologiju. Svaki entitet realnog svijeta (entitet) ima svoja svojstva (atribute), koji su povezani (veze) s nekim atributom nekog drugog entiteta realnog svijeta. Drugim riječima, podatci se u modelu EVA prikazuju entitetima i odnosima definiranim među njima. To je slično kao i kod relacijskog modela, ali relacijski model, za razliku od modela EVA, funkcionira na striktnim matematičkim pravilima, kao i drugi modeli podataka. Prema Chenu (2002), model EVA se bazira na modernoj algebri. S druge strane, relacijski model temelji se na relacijskoj algebri, a to je često teško razumljivo krajnjem korisniku.

### 7.2 Model podataka EVA: prirodan pogled na stvarni svijet

Model EVA (**entiteti, veze, atributi**) pruža prirodan pogled na stvarni svijet, pridajući veliku pažnju vezama – interakcijama među entitetima (Chen, 1976; Riccardi, 2002). Radi se o konceptualnom modelu podataka, koji također objedinjuje mrežni i relacijski model (Mamčenko, 2004).

Peter Chen je model EVA prvi put predstavio na konferenciji 1976. godine (Chen, 2002; Rob & Coronel, 2009). Model EVA prihvaćen je kao grafički alat za modeliranje podataka (Rob & Coronel, 2009), a nadopunio je relacijski model na način da pruža temelj okolini za razvoj relacijskih baza podataka. Struktura podataka se prezentira dijagramima – često se koristi i kratica ERD (Rob & Coronel, 2009; Rouse, 2005). Kako je osnovna komponenta modela EVA dijagram, jedna od definicija bi bila da je to i način grafičkog prikazivanja logičkih veza entiteta s ciljem kreiranja baze podataka (Rouse, 2005; Mamčenko, 2004).

#### 7.2.1 Osnovni elementi modela EVA

Kako je već navedeno, model EVA za prikaz podataka koristi entitete i veze među njima. To bi značilo da se radi o dvjema komponentama modela, kao što i mnogi

autori sugeriraju (Coronel, Morris, & Rob, 2011; Rob & Coronel, 2009; Xiaojuan, 2007). Međutim, postoji i atribut, kao treća komponenta (Vukmirović, 2013; Abhijit & Ahuja, 2007; Chen, 1976). Tako su osnovni elementi modela EVA:

- entitet,
- veza,
- atribut.

Svaki od elemenata je detaljnije pojašnjen u nastavku teksta.

Entitet (engl. *entity*) je glavni podatkovni objekt o kojem se prikupljaju informacije (Mamčenko, 2004). Radi se o prepoznatljivim konceptima, bez obzira jesu li u pitanju stvarne ili apstraktne stvari. Kao što će u narednom odjeljku biti spomenuto, postoji više notacija koje se koriste za grafički prikaz entiteta, a sve koriste pravokutnik (Coronel, Morris, & Rob, 2011). Naziv entiteta uglavnom se piše velikim slovima u sredini pravokutnika, a imenica mora biti u jednini (Rob & Coronel, 2009).

U kontekstu entiteta spominju se tri pojma (Abhijit & Ahuja, 2007; Beynon-Davies, 2004):

- 1) Tip entiteta. Ovaj pojam definira kolekciju entiteta koji imaju sve iste atribute (kategoriju). Kako je *tip entiteta* nezgodan pojam, većina ljudi koristi samo pojam *entitet* kao sinonim (Beynon-Davies, 2004).
- 2) Instanca entiteta. Odnosi se na pojedinačnu stavku iz kolekcije.
- 3) Unija entiteta. Predstavlja novi entitet stvoren na temelju atributa izabranih entiteta, tvoreći tako novi zajednički entitet.

Entitet ima svoju ulogu, tj. isti tip entiteta može sudjelovati u više vrsta veza. Primjerice, entitet ZAPOSLENIK može biti podređen ZAPOSLENIKU, s tim da se radi o različitim ulogama – menadžer i radnik (Davis, 2011). Drugim riječima, isti entitet se može ponašati različito „u vezi“ (Beynon-Davies, 2004). Kada veza povezuje dva entiteta istog tipa, tada moraju biti dodijeljene uloge. Funkcija uloge nije relevantna samo kod povratne veze (kada samo jedan entitet sudjeluje) nego i u binarnoj.

Dodatno, razlikujemo slabe i jake entitete. Jaki entitet je onaj čije postojanje ne ovisi o postojanju nekog drugog entiteta, dok je slabi entitet onaj čije postojanje ovisi o nekom drugom entitetu (Beynon-Davies, 2004).

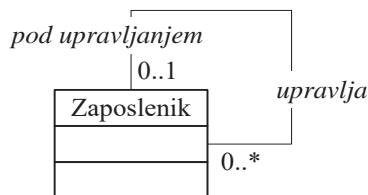
Veza (engl. *relationship*) predstavlja povezanosti (asocijaciju) među entitetima (Mamčenko, 2004). Veze se u grafičkom prikazu najčešće prikazuju linijama (Beynon-Davies, 2004). Primjerice, tip entiteta SUPRUG je s tipom entiteta ŽENA povezan

vezom „je u braku sa“. Kod modela EVA se za oznaku vrste veze koristi pojam povezivanja (engl. *connectivity*). Povezivanje obično nosi naziv glagola u aktivnom ili pasivnom obliku (Rob & Coronel, 2009), a karakteristično je i da je veza uvijek u dva smjera.

Student Mate Matić (instanca entiteta STUDENT) polaže (veza) ispit *Planiranje i analiza informacijskog sustava* (instanca entiteta ISPIT).

Nama bitna osnovna svojstva veze su stupanj veze i kardinalnost veze. Stupanj veze odnosi se na broj entiteta koji sudjeluju u vezi. Pridruživanje (veza) može biti definirano jednim, dvama ili više entiteta. Kardinalnost veze (ili brojnost veze) prikazuje minimalni i maksimalni broj instanci entiteta, koje mogu biti povezane s instancom drugog povezanog entiteta (Abhijit & Ahuja, 2007). Na sljedećim slikama koje prikazuju unarnu i binarnu vezu nalaze se i brojevi zapisani na vezama, koji označavaju kardinalnost veze. Notacija  $n1..n2$  označuje minimalan i maksimalan broj instanci.

Pridruživanje definirano na jednom entitetu zove se unarno pridruživanje. Primjer unarnog pridruživanja prikazan je na slici 7.4.

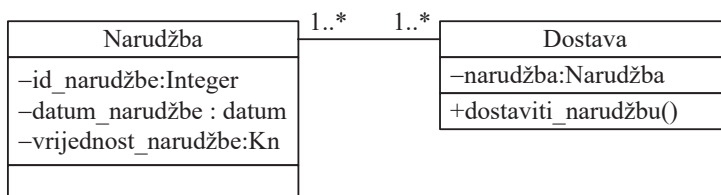


Slika 7.4 Unarno pridruživanje

Pogledajmo skup zaposlenika neke organizacije. O svakom zaposleniku se u kadrovskoj službi vodi isti propisani skup podataka. Međutim, jedan od zaposlenika može biti i voditelj. To znači da se taj zaposlenik razlikuje od ostalih samo po tome što je voditelj (npr. drugi opis poslova). Zbog toga se govori o istom tipu entiteta. Međutim, postoje i situacije u kojima je samo izvršitelj poslova te nema upravitelja (npr. obrt ili poduzeće s jednim zaposlenikom). Ovakve situacije opisujemo unarnom vezom nad istim entitom (zaposlenik). Objedinjavanje svih mogućnosti prikazuje se unarnom vezom na istom entitetu, odnosno skupom instanci Zaposlenik (skupom svih zaposlenika). Ovakvu situaciju modeliramo u skladu sa sljedećim objašnjenjem. Instanca entiteta Zaposlenik (jedan od zaposlenika) je pod upravljanjem nijedne ili jedne (0..1) instance entiteta Zaposlenik. Odnosno, instanca entiteta Zaposlenik (jedan od zaposlenika) upravlja nijednim ili više drugih zaposlenika (oznaka 0..1). Kao

što možemo vidjeti iz prethodnih primjera, u ovakvom pristupu su značajna tri broja (0 – nijedan, 1 – jedan ili \* – više), iako je dozvoljeno unošenje i drugih brojeva, čime se definiraju granice broja instanci u vezi.

Međutim, najčešće je riječ o povezivanju dva entiteta i naziva se binarno pridruživanje. Na slici 7.5 u nastavku prikazana je dvosmjerna binarna asocijacija. Pojam dvosmjerna upućuje da postoji povezanost u oba smjera. Pogledamo li realni sustav, kamion dostave sa sobom uglavnom ima više narudžbi. Isto tako, jedna narudžba može uključivati više artikala za dostavu, tj. više dostava. Formaliziramo li to u model EVA, to znači da dostava sadrži od jedne do neograničeno narudžbi (oznaka 1..\*), odnosno da entitet narudžba može imati jednu ili više dostava (oznaka 1..\*).



Slika 7.5 Binarno pridruživanje

Entitet čini set atributa, a **atribut** je svojstvo ili karakteristika entiteta (Abhijit & Ahuja, 2007; Chen, 1976). Vrijednosti atributa su ono što razlikuje instance entiteta jedne od drugih (Beynon-Davies, 2004). Prilikom navođenja atributa treba znati da atribut mora započeti velikim slovom, dok je ostatak riječi malim slovima. Isto tako, ako se atribut sastoji od više riječi, one moraju započeti velikim slovom.

Postoji više vrsta atributa:

- 1) jednostavni i složeni atributi,
- 2) viševrijednosni i jednovrijednosni atributi,
- 3) derivirani i nederivirani atributi,
- 4) ključni atributi.

Ako se zadržimo na istom primjeru, jednostavan atribut za entitet STUDENT bio bi *matični broj*. Složeni atribut je atribut koji se može podijeliti na dva ili više atributa, a dobar primjer je adresa. Drugim riječima, adresa studenta može se podijeliti na ulicu i kućni broj (dva atributa). Što se tiče viševrijednosnih i jednovrijednosnih atributa, kao dobar primjer može se navesti *kontakt-broj*, koji se može odnositi na „fiksnu liniju“ ili „mobilnu liniju“ (zato kažemo da je atribut viševrijednosni). Primjer jednovrijednosnog atributa može opet biti matični broj jer

postoji samo jedna vrijednost za svakog studenta. Derivirani atributi su oni atributi čija se vrijednost može izračunati iz postojećih atributa. Primjerice, jedan od atributa za studenta je *ocjena*. Iz svih ocjena moguće je izračunati prosječnu ocjenu iz svih kolegija za svakog studenta i tada prosječna ocjena postaje derivirani atribut.

Što se tiče ključnog atributa, on je od iznimne važnosti jer je to atribut koji je jedinstven za svakog studenta i ne može se dogoditi da dva studenta imaju istu vrijednost tog atributa. Kao primjer se opet nameće matični broj, ali može biti i OIB.

### 7.2.2 Notacije modela EVA

Prvo razjasnimo da je tehnika način na koji se neki postupak obavlja – u ovom slučaju način na koji se modeliraju podaci. Notacija je pak samo oblik iskazivanja tehnike i stoga za svaku tehniku postoje notacije.

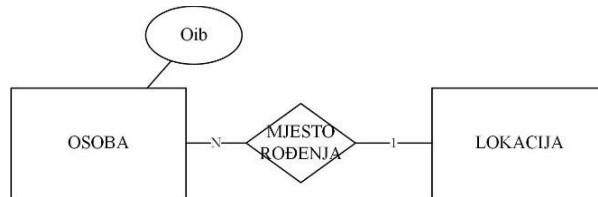
Tehnike modeliranja se odabiru ovisno o publici kojoj će model biti namijenjen. Publika je ta koja upravlja razinom detalja koju će model sadržavati. Primjerice, ako je model namijenjen analitičaru, tada on mora biti jasan i jednostavan za čitanje, čak ako to znači da se koristi *mala razina detalja*. Osnovni cilj modela je da on bude dostupan osobama koje nisu potkovane tehničkim znanjem. Suprotno tome, ako je neki model namijenjen dizajnerima baza podataka, tada on mora biti potpun model, sadržavati *što više detalja i biti što potpuniji*, kako bi što realnije prikazao potrebu za podacima.

Svaka tehnika koristi različite simbole, a bez obzira na to koji se simboli koriste, modeliranje podataka ima samo jedan cilj: opisati stvar za koju organizacija želi prikupiti podatke. Upravo zato se sve notacije lako konvertiraju u druge, a osnovna razlika među njima je u estetici (Hay, 1999).

Odabir notacije navedenih tehnika nema veliku ulogu i nije toliko bitan koliko je bitna kvaliteta i dosljednost konačnog dijagrama. U nastavku slijedi prikaz odabranih notacija (tehnika) na istom primjeru, tj. prikaz dva entiteta: OSOBA i LOKACIJA. *Osoba* je rođena na *lokaciji*, odnosno *lokacija* je mjesto rođenja *osobe*. Veza između entiteta je da jedna osoba može imati samo jednu lokaciju kao mjesto rođenja, dok na istoj lokaciji može biti rođeno više osoba (1:M). Kao primjer je naveden samo jedan atribut (OIB), i to kao ključni atribut entiteta OSOBA. U nastavku slijedi navedeni primjer prikazan pomoću različitih notacija.

Davne 1976. Peter Chen je predstavio model EVA korištenjem vlastite notacije. Iako on nije u potpunosti razvio koncept notacije, njegov rad je pomogao u standardizaciji

i popularizaciji – kako modela, tako i notacije. Radi se o jednostavnoj notaciji koja predstavlja temelj za sve ostale notacije. Chenova notacija prikazuje entitete, atribute i veze, a kako se oni grafički prikazuju, navedeno je na slici u nastavku.

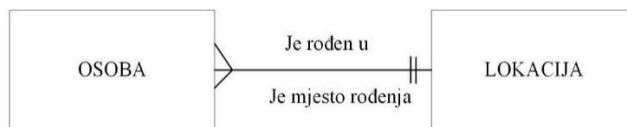


*Slika 7.6 Chenova notacija – Peter Chen: Entity Relationship Model – ERM (1976).*

Vidimo da su entiteti (osoba i lokacija) prikazani pravokutnicima, a atributi se nalaze u krugovima izvan entiteta (OIB). Veze su prikazane imenicama u rombovima, i to u samo jednom smjeru, a kardinalnost se prikazuje brojem i/ili slovom pored entiteta. Ključni atributi nisu posebno naznačeni.

Premda postoji više verzija notacija IE, ne postoji standard. Ipak, bez obzira o kojoj se verziji radi, veliki broj alata za modeliranje podržava ovu notaciju i ona je jedna od najpopularnijih za dizajniranje baze podataka (Halpin, 2011). Notacija IE je jednostavna i laka za čitanje, a često se naziva i vranina noga – engl. *crow's foot* (Murray i Guimaraes, 2009). Riječ je o notaciji koja vezu prikazuje linijom, a ne romбом, kao što je to prethodno bilo kod Chenove notacije.

Primjer notacije vidimo u nastavku.

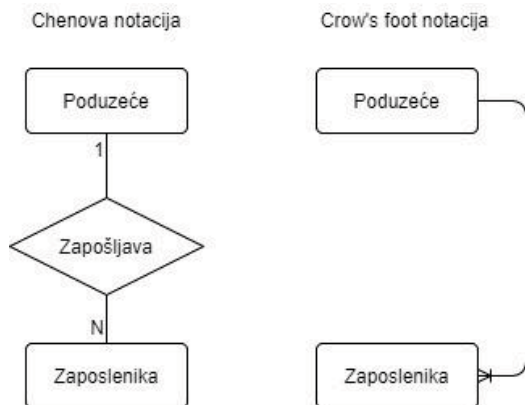


*Slika 7.7 Martinova notacija – James Martin: Information Engineering - IE (1980-ih).*

Entiteti su prikazani pravokutnicima (OSOBA i LOKACIJA), a notacija ne omogućuje identifikaciju atributa. Pretpostavka je da će se atributi modelirati drugim dijagramom ili da će biti opisani u popratnoj dokumentaciji, a veza je prezentirana glagolom, i to na liniji (*je rođen u* i *je mjesto rođenja*).

Iz navedenih primjera vidljivo je kako se se korištenjem različitih notacija na različiti način modeliraju isti entiteti. Primjer koji uspoređuje spomenute notacije, Chenovu i

Martinovu (vranina noga), prikazan je u nastavku. Riječ je o entitetima *poduzeće* i *zaposlenik*.



Slika 7.8 Usporedba Chenove i Martinove notacije (vranina noga)

### 7.3 Relacijski model podataka

Model EVA je korisnicima najrazumljiviji te je stoga prvi u slijedu prikupljanja i modeliranja zahtjeva o podacima. Slijedi korak u kojem je utvrđeni entitet i njegove veze potrebno složiti u jednu optimalnu cjelinu korištenjem matematičkih metoda. Iako je ovaj postupak dosta kompleksniji, u ovoj knjizi ćemo pokazati samo *osnove relacijskih baza podataka* kako bi se shvatila bit problema.

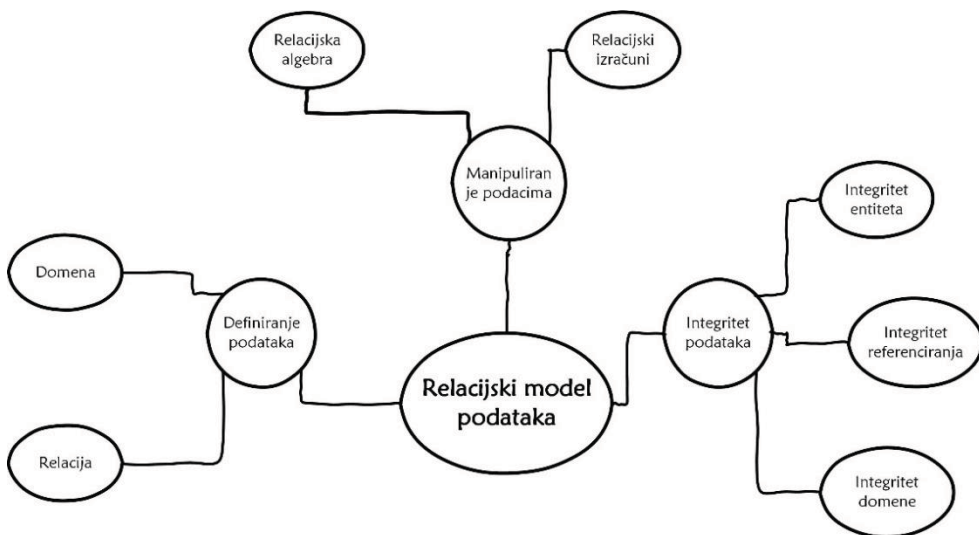
#### 7.3.1 Osnovne komponente relacijskog modela

Relacijski model koristi matematičku algebru za definiranje ovisnosti (strukture) među podacima. Na taj način može postići visoku razinu neovisnosti podataka, bez da se istovremeno izgube semantičke informacije o stvarnom stanju (Chen, 1976). Zahvaljujući jednostavnosti relacijskog modela bilo je moguće proširiti opseg baze podataka te je stoga razvoj relacijskog modela bio samo okidač za nešto više (Rob & Coronel, 2009). Veći broj transakcija tražio je kompleksnije strukture podataka i kvalitetnije alate za dizajn takvih baza. Upravo povećanje količine podataka zahtijeva istraživanje novih struktura za oblikovanje podataka.

Relacijski model je nastao zahvaljujući jednom jedinom čovjeku, E.F. Coodu, i to davne 1970. (Beynon-Davies, 2004; Chen, 2002). U relacijskom modelu podataka postoji samo jedan oblik strukture podataka, a to je relacija (Beynon-Davies, 2004). Pojednostavljeno rečeno, relacija je sve ono što nije funkcija. Iz matematike je poznato da je osnovna karakteristika funkcije da se jedan element iz polaznog skupa (domene) može preslikati u samo jedan element nekog drugog skupa (kodomene). To upućuje na zaključak da relacija može preslikavati jedan element domene u više



elemenata nekog drugog skupa. Relacija u matematici znači određeni odnos između zadanih objekata. Na temelju toga smo definirali polazni okvir relacijskog modela jer se relacijske baze podataka baziraju na relacijama. U informatičkom smislu, jedan podatak se nalazi u odnosu s drugim podacima. Upravo u traženju odgovora na pitanje o odnosu podataka koristimo relacijsku algebru. Iz tog razloga kažemo da se u ovom slučaju logika modeliranja podataka temelji na relacijskoj algebri (Beynon-Davies, 2004; Xiaojuan, 2007; Coronel, Morris, & Rob, 2011). Iako je relacijski model sličan modelu EVA, entiteti i veze u relacijskom modelu moraju poštovati striktna pravila (Xiaojuan, 2007).



Slika 7.9 Relacijski model (Beynon-Davies, 2004)

Osnova karakteristika (prednost) relacijskog modela je da se korisnici mogu fokusirati na samu logiku podataka, a ne na fizičku strukturu (Coronel, Morris, & Rob, 2011). Osnovne komponente relacijskog modela podataka su (Tkalac, 1993):

- skup objekata koje koristimo kao osnovne elemente prilikom izgradnje modela baze podataka (entiteti),
- skup operacija koje možemo izvoditi nad objektima modela,
- skup pravila integriteta baze podataka.

Uzevši u obzir sličnosti modela EVA i relacijskog modela, osnove relacijskog modela bit će objašnjene pomoću modela EVA.

Tablica 7.1 Model EVA i relacijski model (Chen, 2002)

Model EVA	Relacijski model
Instanca entiteta	Redak
Entitet	Relacijska tablica (relacija)
Dodaje semantičke informacije podataka, odnosno dodatno se pojašnjava značenje	Nedostaje semantičkih informacija (nema detaljnijih opisa podataka)
Koristi konstrukciju matematičke relacije za opis strukture entiteta	Koristi konstrukciju matematičke relacije za opis strukture vrijednosti podataka

### 7.3.2 Relacijska shema i relacija

Pojedinačni podatak je najmanji objekt relacijskog modela, što znači da ga nije moguće dalje rastavljati na dijelove (Tkalac, 1993). Pojam entiteta u relacijskom modelu isti je kao i u modelu EVA. Podatke o entitetu (njegova svojstva) opisuju atributi koji, kao i kod modela EVA, imaju svoje vrijednosti. Te vrijednosti zovu se vrijednosti atributa. Vrijednosti koje određeni atribut može poprimiti pripadaju jednom skupu podataka, tj. domeni. To znači da je atribut element informacije kojime je jednoznačno određena vrsta svojstva promatranog entiteta. Relacijski model razlikuje jednostavne i sastavljene attribute. Vrijednost jednostavnog atributa je pojedinačni podatak, dok je vrijednost sastavljenog atributa uređena  $n$ -torka pojedinačnih podataka.

Svaka organizacija kroz svoje poslovanje „provlači“ veliki broj različitih entiteta koji imaju svoje attribute. Prema logici stvari, vrijednosti svakog atributa moguće je ograničiti na određeni skup. Tako npr. ime ili prezime može biti sastavljeno kao određeni niz slova, dok OIB mora poprimiti cjelobrojne vrijednosti. Upravo ti skupovi predstavljaju domenu promatranog atributa. Nadalje, svakom OIB-u može biti pridruženo neko ime i obratno. Prethodni odnos u matematičkom kontekstu možemo definirati kao Kartezijev produkt dva skupa – „Ime“ i „OIB“. Rezultat Kartezijeva produkta ta dva skupa bit će skup uređenih parova svih imena i svih OIB-ova. Normalno, jednom OIB-u može se pridružiti samo jedno ime. Iz toga razloga se govori o podskupu Kartezijeva produkta spomenutih skupova. Ovakvo pojednostavljenje se poopćuje do potrebnog broja svih atributa različitih entiteta za modeliranje podataka organizacije. Taj broj je velik, ali konačan.

Ograničimo li se na jedan entitet s  $n$  atributa, relacija je podskup Kartezijeva produkta svih domena (Silberschatz, Korth, & Sudarshan, 2001). Kako je navedeno, svaki atribut ima svoju domenu, što znači da svaka tablica, koja se sastoji od  $n$  atributa, mora biti podskup (Silberschatz, Korth, & Sudarshan, 2001):

$$D_1 \times D_2 \times D_3 \times \dots \times D_{n-1} \times D_n$$

Kako bi se stvar pojednostavila, uveden je pojam relacijske sheme koja je reprezentant relacije. To je moguće shvatiti kao naslovni redak neke tablice. U nastavku je pružen primjer relacijske sheme STUDENT.

STUDENT (STU\_NUM, STU\_IME, STU\_PREZ, STU\_DATUM, STUD\_SR\_OCJ, ODJEL\_SIFRA, STU\_TEL, PROF\_BROJ)

To znači da je relacija „STUDENT“ podskup Kartezijeva produkta domena svih njegovih atributa. Pojednostavljeno rečeno, relacija „STUDENT“ može biti kombinacija svih vrijednosti navedenih atributa.

Primjerice, tablica (relacija) STUDENT izgleda ovako:

Tablica 7.2 Relacija STUDENT

STUDENT							
STU_NUM	STU_IME	STU_PREZ	STU_DATUM	STUD_SR_OCJ	ODJEL_SIFRA	STU_TEL	PROF_BROJ
M	E	Z	M	J	A		J
1	Mate	Matic	21.9.1990.	4	201	98111111 1	205
2	Ante	Antic	10.12.1990.	5	200	97222222 2	222
3	Iva	Ivic	23.3.1991.	5	201	3214567 8	228
4	Stipe	Stipic	25.5.1989.	3	203	2345619 8	222
5	Ana	Anic	31.12.1989.	2	201	2345678 9	205
6	Vesna	Vesnic	22.4.1990.	3	200	2345678 9	199

Tablica STUDENT sadrži sljedeće atribute:

1. STU\_NUM – jedinstveni broj studenta
2. STU\_IME – ime studenta
3. STU\_PREZ – prezime studenta
4. STU\_DATUM – datum rođenja studenta
5. STUD\_SR\_OCJ – srednja ocjena studenta
6. ODJEL\_SIFRA – šifra odjela
7. STU\_TEL – broj telefona studenta

### 8. PROF\_BROJ – broj profesora dodijeljenog studentu

Matematički koncept relacije prikazan je dvodimenzionalnom tablicom koja se sastoji od redaka i stupaca (Silberschatz, Korth, & Sudarshan, 2001). Svaki redak u tablici prikazuje odnose među vrijednostima u tom retku, a kako je tablica skup tih odnosa (redaka), koncept relacije preklapa se s konceptom tablice.

Tablica (relacija) ima svoja striktna pravila (Beynon-Davies, 2004; Xiaojuan, 2007; Coronel, Morris, & Rob, 2011):

1. svaka tablica u bazi podataka mora imati jedinstveno ime;
2. svaka tablica mora imati primarni ključ (jedan ili više atributa koji identificiraju retke);
3. redovi tablice predstavljaju instance entiteta –  $n$ -torke (po jedan objekt iz skupa entiteta);
4. stupci tablice predstavljaju atribute entiteta;
5. svaki stupac (atribut) unutar iste tablice mora imati jedinstveno ime;
6. unosi unutar istog stupca moraju biti istog tipa (domena atributa);
7. redosljed stupaca u tablici nije bitan;
8. svaki red u tablici mora biti jedinstven, nije dozvoljeno imati dvostruke retke;
9. redosljed redova u tablici nije bitan;
10. svaka ćelija, bilo da se radi o stupcu ili retku, mora sadržavati jednu jedinu vrijednost.

Tablica ili relacija sadrži skup objekata, odnosno grupu pojava entiteta (engl. *entity occurrences*).

#### 7.3.3 Ključevi

Već je prije istaknuto kako je važno da svaka tablica ima primarni ključ po kojem bi se  $n$ -torke, odnosno redovi, međusobno razlikovale (Coronel, Morris, & Rob, 2011). Primarni ključ može biti bilo koji atribut koji ima jedinstvenu vrijednost za svaki redak i ne smije biti 0. Primjerice, OIB ima jedinstvenu vrijednost za svakog čovjeka, pa on može biti primarni ključ.

U prikazanoj tablici STUDENT primarni ključ je atribut STU\_NUM jer je to broj koji je jedinstven za svakog studenta (npr. matični broj). Još jedna funkcija primarnog ključa do izražaja dolazi prilikom uspostavljanja veza s drugim tablicama. Primarni ključ jedne tablice može biti strani ključ u više drugih tablica s kojima je povezana.

Uloga primarnog ključa se temelji na konceptu determinante, što znači da, ako poznamo vrijednost atributa  $A$ , možemo determinirati (odrediti) i vrijednost

atributa B (Coronel, Morris, & Rob, 2011). Primjerice, ako znamo šifru studenta (STU\_NUM), tada možemo znati i sve ostale podatke za tog studenta (ime, prezime, broj telefona itd.). S druge strane, ako znamo vrijednost nekog drugog atributa koji nije primarni ključ, kao što je ime studenta (STU\_IME), tada ne možemo odrediti vrijednost drugih atributa jer su velike šanse da postoji više studenata koji imaju isto ime.

Vrijedi sljedeće (Coronel, Morris & Rob, 2011):

- *Atribut B funkcijski ovisi o atributu A, ako svaka vrijednost stupca A determinira (određuje) samo jednu vrijednost stupca B.*
- Tada je atribut A primarni ključ u tablici, a atribut B jedan od ostalih atributa.
- Postoji više vrsta ključeva, a njihove karakteristike sadržane su u tablici ispod.

*Tablica 7.3. Vrste ključeva i njihove definicije*

Vrsta ključa	Definicija
Superključ	Atribut (ili više atributa) pomoću kojeg se identificiraju n-torke.
Ključ kandidat	Minimalno superključ koji pritom ne sadrži attribute koji su sami sebi superključevi.
Primarni ključ	Ključ kandidat koji je odabran da jedinstveno identificira n-torke. Vrijednost ne smije bit nula.
Sekundarni ključ	Atribut (ili više atributa) koji se koristi isključivo kod dohvaćanja podataka.
Strani ključ	Atribut (ili više atributa) čije vrijednosti odgovaraju vrijednostima primarnog ključa druge tablice.

### 7.3.4 Relacijska algebra

Kako bi se u kontekstu matematike neki skup definirao kao upotrebljiva struktura, potrebno je definirati skup pravila koji je u skladu s određenim zakonitostima. Na početku smo spomenuli kako se relacijski model temelji na relacijskog algebr (Beynon-Davies, 2004; Coronel, Morris, & Rob, 2011). Ne ulazeći dalje u matematički pojam algebre, prihvatit ćemo da je to neka „lijepa“ matematička struktura koja ima definiran skup elemenata i operacije nad tim skupom elemenata. Elementi su definirani u prethodnom odjeljku, a u nastavku ćemo definirati operacije.

Relacijska algebra definira kako manipulirati podacima iz tablice koristeći **osam** relacijskih operacija. Relacijske operacije su (Coronel, Morris & Rob, 2011):

1. SELECT. Operacija koja prikazuje samo one retke ( $n$ -torke) koji zadovoljavaju dane uvjete. Drugim riječima, prikazuje horizontalni podskup tablice.
2. PROJECT. Operacija koja prikazuje vrijednosti samo odabranih atributa (stupaca), tj. vertikalni podskup tablice.
3. UNION. Spaja dvije tablice i pritom ne uključuje dvostruke retke. Da bi se ova operacija izvršila, te dvije tablice moraju imati isti broj atributa i ti atributi moraju biti jednaki (moraju imati jednake domene).
4. INTERSELECT. Operacija koja prikazuje samo  $n$ -torke (retke) koji su jednaki u dvije tablice. Da bi se operacija izvršila, tablice moraju zadovoljiti jednake uvjete kao i kod korištenja operacije UNION.
5. DIFFERENCE. Da bi se ova operacija izvršila, dvije tablice moraju zadovoljiti uvjete kao i kod operacije UNION, a rezultat korištenja ove operacije su  $n$ -torke (redci) koji nisu zajednički dvjema tablicama (rezultat je suprotan operaciji INTERSELECT).
6. PRODUCT. Prikazuje sve moguće parove redaka dviju tablica – Kartezijev produkt. Primjerice, ako jedna tablica ima šest redaka, a druga tri retka, rezultat će bit tablica od 18 ( $6 \times 3$ ) redaka.
7. JOIN. Operacija koja omogućuje kombiniranje informacija iz dvije ili više tablica. Tablice se spajaju pomoću jednakih (zajedničkih) atributa.
8. DIVIDE. Operacija koja za djelitelja koristi tablicu koja sadrži samo jedan stupac, a za djeljenika tablicu s dva stupca. Te dvije tablice moraju imati jedan zajednički stupac.

SQL (engl. *Structured Query Language*) su naredbe kojima je moguće realizirati navedene operacije relacijske algebre.

### 7.3.5 Integritet

Riječ integritet potiče od latinske riječi *integritas*, što znači cjelovitost. Kada se priča o integritetu relacijskog modela podataka, jako je bitno spomenuti dva aspekta integriteta (Beynon-Davies, 2004; Coronel, Morris, & Rob, 2011):

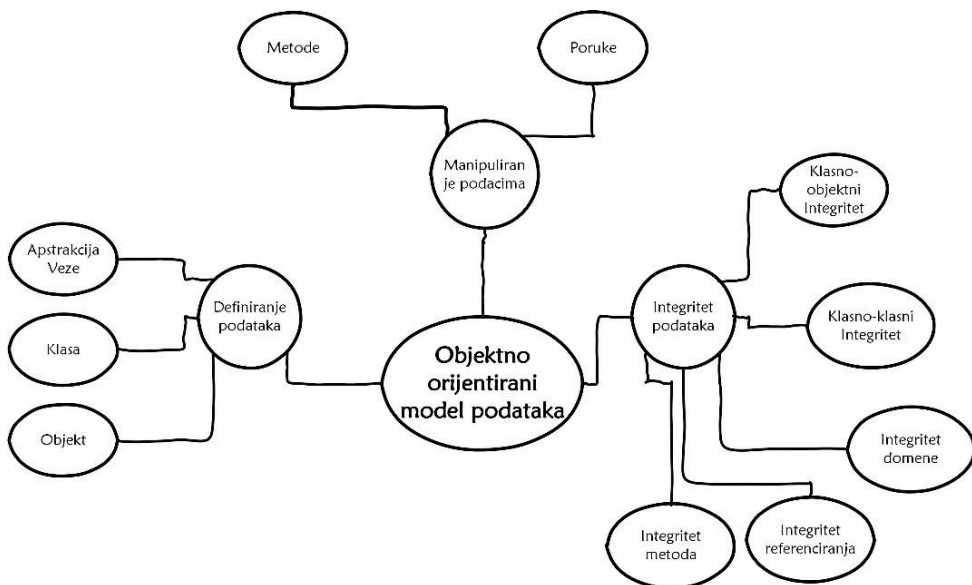
- Integritet entiteta tiče se primarnih ključeva, a pravilo kaže da svaka tablica mora imati jedan primarni ključ koji mora biti jedinstven i različit od nule.
- Integritet referenciranja ovo pravilo se odnosi na strane ključeve, a kaže da vrijednost stranog ključa može (a) biti null vrijednost ili (b) odgovarati vrijednosti primarnog ključa povezane tablice.

Na taj način stvoreni su matematički preduvjeti za razvoj cijele teorije kojom se definiraju odnosi među podacima. Daljnji koraci u optimiranju strukture, tj. oblikovanju baza podataka odnose se na teoriju ovisnosti. Codd je ugradio funkcijske ovisnosti u koncept relacijskog modela (Codd, 1970). Deobel i Casey, a poslije njih i Armstrong, su izučavali problem izvođenja funkcijskih ovisnosti (Armstrong, 1974; Delobel & Casey, 1973). Kao rezultat toga definirane su prva, druga, treća i Boyce-Coddova normalna forma, a na kraju, pomoću višeznačnih ovisnosti, i peta normalna forma.

#### 7.4 Objektno orijentirani model podataka

Objektno orijentirano modeliranje (OOM) pojavilo se 1980-ih, a zbog mogućnosti modeliranja kompleksnih sustava postalo je jako popularno (Hoffer, Venkataraman & Topi, 2013). OO nije proširenje nekih od prethodnih modela već se radi o poprilično različitom pristupu definiranja i korištenja entiteta (Coronel, Morris & Rob, 2011).

Za svako prikazivanje svijeta oko nas potrebno je modelirati (1) podatke, (2) procese, ali i (3) ponašanje podataka (Hoffer, Venkataraman & Topi, 2013). Upravo je ponašanje ono što se dosadašnjim modelima podataka nije moglo prikazati. Primjerice, model EVA ne dopušta bilježenje ponašanja nego samo mogućnost modeliranja podatkovnih potreba organizacije.



Slika 7.10 Objektno orijentirani model podataka (Beynon-Davies, 2004)

OO se temelji na sljedećim pretpostavkama (Coronel, Morris, & Rob, 2011):

1. Objekt predstavlja apstrakciju realnog svijeta, stoga se može smatrati ekvivalentom instanci entiteta iz modela EVA.
2. Atributi opisuju svojstva objekta.
3. Objekti koji imaju zajednička svojstva (atribute) pripadaju istoj klasi.
4. Klase se organiziraju pomoću klasne hijerarhije. To rezultira stablastom strukturom, tj. moguće ih je prikazati u obliku stabla.
5. Nasljeđe je mogućnost da objekt u klasnoj hijerarhiji naslijedi svojstva (atribute) i ponašanje (metode) klase iznad.

U načelu, OO podatke i njihove veze stavlja u jedinstvenu strukturu – objekt (Coronel, Morris, & Rob, 2011), a može se reći i kako OO zapravo predstavlja kombinaciju objektno orijentiranog pristupa i relacijskog modela podataka. Analiza objektno orijentiranog pristupa u poglavlju 5.2 krenula je od objekta prema klasama. U kontekstu modeliranja podataka, a kao nastavak na prethodne odjeljke, analiza postupka OO-a krenut će od klasa, tj. deduktivno.

### 7.4.1 Klasa u kontekstu modeliranja podataka

Klasa opisuje stvari na apstraktan način. Ona se koristi za definiranje svojstva, metoda i veza skupnih objekata. Klase služe kao predlošci za kreiranje objekata (nacrti). Ako se prisjetimo procedura u tradicionalnom programiranju, onda možemo reći kako su metode jednake procedurama jer one opisuju ponašanje objekta.

Klasa je kolekcija objekata koji imaju jednake attribute (svojstva) i ponašanja (metode), odnosno predstavlja skup objekata grupiranih po svojoj sličnosti, tj. grupu sličnih objekata (Silberschatz, Korth, & Sudarshan, 2001). Primjerice, želimo li studenta predstaviti klasom, tada bismo rekli da su njegovi atributi *ime*, *prezime*, *matični broj*, *datum rođenja* i sl., a pod ponašanjem studenta podrazumijevalo bi *polaganje ispita*, *prisustvovanje na vježbama*, *upisivanje nove godine* itd. Ako želimo da klasa „Student„ postane objekt, tada atributi moraju poprimiti vrijednosti. To je razlog zašto kažemo da su objekti instance klase.

Može se reći kako klasa definira namjeru neke objektno orijentirane baze podataka, a objekti određuju opseg objektno orijentirane baze (Beynon-Davies, 2004).

Svi objekti koji pripadaju jednoj klasi imaju ista (1) svojstva, (2) metode i (3) sučelja poruke.



Drugim riječima, klase su tip entiteta koji ima jasno definiranu ulogu u primjeni te organizacija želi održati stanje (atribute i veze), identitet i ponašanje (i reakcije) (Hoffer, Venkataraman, & Topi, 2013).

#### 7.4.2 Instanciranje

Objekt kreiramo iz neke klase, a iz jedne klase možemo stvoriti više objekata. Proces stvaranja objekta naziva se instanciranje, a kreirani objekt je instanca klase (Bergholt, i dr., 1998).

U stvarnom svijetu mnogi objekti su slični, odnosno imaju zajednička svojstva i ponašanje. Ipak, svaki objekt ima svoj identitet i svoje jedinstvene vrijednosti (unutar) zajedničkih svojstava. Unatoč jedinstvenosti identiteta i vrijednosti svakog objekta, smislenije je opisivati objekte u grupama (Pavković, 2000). Objekti koji pripadaju istoj klasi imaju jednako ponašanje i jednake attribute (Hoffer, Venkataraman, & Topi, 2013), što je već prije navedeno.

Metafora koja najbolje opisuje karakteristike objekta je gen jer sadrži mnoštvo informacija potrebnih za ponašanje na odgovarajući način u određenom okruženju (Beynon-Davies, 2004). Drugim riječima, gen se promatra kao jedna cjelina koja sadrži mnoštvo informacija o ponašanju, jednako kao što je to objekt. Njega isto možemo promatrati kao cjelinu koja sa sobom nosi niz informacija.

Analogijom prema prethodnom poglavlju, tip entiteta u modelu EVA može se prikazati kao klasa, a instanca entiteta kao objekt. Međutim, što nedostaje prema objektnom pristupu? Objekt mora opisivati i ponašanje, o čemu je bilo riječi u petom poglavlju. Objekt sadrži informacije – ne samo samog objekta nego i informacije o vezama s drugim objektima. To su razlozi zašto se objektno orijentirani model podataka još naziva i semantički model podataka (dodano mu je značenje).

Kao što je već spomenuto, osnovni elementi objekata su atributi (svojstva) i operacije (metode). Atribut je informacijski detalj svojstven objektu. Ovisno o konkretnom programskom jeziku, atributi se nazivaju i varijablama ili svojstvenim poljima. Operacija je funkcionalni detalj svojstven objektu i spremljen kao dio objekta. Za operacije se koriste i nazivi svojstvena funkcija (engl. *member function*) ili metoda. Evidentno je da ne postoji jedinstvena nomenklatura, a u ovoj knjizi koristit će se pojmovi svojstva i metode.

SVOJSTVO = ATRIBUT = SVOJSTVENO POLJE

METODA = OPERACIJA = SVOJSTVENA FUNKCIJA

Budući da objekt sadrži podatke te različite vrste veza i metoda, on postaje osnovni strukturni element (Coronel, Morris, & Rob, 2011). Ponašanje objekta je skup metoda i odgovora na poruke (Bergholt et. al., 1998). Podatci su sadržani u svojstvima objekta, a metode objekta definiraju procedure. Metode se aktiviraju porukama koje kolaju među objektima (Beynon-Davies, 2004). Poruke sadrže naziv metode, identifikaciju objekta i parametre metode. Poruke na koje objekt reagira mogu imati nula, jedan ili više parametara (Silberschatz, Korth, & Sudarshan, 2001). Drugim riječima, porukama pozivamo metodu.

Metoda je funkcija ili usluga koju osigurava instanca klase, tj. objekt (Hoffer, Venkataraman, & Topi, 2013). Drugi objekti mogu pristupiti i koristiti informacije objekta samo uz pomoć metoda.

Postoje četiri tipa metoda (Hoffer, Venkataraman, & Topi, 2013; Beynon-Davies, 2004):

1. Konstruktor (konstruktivna metoda) je metoda koja kreira novu instancu klase.
2. Upit (pristupna metoda) je metoda koja pristupa stanju objekta, ali ga ne mijenja.
3. Ažuriranje (metoda transformiranja) je metoda kojom se mijenja stanje objekta.
4. Opseg klase je metoda koja se provodi na razini klase, a ne objekta.

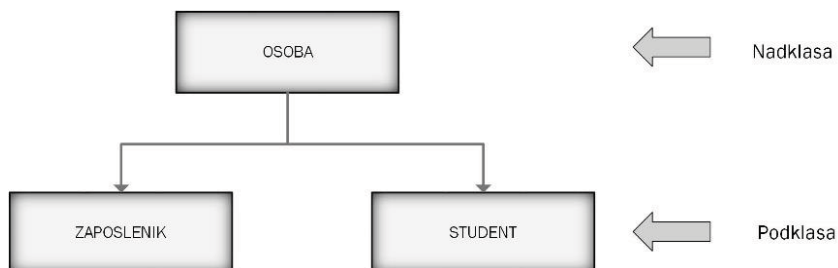
### 7.4.3 Nasljeđivanje

Tri su moguća tipa veze ili relacije, a prethodno su spomenuta u petom poglavlju 5 (Beynon-Davies, 2004; Hoffer, Venkataraman, & Topi, 2013):

- (1) asocijacija, kao ekvivalent vezama kod modela EVA (pogledati poglavlje 7.2.2);
- (2) generalizacija;
- (3) agregacija.

Generalizacija i agregacija se zajednički zovu mehanizmi apstrakcije jer se pomoću njih može upravljati prikazom detalja, odnosno omogućuju kreiranje hijerarhije (Beynon-Davies, 2004). Apstrakciju osigurava i *enkapsulacija* (učahurivanje), pomoću koje se zanemaruju detalji (Bergholt, i dr., 1998). Svi atributi i operacije su u potpunosti skriveni od vanjskog svijeta (učahureni).

Generalizacija je proces definiranja klasa (natklasa) koje su općenitije od klasa koje imaju slične karakteristike (potklase), odnosno izdvajanje zajedničkih značajki klasa i zanemarivanje detalja koji međusobno razlikuju te klase (Fernandez & Le Teno, 1997; Beynon-Davies, 2004). Oprečan pojam generalizaciji je specijalizacija (slika 7.11).

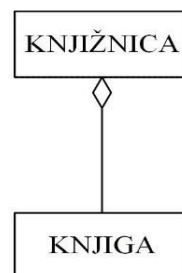


Slika 7.11 Klasna hijerarhija

Posljedica generalizacije je *nasljeđivanje* jer prilikom takve veze svaka potklasa nasljeđuje atribute, asocijacije i metode natklase. Potklase nasljeđuju karakteristike svoje natklase. Nasljeđe je jako snažan alat jer podržava, odnosno omogućuje ponovno korištenje ili prenamjenu modela (Hoffer, Venkataraman, & Topi, 2013; Bergholt et. al., 1998). Nasljeđivanje također osigurava temelj za realizaciju *polimorfizma*, budući da sustav može omogućiti prepoznavanje objekata po klasi u klasnoj hijerarhiji (gdje se događa nasljeđivanje).

Koncept *polimorfizma* omogućuje primjenjivanje jedne metode na više različitih načina kod više različitih klasa. Skupa sa konceptom *enkapsulacija* i *nasljeđivanjem* to troje predstavlja programerima kvalitetan alat za razvoj složenih, robusnih, fleksibilnih i održivih sustava (Hoffer, Venkataraman, & Topi, 2013).

Agregacija je tip veze među objektima u kojem se smatra da je jedan objekt sastavljen od drugih objekata (Fernandez & Le Teno, 1997), odnosno taj tip veze nastaje između cjeline i pripadajućih dijelova cjeline (Beynon-Davies, 2004). Oprečan pojam agregaciji je dekompozicija.



Slika 7.12 prikazuje primjer agregacije. Tako vidimo da je objekt knjižnica sastavljen od objekta knjiga, odnosno knjižnica je cjelina, a knjiga pripadajući dio.

Slika 7.12 Prikaz agregacije

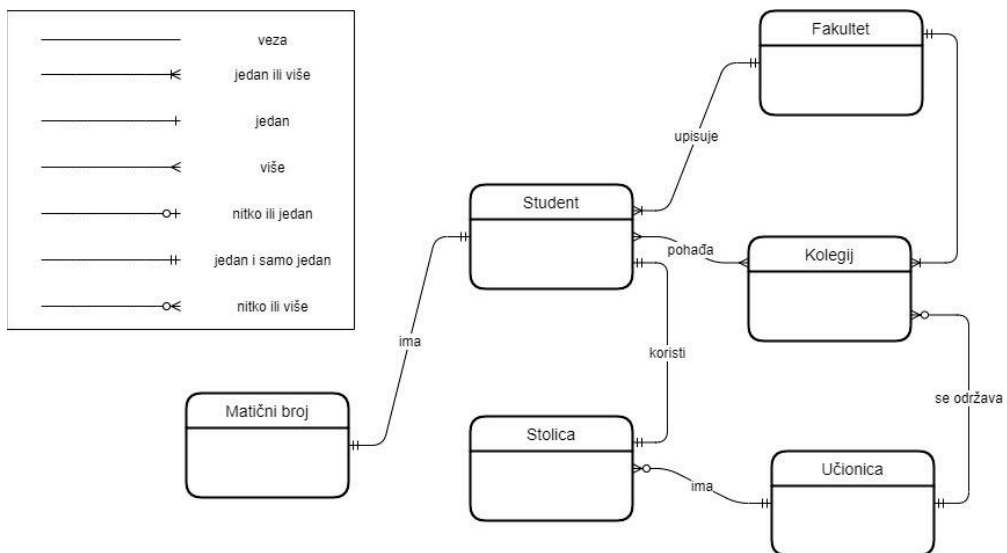
## 7.5 Dijagram modela podataka

Već je spomenuto kako su modeli podataka EVA prikazani kao dijagram (Beynon-Davies, 2004). Dijagram se može koristiti kao alat za provjeru točnosti poslovnih pravila i veza. Ovisno o notaciji koja je korištena, prilikom modeliranja mogu se namjerno izostaviti neki detalji (Lurton, 2011).

Ranije spomenuta notacija pod imenom vranina noga (engl. *crow's foot*) najčešće se koristi kod izrade dijagrama EVA, zbog svoje intuitivnosti. Takvi dijagrami prikazuju entitete kao pravokutnike, a veze kao linije između pravokutnika. Način na koji završavaju te linije označuje prethodno opisanu kardinalnost veze. Model podataka uz pomoć spomenute notacije detaljno je prikazan u nastavku (slika 7.13).

Primjerice, prateći notaciju vidimo kako student upisuje samo jedan fakultet, dok s druge strane fakultet upisuje jednog ili više studenata. Svaki student ima jedinstveni matični broj te veza upućuje kako samo jedan student ima samo jedan matični broj i obrnuto (jedan matični broj pripada samo jednom studentu). Specifičan kolegij se održava samo na jednom fakultetu, a fakultet može imati jedan ili više različitih kolegija. Što se tiče upisa kolegija, više studenata može pohađati više kolegija i obrnuto.

Opisano je prikazano na slici dijagrama EVA u nastavku, koji ujedno sadrži i značenje prikazanih veza.



Slika 7.13 Dijagram EVA

### 7.5.1 Pogled na podatke visoke razine

Razmotrimo jednostavni proces nabave. Svako poduzeće koje se bavi tim poslom ima veliki broj klijenata, narudžbi i proizvoda. Sve te podatke potrebno je organizirati i zapisati u bazi podataka na način da se mogu lako pohranjivati, čitati i mijenjati te da je istima lako upravljati.

Prilikom organizacije podataka potrebno je jasno prikazati odnose svih elemenata koje možemo imati u bazi (slika 7.14). Primjerice, datum računa, ime i adresa kupca, količina kupljenog proizvoda i sl. Pod pojmom organizacije podataka podrazumijeva se prikaz odnosa među elementima i njihove međusobne suradnje.



*Slika 7.14 Prikaz elementata baze podataka*

U tu svrhu možemo koristiti dijagrame EVA. Međutim, prije izrade dijagrama objasnimo osnovne komponente dijagrama.

Prva osnovna komponenta je entitet, koji smo objasnili u prethodnom poglavlju. Kao što smo već spomenuli, svaki entitet ima svoje ime i atribute u skladu s pojmom koji opisuje (slika 7.15).



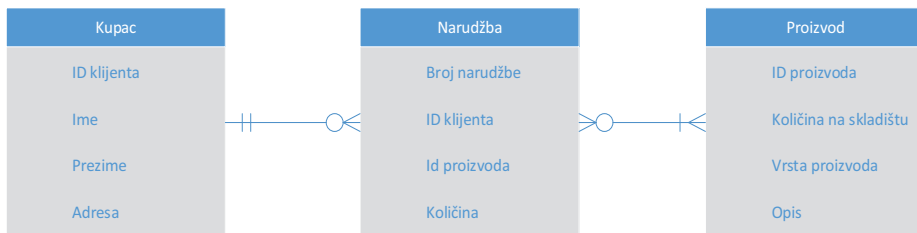
*Slika 7.15 Prikaz entiteta s atributima*

U terminima baze podataka, može se reći da su entiteti predstavljeni redcima, a atributi stupcima (slika 7.16).

	Atribut 1	Atribut 2	Atribut 3	Atribut 4	Atribut 5	Atribut 6	Atribut 7
Entitet 1							
Entitet 2							
Entitet 3							
Entitet 4							
Entitet 5							

Slika 7.16 Tablični prikaz entiteta i atributa

U našem primjeru s prethodne slike imamo više različitih entiteta s različitim skupom atributa, što znači da ćemo trebati više tablica nego na prethodnoj slici. Sljedeća važna komponenta je veza među entitetima, čime se opisuje njihova interakcija (slika 7.17).



Slika 7.17 Prikaz entiteta, atributa i veza

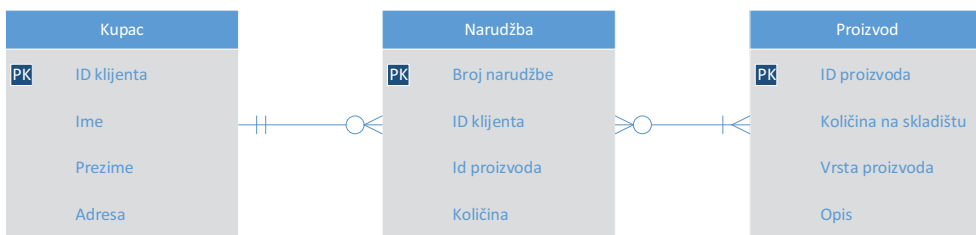
Kako pročitati prethodnu sliku? Kao što smo već prije spominjali, simboli na početku i kraju spojne crte označavaju brojnost veze. Pitanja koja je potrebno postaviti kako bi se prilikom modeliranja veza odredila brojnost odgovaraju logici realnog sustava. Tako je npr. za određivanje veze između entiteta KLIJENT i entiteta NARUDŽBA potrebno postaviti pitanje *Koliki je minimalni i maksimalni broj narudžbi koje klijent može imati*, odnosno *Koji je minimalan i maksimalan broj klijenata koje narudžba može imati*. Jedan klijent može imati nijednu ili više narudžbi, odnosno narudžba može imati jednog i samo jednog klijenta. Na isti način – jedna narudžba može imati jedan ili više proizvoda, odnosno proizvod može biti dio nijedne ili više narudžbi. Na taj način zapravo opisujemo poslovna pravila koja vrijede među entitetima.

### 7.5.2 Detaljni pogled na podatke

Detaljnim pogledom na podatke veze među entitetima objašnjavaju se na razini atributa promatranih različitih entiteta. Iz tog razloga uvodimo pojmove primarni i strani ključ. Prisjetimo se da svaki nacrtani entitet predstavlja tablicu u bazi podataka. Tako primjerice entitet KLIJENT predstavlja kupce, koji su u našoj bazi podataka opisani svojim atributima (ID kupca, ime, prezime, adresa itd.). Kako bi smo jednostavno odredili svakog od kupaca, potrebno je imati njegov identifikator.

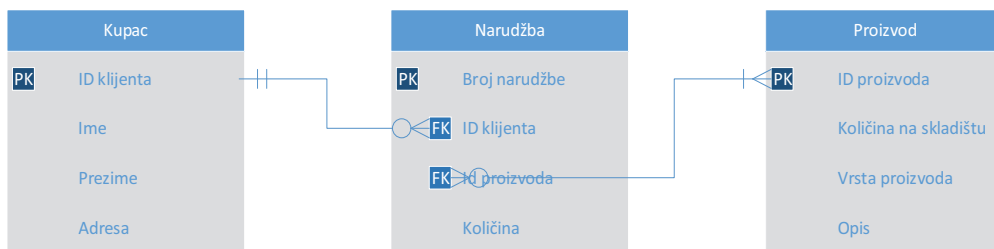
To nije ime i prezime jer u realnom svijetu može biti jako puno osoba koje imaju isto ime i prezime. Je li to OIB? Pa sam naziv *osobni identifikacijski broj* ukazuje da jest. Međutim, svi stanovnici Hrvatske ne moraju biti kupci/klijenti promatranog primjera. Iz tog razloga se često u bazama podataka uvodi identifikator, što je u našem primjeru „ID kupca“. Upravo taj identifikator zovemo primarni ključ. Primarni ključ je atribut koji jednoznačno određuje zapis o promatranom tablici. Možemo zaključiti da svaki entitet treba imati jedan takav primarni ključ. Kao što je prije rečeno, primarni ključ mora biti nepromjenjiv i jedinstven te ne može biti prazno polje (nul-vrijednost).

Jedan svakodnevni primjer je otvaranje korisničkog računa. Kad jednom otvorite korisnički račun (Facebook, Gmail, Instagram i sl.), više ga ne možete promijeniti. Jednako tako, možda ste na nekom sustavu pokušali otvoriti novo korisničko ime, ali sustav to nije dozvolio nego je poslao poruku da takvo korisničko ime već postoji. To je iz razloga što se korisničko ime koristi kao primarni ključ – možete samo otvoriti novi. Naime, e-pošta je jedinstven i nepromjenjiv primjer koji ne može poprimiti nul-vrijednost, što znači da ima sva svojstva primarnog ključa. Kao što vidimo, primarni ključ ne mora biti brojka. Iako jedna osoba može imati više adresa e-pošte, sustav je tretira kao različiti korisnički profil. Vratimo se sada našem primjeru i uvedimo primarne ključeve za sve entitete (slika 7.18).



Slika 7.18 Identifikacija primarnih ključeva (PK)

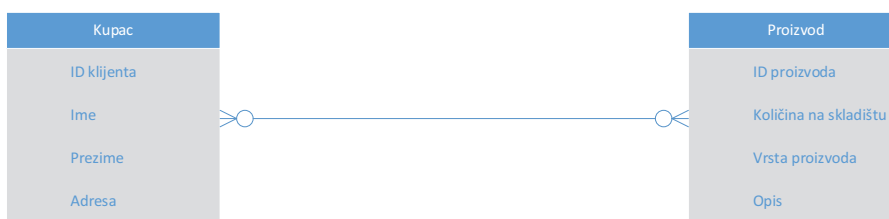
Objasnilo sada što je to strani ključ. Strani ključ je isto što i primarni ključ, samo što je lociran u drugom entitetu. Pogledamo li sliku 7.18, isti atribut (ID klijenta) nalazi se u opisu entiteta Kupac i Narudžba. Na taj način za svaku narudžbu identificiranu po atributu „Broj narudžbe“ jednostavno referenciramo zapis o kupcu. Pritom je vrijednost atributa „ID klijenta“ entiteta „Narudžba“ jednaka vrijednosti istog atributa entiteta „Kupac“. Isto vrijedi za atribut ID proizvoda.



Slika 7.19 Prikaz stranog ključa (FK)

Strani ključ identificira zapis drugog entiteta i može se ponavljati, tj. ne mora biti jedinstven. Pogledamo li ovdje prikazan primjer, logički se nameće da jedan kupac može izvršiti više narudžbi, što znači da se može više puta pojavljivati u opisu entiteta „Narudžba“. Uzevši u obzir ulogu u opisu entiteta, logički se nameće zaključak kako u opisu nekog entiteta može biti više atributa koji su strani ključevi.

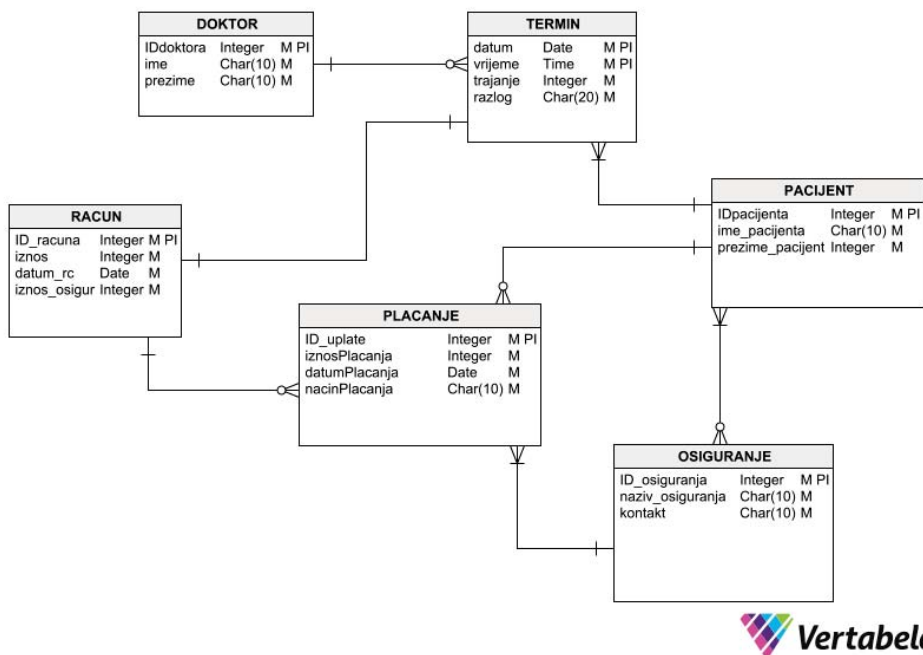
Uvedimo još jedan pojam kako bismo lakše modelirali podatke. *Bridge table* dopušta razbijanje veze „više-više“ na dvije veze „više-jedan“, čime se model spušta na razumljiviju razinu. Zapravo, *bridge table* nije ništa drugo nego novi entitet koji uvodimo kako bismo pojednostavili cijeli model. Primjerice, prethodno prikazan primjer sastojao se od tri entiteta – „Kupac“, „Narudžba“ i „Proizvod“. Međutim, kad biste pitali proizvoljnog kupca kao korisnika našeg primjera, on bi odgovorio da postoje dva entiteta – on kao kupac i proizvod koji on želi kupiti. U toj situaciji, model bi, prema njegovim informacijama, izgledao kao na sljedećoj slici (7.20).



Slika 7.20 Bridge table (prikaz)

Prema slici, jedan klijent (kupac) može kupiti nijedan ili više proizvoda, odnosno jedan proizvod se može prodati nijednom ili više kupaca. Ovakve veze, u svrhu jednostavnijeg modeliranja, razbijamo u više veza uvođenjem novog entiteta. U ovom slučaju je to, kod boljeg poznavatelja procesa kupnje, narudžba (makar fiktivna). Može se reći da je riječ o fizičkom modelu podataka jer model, osim entiteta, atributa i ključeva, sadrži i vrstu podataka za svaki od atributa (npr. broj - *integer*, riječ - *char* ili datum - *date*). Navedeno je prikazano u nastavku na slici 7.21.





Slika 7.21 Dijagram modela podataka korištenjem alata Vertabelo

Zaključimo, za svaki entitet postoji odgovarajuća tablica (Rouse, 2005) jer primarni ključevi omogućuju entitetu da bude prikazan pomoć tablice koja će prikazivati sadržaj baze podataka. Svaka tablica će imati broj stupaca koji odgovara broju atributa tog entiteta. Zapravo, transformacija modela EVA u tablični prikaz je temelj za deriviranje relacijskog modela iz EVA-a.

## 7.6 Evolucija modeliranja podataka

U tablici u nastavku prikazan je kratki pregled generacija modela podataka koje sugeriraju autori Coronel, Morris i Rob (2011).

Tablica 7.4 Evolucija modela podataka od 1960-ih do danas

Generacija	Razdoblje	Model podataka	Komentar
Prva	od 1960-ih do 1970-ih	Sustav datoteka	<ul style="list-style-type: none"> <li>Upravljač je zapisima, nisu bile uspostavljene veze.</li> <li>Uglavnom se koristilo na IBM-ovim sustavima.</li> </ul>

Druga	1970-ih	Hijerarhijski i mrežni	<ul style="list-style-type: none"> <li>Početak sustava baza podataka.</li> <li>Navigacijski pristup.</li> </ul>
Treća	od sredine 1970-ih do danas	Relacijski	<ul style="list-style-type: none"> <li>Jednostavan koncept.</li> <li>Modeliranje EVA i podrška relacijskom modeliranju podataka.</li> </ul>
Četvrta	od sredine 1980-ih do danas	Objektno orijentirani Objektno-relacijski	<ul style="list-style-type: none"> <li>Objektno-relacijski model podržava podatke objektnog tipa.</li> <li>Zvezdasta shema za skladišta podatka.</li> <li>Učestalost korištenja web baza podataka</li> </ul>
Iduća generacija	danas –	XML Hibridni	<ul style="list-style-type: none"> <li>Podrška za nestrukturirane podatke.</li> <li>Objektno-relacijski model podržava XML dokumente.</li> <li>Hibridni dodaje objektni front/end relacijskim bazama.</li> </ul>

Kad smo kod karakteristika koje razlikuju modele podataka, bitno je spomenuti kako, za razliku od modela EVA, OO modeliranje podržava agregaciju, što je semantički jači oblik od „obične“ veze, tj. asocijacije (Hoffer, Venkataraman, & Topi, 2013). Osnovne karakteristike tablično su prikazane u nastavku.

Tablica 7.5 Sličnosti/razlike OOM-a, modela EVA

OOM	Model EVA	Relacijski model
Klasa	Skup entiteta	Relacija
Objekt	Entitet	N-torka
Svojstvo	Atribut	Stupac

### 7.6.1 Prevođenje modela EVA u relacijskih model

Model podataka EVA svojim grafičkim prikazom predstavlja puno razumljiviji koncept od relacijskog modela. S druge strane, relacijski model je linearni prikaz koji je prikladniji računalnom prikazu i primjenom relacijske algebre omogućuje optimizaciju strukture baze podataka. I u jednom i u drugom modelu dvije su osnovne

komponente: objekti i veze. Zbog toga se i prevođenje modela EVA u relacijski modela odvija u dva koraka, pri čemu u oba mora biti poštovana jednoznačnost preslikavanja.

Prevođenje objekata provodi se na sljedeći način:

- *Ime objekta* postaje *Ime relacije*,
- *Svojstva objekta* postaju *atributi* relacijske sheme,
- *Identifikator* postaje *primarni ključ* relacije.

Prevođenje veza se radi u skladu s tipom veze, i to na sljedeći način:

- Veza 1:1 iskazuje se vanjskim ključem u onoj relacijskoj shemi gdje će poprimiti manje nul-vrijednosti.
- Veza 1:M iskazuje se vanjskim ključem, ali uvijek u onoj relacijskoj shemi koja je nastala od objekta na strani "više".
- Veze M:N iskazuju se otvaranjem treće relacijske sheme, čiji je ključ dvokomponentni, a sastoji se od primarnih ključeva objekata koji su u vezi.

### 7.6.2 Objektno-relacijski model

Prošireni relacijski model podataka (engl. *extended relational data model* – ERDM) je model podataka koji pridodaje većinu karakteristika objektno orijentiranih modela relacijskoj strukturi baze podataka. Odnosno, ERDM predstavlja novu generaciju relacijskog modela koja podržava objektno orijentirane karakteristike poput:

- 1) objekta,
- 2) proširene vrste podataka bazirane na klasama,
- 3) svojstva nasljeđivanja.

To su neki od razloga zašto se baza podataka, koja je bazirana na ERDM-u, često naziva objektno-relacijska baza podataka – O/R DBMS (Coronel, Morris, & Rob, 2011).

Zajednička karakteristika objektno-orijentiranog modela podataka (OODM) i proširenog modela EVA je to što oba modela pokušavaju zadovoljiti potrebu za uključivanjem više semantičkih informacija u sam model. Međutim, OODM i ERDM se znatno razlikuju u temeljnoj filozofiji i prirodi problemâ koje treba riješiti.

Unatoč tome što model EVA uključuje snažnu semantičku komponentu u svoj model, on je i dalje primarno baziran na konceptu relacijskog modela podataka. Suprotno tome, OODM je u potpunosti baziran na konceptu semantičkog modela podataka. ERDM je primarno usmjeren na poslovne aplikacije, dok je s druge strane OODM fokusiran na specijalizirane i inženjerske kompleksne softvere. Općenito, kad su u

pitanju baza podataka, scenarij koji se najčešće događa je spajanje koncepata i procedura OOM-a i relacijskog modela podataka.

### 7.6.3 Hibridni model

Hibridni DBMS-ovi također nastoje zadržati prednosti relacijskog modela, a istovremeno programerima pružiti objektno-orijentirani pogled na temeljne podatke. Istraživanja su pokazala kako hibridni model poboljšava pristupačnost, održivost, ali i brzinu postavljanja upita (Clara Kanmani, Suma, & Guruprasad, 2020). Ovaj tip baze podataka zadržava slične karakteristike performansi relacijskih modela, ali i bogatu semantičku podršku koju imaju objektni modeli (Coronel, Morris, & Rob, 2011).

Hibridni modeli podataka rješavaju problem heterogenih podatkovnih okruženja na način da se u zajedničku arhitekturu uključe i osnovne i nove baze podataka (Kobielus, 2013). Hibridni modeli podataka uključuju podatke iz višedimenzijskih i relacijskih izvora podataka. Ta značajka omogućuje programerima da stvaraju logičke činjenične tablice koje uključuju povijesni izvor podataka iz relacijskih izvora podataka, ali i da uključe buduće podatke koji mogu potjecati iz različitih, novih izvora (Rittman, 2009).

Složenost hibridnih arhitektura ovisi o rasponu izvora, opterećenju i samoj aplikaciji za koju se pruža podrška (Kobielus, 2013). Hibridni modeli su budućnost velikih podataka (engl. *big data*) jer korisnici sve više shvaćaju da ne postoji jedinstvena analitička platforma. Hibridni modeli imaju veliku primjenu kod manipuliranja velikim brojem podataka jer takav model pruža (Kobielus, 2013):

- 1) ekstremnu skalabilnost i brzinu,
- 2) ekstremnu agilnost i elastičnost,
- 3) ekstremnu dostupnost i upravljanje.

### 7.6.4 Budućnost modela podataka

Danas je O/R DBMS dominantna baza podataka za poslovne aplikacije. Razlog sve široj primjeni je jednostavnost modela, jednostavno korištenje, jezik *query*, visoke performanse transakcija, velika dostupnost, sigurnost, stabilnost te mogućnost nadogradnje i proširivanja. Suprotno tome, OO DBMS je popularniji kod sustava koji zahtijevaju podršku kompleksnih objekata, poput sustava za crtanje i proizvodnju (CAD/CAM), geografskih informacijskih sustave (GIS) te telekomunikacije i multimedije (Coronel, Morris, & Rob, 2011). U novije vrijeme testira se i razvoj sustava ERP na bazi podataka NoSQL, koja se pokazala fleksibilnijom od relacijske baze (Pustulka, von Arx & de Espona, 2023).

Činjenica je da modeli podataka postoje već dosta dugo i da će postojati i dalje, samo će se oblik podataka mijenjati i dalje razvijati (Stiglich, 2011). Koristeći semantičke tehnologije, podatci se mogu bolje razumjeti i koristiti na nove načine za automatizaciju otkrivanja znanja. Semantičke tehnologije (npr. semantički internet, rudarenje podataka i sl.) oslanjaju se na ontologiju, koju Stiglich (2011) naziva *modeli podataka na steroidima*. Bez obzira na tehnološki napredak koji se javlja (i javljat će se), i dalje će postojati potreba za ljudima koji su u stanju razumjeti poslovne potrebe za podacima i koji u skladu s tim mogu oblikovati i definirati te podatke, čak i ako se to radi samo na konceptualnoj i semantičkoj razini.

Na pitanje kakva je budućnost modeliranja podataka u oblaku, odgovor je ponudio Chisholm. Postoji nova kategorija baze podataka, „stupčasta baza podataka“ (engl. *columnar database*), koja se veže uz oblak iz razloga što je ta baza podataka namijenjena upravljanju podataka iznimno velikih razmjera (Chisholm, 2012). Riječ je o bazi koja se sastoji od tablica koje imaju retke. Redci su definirani ID-jem retka i mogu imati (gotovo) proizvoljan broj stupaca. To znači da, unutar jedne tablice, jedan redak može biti povezan s jednim stupcem ili pak s tisuću stupaca. U teoriji je moguće da ne postoje dva reda s potpuno istom kombinacijom stupaca.

Oblak se često povezuje s već spomenutim hibridnim modelom (Kobielus, 2013).

## 7.7 Literatura

1. Abhijit, A. P., & Ahuja, R. K. (2007). *Developing Web-Enabled Decision Support Systems*. Belmont, Massachusetts: Dynamic Ideas.
2. Armstrong, W. (1974). *Dependency Structure of database relationships*. IFIP Congress. Stockholm.
3. Beal, V. (2013). Entity-relationship model (diagram), dostupno na: [http://www.webopedia.com/TERM/E/entity\\_relationship\\_diagram.html](http://www.webopedia.com/TERM/E/entity_relationship_diagram.html)
4. Bergholt, L., Due, J. S., Hohn, T., Knudsen, J. L., Nielsen, K. H., Olesen, T. S., & Pedersen, E. H. (1998). *Database Management Systems: Relational, Object-Relational, and Object-Oriented Data Models*. The Centre of Object Technology.
5. Beynon-Davies, P. (2004). *Database Systems*. New York: PALGRAVE MACMILLAN.
6. Carli, J., & Maguire, J. (2000). *Mastering Data Modeling: A User - Driven Approach*.
7. Chen, P. P. (1976). The Entity-Relationship Model -Toward a Unified View of Data. *ACM Transactions on Database Systems* 1(1), 9-36.
8. Chen, P. P. (2002). Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned. *Software Pioneers*.
9. Chisholm, M. (2012). *Does Data Modeling Have a Cloudy Future?*, dostupno na: <http://erwin.com/community/expert-blogs/does-data-modeling-have-a-cloudy-future/>
10. Clara Kanmani, A., Suma, V., & Guruprasad, N. (2020). Hybrid data model of PACE and quadruple: An efficient data model for cloud computing. *International Journal of Computer Aided Engineering and Technology*, 13(1-2), 73-100. doi:10.1504/IJCAET.2020.108108
11. Codd, E. (1970). A Relational Model of Data for Large Shared Data Banks. *Communication of ACM* 13(6), 377-387.
12. Coronel, C., Morris, S., & Rob, P. (2011). *Database systems, design, implementation and management*. Cengage Learning.
13. Davis, U. (2011). *Conceptual Modeling using the Entity-Relationship Model*, dostupno na: <http://web.cs.ucdavis.edu/~green/courses/ecs165a-w11/2-er.pdf>
14. Delobel, C., & Casey, R. G. (1973). *Decomposition of a Data Base and the Theory of Boolean Switching Functions*. IBM J. Res. Develop. 17.
15. Fernandez, I., & Le Teno, J.-F. (1997). An Object Oriented Data Model for Geographic Application. *LCD Metodology*, 81-89.

16. Halpin, T. (2011). *Entity Relationship modeling from an ORM perspective: Part 3*. Microsoft Corporation.
17. Hay, D. C. (1999). *A COMPARISON OF DATA MODELING TECHNIQUES*. Essential Strategies, Inc.
18. Hoffer, J. A., Venkataraman, R., & Topi, H. (2013). *Modern Database Management*, 11/E. Prentice Hall PTR.
19. Kobielus, J. (2013). The Next Big "H" in Big Data: Hybrid Architectures, dostupno na: <http://ibmdatamag.com/2013/05/the-next-big-h-in-big-data-hybrid-architectures/>
20. Lurton, S. (2011). *Introduction to data modeling*. Deloitte.
21. Mamčenko, J. (2004). *Introduction to Dta Modeling and MSAccess*. Vilnius Gediminas Technical University. Dohvaćeno iz [http://gama.vtu.lt/biblioteka/Information\\_Resources/i\\_part\\_of\\_information\\_resources.pdf](http://gama.vtu.lt/biblioteka/Information_Resources/i_part_of_information_resources.pdf)
22. Panian, Ž. (2005). *Poslovna informatika za ekonomiste*. Zagreb: Masmedia.
23. Pavković, N. (2000). *OBJEKTNO ORIJENTIRANI PRISTUP*. disertacija. Zagreb, Hrvatska: Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje.
24. Pustulka, E., von Arx, S., & de Espona, L. (2023). Building a NoSQL ERP doi:10.1007/978-981-19-1610-6\_59
25. Riccardi, G. (2002). *Database Management*. Prentice Hall.
26. Rittman, M. (2009). *Creating Hybrid Relational-Multidimensional Data Models using OBIEE and Essbase*. Monterey, USA: ODTUG Kaleidoscope Conference.
27. Rob, P., & Coronel, C. (2009). *Database systems*. Course Technology.
28. Rouse, M. (2005). ERM or ER model. Dostupno na: <http://searchsqlserver.techtarget.com/definition/entity-relationship-model>
29. Silberschatz, Korth, & Sudarshan. (2001). *Database System Concepts, Fourth Edition*. The McGraw-Hill.
30. Stiglich, P. (2011). *How semantic modeling will change data modeling in the future*, dostupno na: <http://searchdatamanagement.techtarget.com/answer/How-semantic-modeling-will-change-data-modeling-in-the-future>
31. Tkalac, S. (1993). *Relacijski model podataka*. DRIP.
32. Vukmirović, S. (2013). *MODELIRANJE I ANALIZA PODATAKA U POSLOVANJU*. Rijeka: Ekonomski fakultet Sveučilište u Rijeci.
33. Xiaojuan, W. (2007). *How to transform an entity-relationship model into a relational model?*, dostupno na: <http://toyhouse.cc/profiles/blogs/how-to-transform-an-entity-relationship-model-into-a-relational>

## Popis slika

Slika 1.1 Sinergijske razine učeće organizacije.....	3
Slika 1.2 Informacijski tijekovi među organizacijskim podsustavima .....	6
Slika 1.3 Osnovne značajke odnosa organizacijskog sustava i njegova IS-a .....	8
Slika 1.4 Genetički taksonomski prostor (Brumec, Maček & Dušak, 2000) .....	10
Slika 1.5 Razvoj kriterija za izbor IS-a/IT-ja u okvirima organizacijske strategije.....	12
Slika 1.6 Odnos troška i učinka prema Nolanovu modelu razina (Wettstein & Kueng, 2002; Grabavac, 1991) .....	14
Slika 1.7 Procjena važnosti IS-a/IT-ja za organizaciju (Brumec, Maček & Dušak, 2000) .....	17
Slika 1.8 Faze planiranja i značenja IS-a (Earl, 1989).....	19
Slika 1.9 Prošireni model TAM .....	20
Slika 1.10 Dionici razvoja informacijskog sustava (izrada autora) .....	25
<i>Slika 2.1 Proces strateškog planiranja (Stacey, 2007) .....</i>	<i>36</i>
Slika 2.2 Odnos razvojne, IS i IT strategije (Ward & Griffiths, 1996).....	37
Slika 2.3 Matrica procjene važnosti IS-a za organizaciju (Brumec J., 1997; Hammer, 1990) .....	39
Slika 2.4 Smjernice korištenja novih informacijskih tehnologija procijenjenih na temelju analize SWOT .....	41
Slika 2.5 Alternativne strateške smjernice pSWOTa.....	42
Slika 2.6 Pet konkurentskih sila koje determiniraju industrijsku profitabilnost (Porter, 1985).....	44
Slika 2.7 Pogodnosti korištenja IS-a u kontekstu Porterova modela 5F .....	46
Slika 2.8 Vrijednosni sustav i generički lanac vrijednosti (Porter, 1985) .....	46
Slika 2.9 Primjena analize kritičnih faktora pri strateškom planiranju IS-a (Kaplan & Norton, 1996).....	50
Slika 2.10 Povratne veze analize End-means (Robson, 1997).....	51
Slika 2.11 Koncept metode Balanced Scorecard (Kaplan & Norton, 2014; Kaplan & Norton, 1996).....	52
Slika 2.12 Utjecaj inovacija i učenja na buduća ostvarenja organizacije (Martinsons, Davidson, & Tse, 1999) .....	56
Slika 3.1 Vrste i tipovi korištenih strategija.....	67
Slika 3.2 Postupak formalizacije strategije (Hell, Vidačić & Garača, 2009).....	72
Slika 3.3 Strateška mape ciljeva .....	76
Slika 3.4 Grafički prikaz razine ostvarenosti ciljeva .....	88
Slika 3.5 Očekivani odaziv sustava prema matričnom i SD modelu.....	91
Slika 3.6 Očekivani odaziv sustava prema matričnom i SD modelu.....	94



Slika 3.7 Dijagram tijeka aktivnosti za operativnu upotrebu matrice i SD modela .....	106
Slika 4.1 Proces upravljanja informatičkim projektom .....	116
Slika 4.2 Vodopadni pristup razvoju IS-a.....	122
Slika 4.3 Spiralni model razvoja IS-a (Boehm, 1988).....	124
Slika 4.4 Najpoznatije agilne metode (Collabnet, 2017).....	129
Slika 4.5 Formalizacija SPIS postupka (Brumec J. , Strategic Planing of IS, 1998)..	131
Slika 4.6 Metamodel metodologije DASPI (Vidačić & Brumec, 2013).....	134
Slika 5.1 Analiza zahtjeva: dijagram slučaja korištenja .....	142
Slika 5.2 Matrica ovisnosti zahtjeva.....	146
Slika 5.3 Prikaz objekta prema notaciji UML.....	150
Slika 5.4 Primjer suradnje među objektima.....	151
Slika 5.5 Primjer klasa i hijerarhije klasa .....	152
Slika 5.6 Prikaz potklasa .....	153
Slika 5.7 Dijagrami UML-a (UML-Diagrams, 2018.).....	156
Slika 5.8 Primjer klasnog dijagrama (izrađeno pomoću alata Draw.io) .....	158
Slika 5.9 Primjer objektnog dijagrama (alat Visual Paradigm).....	159
Slika 5.10 Primjer paketnog dijagrama (Lucid) .....	161
Slika 5.11 Dijagram uvođenja (internetski alat Lucid).....	162
Slika 5.12 Primjer kompozitnog strukturnog dijagrama (izrađeno pomoću internetskog alata Creately).....	164
Slika 5.13 Dijagram komponenti (izrađeno pomoću alata Lucid) .....	165
Slika 5.14 Primjer dijagrama aktivnosti.....	167
Slika 5.15 Primjer dijagrama slučaja korištenja .....	169
Slika 5.16 Slučaj korištenja na primjeru recepcije .....	171
Slika 5.17 Dijagram stanja .....	173
Slika 5.18 Primjer dijagrama slijeda kod rezervacije.....	175
Slika 5.19 Komunikacijski dijagram .....	176
Slika 5.20 Vremenski dijagram .....	177
Slika 5.21 Sučelje alata Visio za izradu UML dijagrama .....	178
Slika 5.22 Visual Paradigm .....	179
Slika 5.23 Prikaz internetskog sučelja alata Draw.io.....	180
Slika 5.24 Prikaz internetskog alata Lucidchart .....	181
Slika 5.25 Prikaz internetskog alata Creately.....	181
Slika 6.1 Cjelovit pogled na BPM (Willaert, Van den Bergh, Willems, & Deschoolmeester, 2012) .....	185
Slika 6.2 Primjer poslovnog procesa .....	187
Slika 6.3 Proces izvođenja nastave iz perspektive nastavnika i studenta .....	190
Slika 6.4 Proces izvođenja nastave.....	196
Slika 6.5 Dijagram s različitim vrstama aktivnosti.....	197

Slika 6.6 Dijagram koji sadrži više vrsta skretnica .....	198
Slika 6.7 Slika s prikazanim tijekovima .....	200
Slika 6.8 Primjer poslovnog procesa .....	200
Slika 6.9 Slika privatnog procesa .....	202
Slika 6.10 Slika javnog procesa.....	202
Slika 6.11 Prikaz kolaboracijskog dijagrama između dva privatna procesa: nadogradnja na prethodnu sliku.....	203
Slika 6.12 Slika kolaboracijskog dijagrama između sudionika .....	204
Slika 6.13 Konverzijski dijagram .....	205
Slika 6.14 Koreografski dijagram (Izrađeno pomoću alata ConceptDraw) .....	206
Slika 6.15 Šest faza BPM modela zrelosti (Kerremans, 2008).....	208
Slika 6.16 Procesno orijentirane aplikacije podržavaju cjeloviti poslovni proces..	212
Slika 6.17 Primjena servisa ovisno o različitim logičkim grupacijama .....	215
Slika 6.18 Prema Bruning, Weißleder & Mirosław Malek, 2007.....	216
Slika 7.1 Jasnoća specifikacija za razvoj novog IS-a .....	219
Slika 7.2 IS kao kuća (Lurton, 2011) .....	220
Slika 7.3 Razvoj baze podataka (Vukmirović, 2013).....	221
Slika 7.4 Unarno pridruživanje .....	224
Slika 7.5 Binarno pridruživanje .....	225
Slika 7.6 Chenova notacija .....	227
Slika 7.7 Vranina noga (engl. crow's foot) (IE) notacija .....	227
Slika 7.8 Usporedba Chenove i Martinove notacije (vranina noga).....	228
Slika 7.9 Relacijski model (Beynon-Davies, 2004).....	229
Slika 7.10 Objektno orijentirani model podataka (Beynon-Davies, 2004).....	235
Slika 7.11 Klasna hijerarhija .....	239
Slika 7.12 Prikaz agregacije .....	239
Slika 7.13 Dijagram EVA .....	240
Slika 7.14 Prikaz elementata baze podataka .....	241
Slika 7.15 Prikaz entiteta s atributima .....	241
Slika 7.16 Tablični prikaz entiteta i atributa.....	242
Slika 7.17 Prikaz entiteta, atributa i veza .....	242
Slika 7.18 Identifikacija primarnih ključeva (PK) .....	243
Slika 7.19 Prikaz stranog ključa (FK).....	244
Slika 7.20 Bridge table (prikaz).....	244
Slika 7.21 Dijagram modela podataka korištenjem alata Vertabelo .....	245

## Popis tablica

Tablica 1.1 Earlov model planiranja IS-a/IT-ja (Earl, 1989) .....	16
Tablica 3.1 Tablica zadanih strateških ciljeva .....	65
Tablica 3.2 Tablični prikaz elemenata SWOT-a .....	66
Tablica 3.3 Tablica SWOT .....	68
Tablica 3.4. Proširenje tablice SWOT – matrica aktivnosti i ISC-a .....	69
Tablica 3.5. Proširenje tablice SWOT – matrica strukture ciljeva (MSC).....	70
Tablica 3.6 Matrica resursa .....	82
Tablica 3.7 Preslikavanja objekata iz matričnog u SD model .....	93
Tablica 3.8 Naziv i opis aktivnosti operativne upotrebe matričnog i SD modela .....	95
Tablica 3.9 Popis i opis tijekova u operativnoj upotrebi matričnog i SD modela ....	100
<b>Tablica 4.1 Najpoznatije agilne metode (Collabnet, 2017) .....</b>	<b>129</b>
Tablica 4.2 Pregled problemskih koraka DASPIS-a te metoda i tehnika za njihovo rješavanje (Hell, Vidačić & Brumec, 2013).....	132
Tablica 5.1 Notacaija dijagrama klasa.....	157
Tablica 5.2 Notacija objektnog dijagrama .....	158
Tablica 5.3 Notacija paketnog dijagrama.....	160
Tablica 5.4 Notacija dijagrama uvođenja.....	161
Tablica 5.5 Notacija dijagrama raspodjele .....	163
Tablica 5.6 Notacija dijagrama komponenti .....	164
Tablica 5.7 Notacija dijagrama aktivnosti .....	165
Tablica 5.8 Notacija dijagrama slučaja korištenja .....	170
Tablica 5.9 Notacija dijagrama stanja .....	172
Tablica 5.10 Notacija dijagrama slijeda .....	174
Tablica 6.1 Notacija BPMN-a .....	193
Tablica 6.2 Notacaija konverzijskog dijagrama .....	205
Tablica 6.3 Notacija konverzijskog dijagrama .....	206
Tablica 6.4 Faze modela zrelosti.....	208
Tablica 7.1 Model EVA i relacijski model (Chen, 2002).....	229
Tablica 7.2 Relacija STUDENT .....	231
Tablica 7.3 Vrste ključeva i njihove definicije .....	233
Tablica 7.4 Evolucija modela podataka od 1960-ih do danas.....	245
Tablica 7.5 Sličnosti/razlike OO, ER i relacijskog modela .....	246

## Indeks pojmova

- BPMN, III, 178, 189, 190, 191, 192, 193, 194, 195, 197, 198, 200, 203, 208, 209, 212, 213, 215, 219
- BSC, 51, 52, 53, 54, 62, 64, 70, 74, 75, 79, 95, 97, 100, 101, 102, 104, 136
- informacijski sustav, I, II, 1, 2, 4, 7, 8, 27, 36, 38, 49, 109, 146, 185, 222
- IS/IT strategija, 36, 41, 54, 57, 66
- klasa, 114, 119, 142, 151, 152, 153, 155, 157, 158, 164, 166, 168, 172, 239, 240, 242, 243
- korisnički zahtjevi, 122, 141, 216
- kritični faktori uspjeha, 50
- matrični model, 78, 89, 90, 92, 104
- modeliranje, III, IV, 94, 111, 113, 115, 119, 120, 129, 133, 141, 154, 163, 172, 176, 177, 179, 180, 188, 189, 190, 192, 193, 194, 199, 201, 206, 212, 215, 222, 223, 224, 225, 229, 230, 233, 238, 249
- modeliranje podataka, IV, 222, 224, 225, 229, 233
- Modeliranje poslovnih procesa, 189, 215, 219
- Objekt, 65, 69, 149, 158, 173, 195, 239, 240, 249
- Objektni model, 133, 148
- Objektno orijentirani model, 154, 238, 239
- optimizacija, 3, 81, 89, 104, 135
- organizacija, III, 1, 2, 3, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18, 21, 33, 34, 35, 38, 41, 47, 48, 49, 51, 53, 63, 65, 66, 85, 88, 109, 110, 111, 113, 127, 135, 185, 186, 188, 191, 195, 211, 212, 213, 229, 233, 240
- organizacijski sustav, II, 1, 2, 4, 5, 7, 11, 13, 36, 65, 89, 109, 185
- plan, 51
- planiranja, I, III, 8, 10, 12, 14, 16, 18, 19, 27, 33, 34, 35, 36, 37, 38, 39, 40, 43, 45, 48, 50, 61, 63, 64, 82, 107, 118, 122, 129, 130, 146, 224
- Planiranje informacijskih sustava, 118
- poslovni procesi, 4, 132, 169, 185, 186, 188, 203, 210, 212, 222
- Preustroj poslovne tehnologije, 48
- prikupljanja zahtjeva, 29, 118, 122, 141, 142, 143, 145
- razvoj IS, III, 7, 15, 18, 21, 27, 53, 82, 109, 110, 118, 119, 120, 123
- SPIS, 38, 39, 130, 131, 133, 134, 135, 137
- strateški ciljevi, 5, 43, 53, 62, 64, 65, 71, 97, 100, 103, 104
- SWOT, 40, 41, 42, 43, 54, 55, 58, 62, 63, 65, 66, 67, 68, 69, 70, 71, 103, 104, 107, 132, 136, 138
- UML, 120, 125, 138, 141, 142, 149, 150, 154, 155, 156, 157, 159, 160, 166, 170, 177, 178, 179, 180, 181, 183, 190, 222
- zadani strateški cilj, 43, 65, 66, 77, 100, 103, 206

