

UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA

MAGISTRSKO DELO  
**ANALIZA INFORMACIJSKIH POTREB PREVAJALSKE AGENCIJE**

Ljubljana, junij 2014

SEBASTJAN HRIBAR

## IZJAVA O AVTORSTVU

Spodaj podpisani Sebastjan Hribar, študent Ekonomske fakultete Univerze v Ljubljani, izjavljam, da sem avtor magistrskega dela z naslovom Analiza informacijskih potreb prevajalske agencije, pripravljenega v sodelovanju s svetovalcem prof. dr. Mirom Gradišarjem.

Izrecno izjavljam, da v skladu z določili Zakona o avtorski in sorodnih pravicah (Ur. l. RS, št. 21/1995 s spremembami) dovolim objavo zaključne strokovne naloge/diplomskega dela/specialističnega dela/magistrskega dela/doktorske disertacije na fakultetnih spletnih straneh.

S svojim podpisom zagotavljam, da

- je predloženo besedilo rezultat izključno mojega lastnega raziskovalnega dela;
- je predloženo besedilo jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem
  - poskrbel(-a), da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam v zaključni strokovni nalogi/diplomskem delu/specialističnem delu/magistrskem delu/doktorski disertaciji, citirana oziroma navedena v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, in
  - pridobil(-a) vsa dovoljenja za uporabo avtorskih del, ki so v celoti (v pisni ali grafični obliki) uporabljena v tekstu, in sem to v besedilu tudi jasno zapisal(-a);
- se zavedam, da je plagiatstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku (Ur. l. RS, št. 55/2008 s spremembami);
- se zavedam posledic, ki bi jih na osnovi predložene zaključne strokovne naloge/diplomskega dela/specialističnega dela/magistrskega dela/doktorske disertacije dokazano plagiatstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom.

V Ljubljani, dne \_\_\_\_\_

Podpis avtorja(-ice): \_\_\_\_\_

# KAZALO

<b>UVOD</b> .....	<b>1</b>
<b>1 TIPI PREVAJALSKIH AGENCIJ IN SESTAVINE POSLOVNEGA PROCESA</b> .....	<b>4</b>
1.1 Organizacijska struktura.....	6
1.2 Zvrsti ravnalne strukture.....	7
1.2.1 Hierarhične zvrsti ravnalne strukture.....	7
1.2.2 Nehierarhične zvrsti ravnalne strukture.....	9
1.3 Poslovni procesi.....	10
1.3.1 Poslovni model.....	10
1.3.2 Opredelitev poslovnih procesov.....	13
1.3.3 Vrste elementov grafičnega prikaza poslovnih procesov.....	14
1.4 Celovite programske rešitve.....	14
<b>2 INFORMATIZACIJA POSLOVANJA</b> .....	<b>18</b>
2.1 Faze razvoja informacijskega sistema.....	18
2.2 Dva pristopa k razvoju sistema.....	22
2.2.1 Načrtovanje projekta.....	24
2.2.2 Analiza.....	24
2.2.3 Shranjevanje informacij in baze podatkov.....	38
<b>3 PRENOVA INFORMATIZACIJE POSLOVANJA</b> .....	<b>46</b>
3.1 Predstavitev podjetja CeFormT d.o.o. ....	46
3.1.1 Poslanstvo.....	46
3.1.2 Vizija.....	47
3.1.3 Poslovni koncept.....	47
3.1.4 Poslovni model.....	48
3.1.5 Organizacijska struktura.....	48
3.2 Opredelitev problema – dokument obsega sistema.....	50
3.2.1 Opis problema.....	50
3.2.2 Pričakovane poslovne koristi.....	54
3.2.3 Pričakovane zmožnosti novega sistema.....	54
3.3 Zbiranje informacij.....	55
3.3.1 Poslovni procesi v podjetju CeFormT d.o.o. ....	55
3.3.2 Preučitev obstoječih rešitev ponudnikov programske opreme.....	66
3.4 Opredelitev sistemskih zahtev – splošno.....	68

3.4.1 Seznam dogodkov in primerov uporabe v tabeli dogodkov .....	68
3.4.2 Opredelitev stvari.....	72
3.4.3 Diagrami razredov .....	74
3.5 Opredelitev sistemskih zahtev – objektni pristop.....	77
3.5.1 Opisi primerov uporabe .....	77
3.5.2 Diagrami primerov uporabe.....	92
<b>4 MOŽNOSTI IMPLEMENTACIJE.....</b>	<b>95</b>
<b>SKLEP.....</b>	<b>96</b>
<b>LITERATURA IN VIRI.....</b>	<b>98</b>

## KAZALO TABEL

Tabela 1: Število novoustanovljenih podjetij v zadnjih petih letih.....	5
Tabela 2: Povprečno število zaposlenih v Sloveniji v prevajalski panogi na podlagi delovnih ur v obračunskem obdobju .....	5
Tabela 3: Lasten razvoj ali nakup .....	37
Tabela 4: Kriteriji za ocenitev možnosti implementacije .....	37
Tabela 5: Uporabljene aktivnosti v življenjskem ciklu razvoja informacijskega sistema .....	38
Tabela 6: Moduli in problemska področja .....	52
Tabela 7: Seznam dogodkov in primerov uporabe za projektni management .....	68
Tabela 8: Seznam dogodkov in primerov uporabe za poročanje .....	70
Tabela 9: Seznam dogodkov in primerov uporabe za upravljanje s človeškimi viri .....	71
Tabela 10: Seznam dogodkov in primerov uporabe za upravljanje s strankami .....	71
Tabela 11: Opredelitev stvari.....	72
Tabela 12: Opis primera uporabe – odpiranje novega projekta.....	77
Tabela 13: Opis primera uporabe – odpiranje novega projekta z zahtevo za ponudbo .....	79
Tabela 14: Opis primera uporabe – časovni načrt projekta .....	81
Tabela 15: Opis primera uporabe – izvedba opravila .....	84
Tabela 16: Opis primera uporabe – zapiranje projekta.....	85
Tabela 17: Opis primera uporabe – poročilo o napredku projekta .....	86
Tabela 18: Opis primera uporabe – finančno poročilo projekta .....	87
Tabela 19: Opis primera uporabe – priklic mesečnega poročila .....	88
Tabela 20: Opis primera uporabe – nov človeški vir.....	89
Tabela 21: Opis primera uporabe – posodobitev matičnih podatkov človeškega vira .....	90
Tabela 22: Opis primera uporabe – priprava obračuna plač.....	90
Tabela 23: Opis primera uporabe – nova stranka .....	91
Tabela 24: Opis primera uporabe – posodobitev matičnih podatkov stranke.....	92

## KAZALO SLIK

Slika 1: Simboli diagrama aktivnosti .....	15
Slika 2: Model slapa življenjskega cikla razvoja sistema .....	21
Slika 3: Iteracija aktivnosti razvoja sistema .....	22
Slika 4: Diagram razreda modela domene – naročilo izdelka.....	30
Slika 5: Diagram primera uporabe .....	32
Slika 6: Diagram primera uporabe, organiziran po podsistemih.....	33
Slika 7: Diagram zahtev za tradicionalni in objektni pristop .....	34
Slika 8: Možnosti implementacije.....	36
Slika 9: Interakcija med sistemom za upravljanje baz podatkov, bazo podatkov in njenimi uporabniki .....	41
Slika 10: Dedovanje med objekti .....	45
Slika 11: Poslovni model podjetja CeFormT d.o.o. ....	48
Slika 12: Različica izmenične ravnalne strukture .....	49
Slika 13: Moduli predlagane celovite programske rešitve .....	51
Slika 14: Poslovni proces – pregled .....	58
Slika 15: Naročilo.....	60
Slika 16: Dinamični načrt.....	62
Slika 17: Prevod .....	64
Slika 18: Oddaja končnega izdelka .....	66
Slika 19: Diagram razredov za projektni management .....	75
Slika 20: Diagram razredov za upravljanje s človeškimi viri .....	76
Slika 21: Diagram primerov uporabe za projektni management.....	93
Slika 22: Diagram primerov uporabe za poročanje.....	94
Slika 23: Diagram primerov uporabe za upravljanje s človeškimi viri.....	94
Slika 24: Diagram primerov uporabe za upravljanje s strankami .....	95



## UVOD

Prevajalske agencije se razlikujejo po svoji organizacijski in kadrovski strukturi. Ena od glavnih razlik med posameznimi agencijami je portfelj storitev in jezikovnih kombinacij, ki jih nudijo svojim strankam, druga razlika pa se kaže v strukturi prevajalskih timov. Poznamo agencije, ki delajo izključno s timi stalno zaposlenih prevajalcev (v nadaljevanju stalni prevajalski tim), tiste, ki storitve prevajanja dajejo v zunanje izvajanje (v ta namen si zgradijo velike baze posameznih prevajalcev), in pa tiste, ki kombinirajo oba načina. V magistrskem delu se bom posvetil predvsem raziskovanju in analiziranju informacijskih potreb agencij s stalnimi prevajalski timi oziroma tistih agencij, pri katerih je odstotek storitev, ki jih dajejo v zunanje izvajanje, v primerjavi s celotnim obsegom storitev neznan. Osnova za to so večletne lastne izkušnje na tem področju.

Razlika v kadrovski strukturi posledično pogojuje različne procese, ki so potrebni za izvajanje storitev. Medtem ko agencije s pretežno zunanjim izvajanjem dosegajo manjše stroške iz naslova človeških virov, strojne in programske opreme ter projektne managementa, pa morajo več napora vložiti v zagotavljanje kakovosti in enotnega sloga besedila. To še posebej velja za projekte večjih obsegov, ki jih morajo razdeliti med več prevajalcev. Pri tem imajo vsekakor prednost agencije s stalnimi prevajalski timi, saj le-ti zagotavljajo večjo produktivnost na eni in višjo standardizirano kakovost z enotnim slogom na drugi strani, na strani projektne managementa, strojne in programske opreme pa je tudi vložek večji.

Skupen problem vseh treh tipov agencij so informacijske potrebe in procesi. Z informacijsko tehnologijo je namreč treba kar najbolj učinkovito podpreti procese v podjetju. Kot v svoji raziskavi ugotavlja podjetje XTRF, so posledica slabo definiranih procesov napake na naslednjih področjih: projektni management, ravnanje z datotekami, ravnanje z dokumenti, plačila strank in plačila ponudnikom ter upravljanje podjetja. Ker so ta področja med seboj odvisna ali pa vsaj povezana, lahko napaka z enega področja močno vpliva na drugo področje (How do translation companies work?, 2012).

Na prvi pogled se to ne zdi velik problem, saj na trgu obstaja veliko število programskih rešitev, ki so namenjene ravno prevajalskim agencijam in naj bi podpirale vse procese v takšnih podjetjih. To pa ne drži, saj obstaja možnost, da izbrana rešitev ne podpira vseh procesov. Večina rešitev spada v kategorijo TMS (angl. *Translation Management System*), ki pokrivajo področje ravnanja s prevajalskimi projekti, ne pa celotnih informacijskih potreb podjetja, kot jih recimo pokrivajo celovite programske rešitve ali ERP (angl. *Enterprise Resource Planning*). Po drugi strani pa obstaja verjetnost, da je treba rešitev v veliki meri prilagoditi procesom v podjetju, kar pa spet predstavlja velik strošek, ki ga je težko opredeliti pred začetkom prilagajanja. Zato se nekatere agencije odločajo za to, da same razvijejo programsko rešitev, ki jim najbolj ustreza. Včasih je to res boljša možnost, pogoj zanjo pa seveda je, da ima agencija dobro definirane procese, ki jih je treba s programsko rešitvijo podpreti.

Gradišar, Jaklič in Turk (2007, str. 274) zgornji dve možnosti informatizacije poslovanja razširijo na štiri:

- nadaljevanje lastnega razvoja obstoječih rešitev (dogradnja),
- lasten razvoj novih rešitev s pomočjo celovitih informacijskih orodij,
- nakup ali najem in uvajanje že izdelanih, standardiziranih, celovitih programskih rešitev,
- outsourcing programskih rešitev – zunanje izvajanje dejavnosti informatike.

Sargent (2012, str. 18–20) razpravlja o odločitvi prevajalskih agencij, ali naj programsko opremo za ravnanje s prevajalskimi projekti kupijo ali same razvijejo. V splošnem navaja, da je razvoj programske opreme za prevajalsko industrijo v fazi C na S-krivulji življenjskega cikla programske opreme, kar pomeni, da imamo trenutno na trgu dve skupini ponudnikov, in sicer tiste, ki so na trgu prisotni dlje časa in nudijo kompleksne programske rešitve z več možnostmi prilagajanja, ter „mlade“ ponudnike, ki nudijo manj kompleksne rešitve, ki so enostavnejše za implementacijo. Sargent napoveduje, da bo konkurenčni boj med tema dvema tipoma ponudnikov trajal še nekaj let, preden bo razvoj prešel v fazo D na S-krivulji. V tej fazi pa bo rešitev TMS vključena v širši ERP-sistem.

V našem podjetju smo trenutno na stopnji, ko imamo vsaj do neke mere opredeljene vse module ERP-rešitve, vendar le-ti med seboj niso povezani. Lastno rešitev smo oblikovali v okolju Microsoft Office Excel. Različne funkcionalnosti smo avtomatizirali in podprli z makri na podlagi programskega jezika Visual Basic for Applications (v nadaljevanju VBA). Čeprav je rešitev funkcionalna in fleksibilna, pa različni moduli ali pa celo različni sestavni deli modulov med seboj niso povezani s skupno bazo podatkov, ki bi zagotavljala točne in uporabne podatke.

Tako na primer orodje za načrtovanje izvedbe projektov (Evidenca naročil in načrtovanje prevajalskih projektov) vsebuje pregled nad vsemi prevajalskimi projekti in razporeditvijo prevajalcev na projekte (CeFormT d.o.o., 2012g, str. 50), samo orodje pa ni povezano s posameznimi evidencami prevajalcev, kjer je zabeleženo dnevno delo na posameznih projektih. Zaradi tega nimamo takojšnjega pregleda nad razporejenimi in porabljenimi urami na določenem projektu. Na nek način lahko govorimo o »otoški avtomatizaciji« (van der Hoeven, 2009, str. 7), saj zaradi nezadostne medsebojne povezanosti vsak oddelek deluje kot osamljen otok.

Iz zgornjega zapisa je razvidno, da potrebujemo celovito rešitev, ki bo na eni strani pokrivala vse procese v podjetju, in sicer z multilateralnimi razmerji (van der Hoeven, 2009, str. 7), na drugi strani pa ne bo zanemarjala ravnanja s prevajalskimi projekti, ampak ga bo učinkovito vključevala v proizvodni modul.

Treba je tudi omeniti, da se v prevajalski panogi v zadnjih letih trendi glede organizacijske in kadrovske strukture vse bolj premikajo v ustroj agencij tipa zunanjega izvajanja. Temu seveda sledijo ponudniki programskih rešitev za prevajalske agencije. Zato je toliko težje najti programsko rešitev, ki bi učinkovito podpirala procese in izpolnjevala informacijske potrebe prevajalske agencije s stalnim prevajalskim timom.



Problematiko, s katero se bom ukvarjal v magistrskem delu, lahko torej razdelim na dva dela: kakšni so proizvodni procesi v agenciji s stalnim prevajalskim timom in kakšne so informacijske potrebe za podporo teh procesov.

Z magistrskim delom želim analizirati informacijske potrebe prevajalske agencije s stalnim prevajalskim timom. Pogoj za to je opredelitev organizacijske in kadrovske strukture tega tipa prevajalske agencije. Nadalje je treba ugotoviti, kakšni proizvodni procesi se odvijajo v takšni organizaciji, jih opredeliti in razdeliti v sklope, ki sledijo teoriji ERP-rešitev in ki olajšajo ugotavljanje informacijskih potreb.

Cilj magistrskega dela je teoretični model celovite programske rešitve, ki bo prilagojen prevajalski agenciji s stalnim prevajalskim timom in ki bo temeljil na predhodno opravljeni analizi informacijskih potreb. Takšen model naj bi ustrezal celoviti programski rešitvi v fazi D življenjskega cikla na S-krivulji. Na podlagi tega se bo agencija lažje odločila, katero izmed štirih možnosti informatizacije poslovanja naj izbere.

Pri svojem delu bom za analizo procesov pretežni del informacij črpal iz večletnih izkušenj na področju prevajanja in lokalizacije, ki sem jih pridobil kot projektni manager v prevajalski agenciji s stalnim prevajalskim timom. V pomoč mi bodo tudi različni dokumenti, ki vsebujejo standarde prevajanja in lokalizacije, ter predvsem interno gradivo prevajalske agencije.

Pri postavitvi teoretičnega modela celovite programske rešitve se bom opiral predvsem na literaturo in vire, ki obravnavajo analizo in oblikovanje sistemov na eni strani in celovite programske rešitve za upravljanje in vodenje podjetij na drugi strani. Uporabil bom metodo analize, katere predmet bo stanje v prevajalski agenciji s stalnim prevajalskim timom. Prav tako bom uporabil primerjalno analizo procesov, da pokažem glavne razlike med procesi več tipov agencij.

Na podlagi analize stanja in procesov bom z metodo sinteze zasnoval teoretični model celovite programske rešitve za prevajalsko agencijo s stalnim prevajalskim timom. Magistrsko delo bom začel s kratko predstavitvijo različnih tipov prevajalskih agencij in razlikami med njihovimi organizacijskimi in kadrovske strukturi. V nadaljevanju bom pod drobnogled vzel tip agencije s stalnim prevajalskim timom in njene procese. Na podlagi tega bom procese razdelil v module, ki jih predlagajo teorije ERP-rešitev. Delo bom sklenil s celovitim teoretičnim modelom ERP-rešitve za prevajalsko agencijo s stalnim prevajalskim timom.

# 1 TIPI PREVAJALSKIH AGENCIJ IN SESTAVINE POSLOVNEGA PROCESA

Kot sem navedel uvodoma, se prevajalske agencije razlikujejo po organizacijski in kadrovski strukturi, storitvah, ki jih nudijo svojim strankam, in po poslovnih procesih, ki izhajajo iz prvih treh razlik. Potrebe po informacijski podpori poslovnih procesov oziroma informatizaciji, ki bi optimizirala izvajanje poslovnih procesov in podjetju do neke mere zagotovila prednost na trgu, so v večji meri skupne vsem tipom prevajalskih agencij, vsekakor pa niso enake. Če sem prej omenil razlike v poslovnih procesih pri večjih projektih, ki zahtevajo delitev med več prevajalcev in s tem povezane stroške projektnega managementa na eni strani ter zagotavljanjem kakovosti na drugi strani, naj tu podrobneje opredelim večji napor pri projektnem managementu.

Pri velikem prevajalskem projektu je treba delo razdeliti med več prevajalcev in ostalih članov projektnega tima, kar pa samo po sebi ne predstavlja bistvenega problema. Ta se pojavi pri agenciji s stalnim prevajalskim timom, ki hkrati izvaja množico projektov različnih obsegov z različnimi roki. Pri svojem vsakodnevem delu se srečujem z danim številom prevajalcev, kar pomeni dano število razpoložljivih delovnih ur na dan na eni strani in neznanim številom prevajalskih projektov in potrebnih delovnih ur na drugi strani. Težave se pojavijo, ko je število potrebnih delovnih ur višje od števila razpoložljivih delovnih ur. V teh primerih moram izvajati prerazporejanje prevajalcev z enim projektom na druge, pri čemer je treba vsako uro, ki jo vzamem enemu projektu, nadomestiti. To aktivnost sem poimenoval dinamično načrtovanje in razporejanje prevajalskih projektov. Razporejanje ne bi bilo potrebno oziroma bi se pojavljalo v manjši meri, če bi na primer prevajalske storitve dal v zunanje izvajanje in bi v ta namen imel na razpolago veliko bazo zunanjih izvajalcev.

Ko sem raziskoval ponudbo obstoječih programskih rešitev za ravnanje s prevajalskimi projekti in jih nekaj tudi testiral, sem ugotovil, da je večina teh povsem ustrezna za prevajalske agencije, ki svoje prevajalske projekte dajejo v zunanje izvajanje. Tak poslovni proces lahko na kratko opišemo, kot sledi. Agencija prejme naročilo, ga evidentira in po potrebi izda ponudbo. Po potrditvi ponudbe naročilo dodeli enemu prevajalcu ali razdeli več prevajalcem. Veliko ponudnikov programskih rešitev skuša svojo konkurenčno prednost uveljaviti ravno na tem koraku. Sistem namreč samodejno izbere ustrezne prevajalce glede na zahteve projekta (razpoložljivost do danega roka, področje prevajanja, jezikovna kombinacija idr.) iz baze prevajalcev in jim pošlje obvestilo o potencialnem projektu. V mojem primeru tak način seveda ne pride v poštev, saj ne predvideva večjega števila potrebnih razporejanj. To je bil tudi glavni motiv, da sem začel razmišljati o tem, da bi naše poslovne procese informatizirali z lastno razvito programsko rešitvijo.

Za slovenski trg sem s pomočjo podatkov, dostopnih na spletni strani Bonitete.si, analiziral podatke o ponudnikih prevajalskih storitev v Sloveniji, in sicer po številu zaposlencev na podlagi delovnih ur v določenem obračunskem obdobju. V Sloveniji je trenutno registriranih 1745 pravnih subjektov, ki opravljajo storitve prevajanja in tolmačenja.

Tabela 1 prikazuje porast števila novoustanovljenih podjetij v zadnjih petih letih. Število sicer počasi upada, vseeno pa lahko iz števil za posamezno leto na eni strani in iz dejstva, da za subjekte, ustanovljene v zadnjih petih letih, ni podatkov o številu zaposlenih ali pa to število redko presega število 1, sklepamo, da gre predvsem za prevajalce, ki prevajalske storitve izvajajo za agencije kot podizvajalci ali pa ti subjekti samo posredujejo prevajalske projekte v podizvajanje.

*Tabela 1: Število novoustanovljenih podjetij v zadnjih petih letih*

Datum ustanovitve	Število podjetij
2009	226
2010	196
2011	151
2012	139
2013	138

*Vir: M 74.300 - Prevajanje in tolmačenje, b.l.*

Tabela 2 prikazuje število zaposlenih za vseh 1745 subjektov, in sicer na podlagi delovnih ur v obračunskem obdobju. Samo 12 subjektov ima 5 ali več zaposlenih, kar pomeni, da so agencije s stalnim prevajalskim timom oziroma z redno zaposlenimi prevajalci v manjšini.

*Tabela 2: Povprečno število zaposlenih v Sloveniji v prevajalski panogi na podlagi delovnih ur v obračunskem obdobju*

Število zaposlenih	Število podjetij
Brez podatka	929
Manjše od 1	717
1	60
2	15
3	9
4	3
5	2
6	2
7	3
8	1
9	1
10	1
13	1
32	1

*Vir: M 74.300 - Prevajanje in tolmačenje, b.l.*

Najprej bom predstavil teoretične okvire organizacijske strukture, poslovnih procesov ter analize in zasnove informacijskega sistema oziroma informatizacije poslovanja. Prvo poglavje bom zaključil s kratko predstavitvijo razvoja celovitih programskih rešitev. Nato bom na primeru konkretne prevajalske agencije s stalnim prevajalskim timom zasnoval model celovite programske rešitve v podporo poslovnim procesom prevajalske agencije.

## 1.1 Organizacijska struktura

Organizacije so rezultat povezovanja ljudi v različne združbe z namenom, da bolje dosežajo svoje cilje, kot če bi se za njih trudili posamično. Ljudje v organizacijah so medsebojno povezani in odvisni, med njimi pa se oblikujejo različna razmerja (Rozman (2000, str. 4).

Lipovec (1987, str. 35) organizacijo opredeli kot »sestav medsebojnih razmerij med ljudmi (s povezavo v strukturo postanejo člani s tem nastale združbe), ki zagotavlja obstoj in posebne značilnosti s tem omogočeni združbi ljudi ter smotrno uresničevanje v strukturi usklajenih ciljev in delovanja združbe«.

Pučko (2008, str. 4) takšno opredelitev organizacije podjetja, ki je »sestav razmerij med ljudmi, člani podjetja, ki zagotavlja obstoj, družbeno-ekonomske in druge značilnosti podjetja ter smotrno uresničevanje cilja podjetja«, razdeli na dva dela: prvi del, ki predstavlja razmerja med ljudmi in strukturo odnosov v podjetju, je statičen, drugi, dinamičen del, pa predstavlja proces zagotavljanja obstoja, značilnosti in smotnosti delovanja v podjetju.

Tudi Orna (2004, str. 15) pojmovanje organizacije povzame v podobnih sedmih točkah in sicer pravi, da je za organizacijo značilno:

- da gre za združbo ljudi,
- da ima ekspliciten ali impliciten namen,
- da ustvarja ponudbo proizvodov in/ali storitev,
- da potekajo interakcije znotraj in zunaj organizacije,
- da zagotavlja obstoj,
- da ima strukturo in meje,
- da ima tako družbeni kot tudi tehnični sistem.

Člani združbe oziroma organizacije imajo v njej različne vloge in funkcije, ki jih povezujejo razmerja. Če so razmerja iste vrste, so enovita, različna razmerja pa so složena. Kadar je med seboj povezanih več enovitih razmerij, pride do enovite strukture, pri več različnih razmerjih pa do složene strukture (Rozman, 2000, str. 4).

V organizaciji nastajajo tudi procesi, ki jih glede na nastanek delimo na neformalne in formalne. Neformalni nastanejo zgolj z zblíževanjem, sodelovanjem in medsebojnim vplivanjem ljudi, kar vodi do oblikovanja pravil, norm in kulture organizacije. Formalne procese pa ljudje razvijejo

načrtno, da bi zagotovili smotrno delovanje in večjo učinkovitost. Formalni procesi so načrtovanje, uveljavljanje in kontrola (Rozman, 2000, str. 4).

Pri organizaciji moramo omeniti tudi temeljno razmerje, ki se oblikuje v podjetju. To je razmerje med lastniki in nelastniki proizvodnih sredstev. Lastnik proizvodnih sredstev lahko z njimi sam upravlja, tako da sam dela v svojem podjetju, v večjih podjetjih pa lastnik (lastniki) proizvodna sredstva upravljajo tako, da uravnavajo delo delavcev, ki izvajajo neposredno delo. Naslednja stopnja pa je, da lastniki tudi uravnavanje izvedbenega dela prepustijo drugim delavcem, ki so v vlogi ravnalcev oziroma ravnateljev (managerjev). Tako imamo lahko v podjetju razvite tri glavne funkcije: **upravljanje**, **ravnanje** in **izvedba** Rozman (2000, str. 5).

Rozman (2000, str. 6) upravljanje opredeli kot vrhovno »oblast v podjetju, ki varuje in zastopa interese lastnikov, s tem da daje ključne odločitve.« Upravljanje daje oblast ravnanju, ki usklajuje tehnično razdeljeno delo, tako da planira, uveljavlja in kontrolira. Njegovo bistvo je dodeljevanje dela drugim in neprestano odločanje. Zadnja funkcija je izvajanje, ki svojega dela ne more delegirati naprej, ampak izvede svoj del delovne naloge (Rozman, 2000, str. 6).

Strukture, ki jih tvorijo razmerja, so različne. V podjetju poznamo naslednje enovite strukture: tehnično, komunikacijsko, motivacijsko in oblastno. Tehnična struktura je povezana s tehnično delitvijo dela, komunikacijska s komuniciranjem članov združbe oziroma podjetja, motivacijska je povezana s potrebami in doseganjem ciljev, oblastna pa z dolžnostjo, odgovornostjo in avtoriteto v podjetju, ki morajo biti v pravilnem sorazmerju. Vse te enovite strukture pa so povezane v složeno strukturo, ki jo imenujemo (složena) organizacijska ali ravnalna struktura. Ta formalno določa položaj vsakega zaposlenega v podjetju (Rozman, 2000, str. 6–9).

Po Rozmanu (2000, str. 9) je ravnalna struktura »struktura zadolžitve, odgovornosti, avtoritete in mesta v komuniciranju« in »je najbolj izdelana formalna organizacijska struktura«. Ravnalne strukture so različne in v povezavi s hierarhijo poznamo linijsko, funkcionalno, štabno-linijsko in odborovsko. Med nehierarhične pa Lipovec (1987, str. 173–175) šteje problemsko ali timsko, matrično, projektno in izmenično ravnalno strukturo. V nadaljevanju bom na kratko predstavil zgoraj navedene zvrsti ravnalne strukture.

## **1.2 Zvrsti ravnalne strukture**

### **1.2.1 Hierarhične zvrsti ravnalne strukture**

**Linijska zvrst ravnalne hierarhije** temelji na tehnični delitvi dela in je najstarejši, najbolj razširjen in najpreprostejši tip hierarhije. Nastaja tako, da vsak nadrejeni svojo kompleksno delovno nalogo razdeli na preprostejše in jih dodeli v dolžnost podrejenim ravnalcem ali izvajalcem. Tem delegira tudi ustrezne pristojnosti. Vsak podrejeni ima samo enega predstojnika in temu odgovarja za izvedbo celotne naloge. Linijska struktura je preprosta, jasna in jo je najlažje organizirati, zato so tudi stroški organizacije najnižji. Spodbuja podjetnost ravnalcev, ker imajo veliko odgovornosti. Omogoča strogo kontrolo in disciplino. Je dobra v kriznih situacijah zaradi enotnih ukazov. Poleg tega so odgovornosti vsakega podrejenega zelo jasne.

Ena izmed pomanjkljivosti linijske strukture je zaradi dolgih komunikacijskih kanalov nejasna in nenatančna koordinacija. Tudi kadre je težje najti in se le-ti počasi oblikujejo, ker je delo ravnalca zelo zahtevno. Dejansko je primerna samo za enostavne naloge (Lipovec, 1987, str. 162).

**Funkcionalna zvrst ravnalne hierarhije** je prilagojena organizacijskim potrebam razvitejših industrijskih podjetij in se ne naslanja več strogo na klasične delitve dela. Nadrejeni ravnalec razdeli svojo delovno nalogo po dveh kriterijih. Po prvem kriteriju razdeljeno delo dodeli izvajalcem v izvedbo. Po drugem kriteriju isto nalogo razdeli na specialistična opravila. Ta specialistična opravila razdeli podrejenim funkcionalnim ravnalcem. Njih pooblašča za ravnanje vseh izvajalcev, ki so zavezani za delovne naloge, izoblikovane po prvem kriteriju. Vsak podrejeni izvajalec ima torej več nadrejenih funkcionalnih ravnalcev, a nobenemu od njih ne odgovarja za svojo celotno nalogo, temveč samo za ustrezen funkcionalni del. Glede na to, na kateri hierarhični stopnji uporabimo funkcionalno zvrst, ločimo tri zvrsti: delavniško zvrst (najnižja hierarhična stopnja), zvrst po oddelkih (srednja hierarhična stopnja) in zvrst po direkcijah (najvišja hierarhična stopnja) (Lipovec, 1987, str. 163).

Funkcionalna zvrst je primerna za razvito industrijo, ravnalno delo je možno močno razdeliti, lažje je najti in oblikovati kadre, ker so bolj specializirani, hitro deluje in se prilagaja novim razmeram ter zagotavlja večjo strokovnost v izvajanju ravnalnih nalog. Slabe strani so težavno in drago organiziranje, težavno vzgajanje kadrov za višje položaje, ker so le-ti zelo specializirani, in zaradi nasprotujočih si navodil več možnosti za spore med ravnalci o pristojnostih (Lipovec, 1987, str. 166).

**Štabno-linijska zvrst ravnalne hierarhije** ohranja funkcionalno specializacijo, vendar pa funkcionalni specialisti dobijo predvsem svetovalna pooblastila in delujejo v obliki svetovalnega štaba. Svoje izvedenske predloge dajejo linijskemu ravnalcu, ki odloča o tem, katero rešitev bo sprejel. Po značilnostih štaba ločimo tri podzvrsti: začasni svetovalci so zunanji strokovni sodelavci (npr. pravni svetovalci), stalni svetovalci so stalno zaposleni v podjetju, svetovalni oddelki so skupine strokovnjakov (Lipovec, 1987, str. 164).

Prednosti štabno-linijske strukture so, da je ni težko organizirati, hitro se prilagaja spremembam, štabi pa razbremenijo linijskega izvajalca. Pomanjkljivosti pa so možni konflikti med štabom in linijskim ravnalcem ter veliki stroški organizacije (Lipovec, 1987, str. 167).

**Odborovska zvrst ravnalne hierarhije** je pravzaprav linijska organizacija z enim ali več odbori. Odbor je navadno sestavljen iz linijskih ravnalcev z iste ali z različnih hierarhičnih stopenj. Vsak član odbora na svojem delovnem mestu izvaja naloge v skladu z dogovori v odboru. Podzvrsti se razlikujejo po tem, koliko vsak član odbora svoja pooblastila deli z drugimi člani (Lipovec, 1987, str. 165).

Odbore sestavljajo ljudje z istimi problemi, zato odbori omogočajo učinkovitejšo koordinacijo. Poleg tega so odbori zelo dobro sredstvo za informiranje. Odločitve odborov so za ljudi lažje

sprejemljive kot odločitve posameznika. Odbori zaradi razvite medsebojne komunikacije spodbujajo sodelovanje v podjetju in blažijo spore, porabijo pa tudi precej časa za seje in odločanje. V odborih se pojavljajo težnje po širitvi. Ko so preobsežni, niso več učinkoviti. Poleg tega so njihova pooblastila precej omejena, kar članom zbija iniciativnost (Lipovec, 1987, str. 167).

### 1.2.2 Nehierarhične zvrsti ravnalne strukture

Pri **problemski ali timski ravnalni strukturi** se naloge oblikujejo na podlagi problema, ki se rešuje, zato se tudi delo deli po problemih. Naloge so lahko tako kompleksne in spremenljive, da jih je težko vnaprej zajeti v kakšno strukturo. Poimenovanje timska struktura temelji na skupini delavcev, ki takšno nalogo lahko izvedejo, so pogosto z različnih strokovnih področij ter zelo enakopravno sodelujejo pri reševanju nalog (Lipovec, 1987, str. 174).

Prednosti timskega načina dela so, da aktivirajo celotni ustvarjeni potencial posameznika, izravnajo prednosti in pomanjkljivosti posameznih členov, pripomorejo k boljši kakovosti odločitev, izboljšajo komunikacijo in povečajo fleksibilnost podjetja. Slabosti pa so predvsem v porabljenem času za razprave, nejasne odgovornosti, neupoštevanje individualizma in podobno (Rozman, 2000, str. 103–104).

Pri **matrični ravnalni strukturi** postavimo samo ravnalca novega projekta, ne pa tudi oddelka, in določimo, kateri delavci bodo delali na projektu v poslovno-funkcijskih oddelkih. Le-ti imajo torej nadrejenega ravnalca svojega poslovno-funkcijskega oddelka, z ravnalcem novega projekta pa so v servisnem razmerju; ravnalec novega projekta je v tem primeru samo koordinator izvajanja nalog v zvezi s projektom (Lipovec, 1987, str. 175). Pri matrični strukturi igra veliko vlogo timsko delo. Ravno timsko delo lahko predstavlja šibko točko matrične ravnalne strukture. Usposabljanje ljudi za takšen način dela je lahko dolgotrajno in odraža pomanjkljivost matrične ravnalne strukture (Lipičnik, 2002, str. 56).

Za **projektno ravnalno strukturo** je značilno, da se poleg obstoječih poslovno-funkcijskih oddelkov formira tudi projektni oddelek, ki služi uvajanju novega proizvoda oziroma projekta v podjetje. V ta oddelek se razvrstijo ustrezni kadri, ki prostorsko ostanejo v poslovno-funkcijskih oddelkih. Vsak delavec ima torej dva nadrejena: ravnalca projekta in poslovno-funkcijskega ravnalca (Lipovec, 1987, str. 175).

Pri **izmenični strukturi** se mora delegiranje pooblastil prilagoditi spremenljivosti nalog. Tako na primer eden od delavcev dobi nalogo in postane njen ravnalec in hkrati izvajalec. Nato si med ostalimi raziskovalci poišče ustrezne sodelavce, ki tako postanejo izvajalci te naloge. Oni so hkrati lahko že ravnalci (oziroma izvajalci) kakšne druge samostojne naloge; vsak raziskovalec je torej lahko ravnalec, izvajalec ali pa oboje hkrati. Prednosti takšne strukture so, da omogoča zelo popolno komunikacijsko strukturo, demokratično delovno ozračje in dobro izrablja sposobnosti raziskovalcev (Lipovec, 1987, str. 174).

Zgoraj opisane organizacijske strukture vsaka na svoj način pogojujejo in določajo poslovne procese in njihovo izvajanje. V naslednjem poglavju bom predstavil nekaj splošnih značilnosti poslovnih procesov, ki bodo osnova za ponazoritev dejanskih poslovnih procesov v prevajalski agenciji s stalnim prevajalskim timom.

### **1.3 Poslovni procesi**

Sherry (2011, str. 11) pri oblikovanju poslovnih procesov opozarja na nekaj ključnih smernic, ki jih moramo upoštevati, če želimo, da je podjetje učinkovito in uspešno:

- V današnjem svetu je v celotni verigi vrednosti stranka najpomembnejša, zato morajo biti v okviru poslovnega modela podjetja zanj najbolj pomembne strankine potrebe in izpolnjevanje le-teh. V ta namen morajo podjetja nenehno izpopolnjevati in izboljševati svoje poslovne koncepte, strategijo in poslovni model (Sherry, 2011, str. 11).
- Poslovne procese moramo najprej dokumentirati. Ko so potrjeni, ugotavljamo tehnološke zahteve za podporo poslovnim procesom. S tem preprečimo nakup take tehnološke rešitve, ki ne bi ustrezala tehnološkim zahtevam (Sherry, 2011, str. 61). V nasprotnem primeru bi tvegali scenarij, da bi šele po nakupu tehnološke rešitve njej morali prilagajati poslovne procese, da bi jo sploh lahko uporabljali.
- Podjetje in poslovne procese moramo gledati od zunaj navznoter, torej s strankinega zornega kota. Le tako bomo lahko oblikovali poslovne procese, ki bodo ne samo izpolnjevali potrebe strank, ampak bomo lahko strankam ponudili rešitve, za katere se stranka sploh še ne zaveda, da jih potrebuje (Sherry, 2011, str. 13).
- Podjetje mora imeti pred oblikovanjem ali izboljšanjem poslovnih procesov izdelan poslovni koncept, strategijo in poslovni model (Sherry, 2011, str. 11). V nadaljevanju bom predstavil poslovni model podjetja, ki služi kot vzorčni model za prevajalsko agencijo s stalnim prevajalskim timom.

#### **1.3.1 Poslovni model**

Osterwalder in Pigneur (2010, str. 14) poslovni model opredelita kot način, kako podjetje ustvarja, posreduje in ohranja vrednost. Poslovni model delita na devet ključnih gradnikov. Ti nakazujejo, na kakšen način želi podjetje ustvarjati dobiček. Gradniki pokrivajo štiri glavna področja poslovanja: stranke, ponudbo, infrastrukturo in finančno uspešnost. Poslovni model lahko uporabimo kot načrt za strategijo, ki se bo odražala v strukturi, procesih in sistemih organizacije (2010, str. 15).

#### **Stranke**

Gradnik »Stranke« predstavlja kupce, ki jim podjetje želi ponujati svoje izdelke in storitve, in je osrednji del vsakega poslovnega modela. Pri oblikovanju poslovnega modela podjetje lahko stranke razdeli na več segmentov in se nato odloči, na katere segmente se bo osredotočilo in



katere bo opustilo. To je pogoj za oblikovanje poslovnega modela, ki bo upošteval specifične potrebe strank (Osterwalder & Pigneur, 2010, str. 20).

### **Ponudba vrednosti**

Gradnik »Ponudba vrednosti« predstavlja sklop izdelkov in storitev, ki ustvarja vrednost za določen tržni segment. Ponudba vrednosti, ki izpolnjuje določene potrebe strank, je tisto, zaradi česar se stranke odločajo za določeno podjetje. Ponudba vrednosti je lahko novost na trgu ali pa je podobna obstoječim ponudbam, vendar ponuja neko dodano vrednost. Značilnosti ponudbe vrednosti so: novost, zmogljivost, prilagajanje, oblikovanje, blagovna znamka, izpolnjevanje obljubljenega, cena, zniževanje stroškov, zmanjševanje tveganja, dostopnost, uporabnost in priročnost (Osterwalder & Pigneur, 2010, str. 23–25).

### **Kanali**

Gradnik »Kanali« opisuje načine, kako podjetje komunicira s potencialnimi tržnimi segmenti in kako jim dobavlja svojo ponudbo vrednosti. Kanale delimo na komunikacijske, distribucijske in prodajne ter igrajo pomembno vlogo v strankini izkušnji. Podjetje lahko svoje stranke doseže prek lastnih kanalov (lastna maloprodaja), partnerskih kanalov (distributerji) ali s kombinacijo obeh (Osterwalder & Pigneur, 2010, str. 26–27).

### **Odnosi s strankami**

Gradnik »Odnosi s strankami« opisuje odnose, ki jih podjetje vzpostavi s posameznimi tržnimi segmenti. Lahko so osebni ali pa popolnoma avtomatizirani. To je odvisno od tega, ali podjetje želi pridobiti nove stranke, zadržati obstoječe ali pa povečati prodajo. Seveda pa izbira vrste odnosov s strankami močno vpliva na izkušnjo strank. Ne glede na število različnih vrst odnosov s strankami, pa lahko podjetje le-te kombinira za vsak tržni segment. Primeri: osebna pomoč, namenska osebna pomoč, samopostrežne storitve, samodejne storitve (samopostrežne storitve z avtomatiziranimi procesi), skupnosti, soudeležba pri ustvarjanju vrednosti (Osterwalder & Pigneur, 2010, str. 28–29).

### **Prihodki**

Gradnik »Prihodki« predstavlja prihodke, ki jih podjetje ustvari na vsakem tržnem segmentu. Ključno vprašanje, ki si ga mora podjetje postaviti, je: »Kakšno ceno je stranka za izdelke in storitve pripravljena plačati?« Če uspešno odgovori na to vprašanje, lahko na vsakem tržnem segmentu vzpostavi enega ali več prihodkovnih tokov. Vsak prihodkovni tok pa ima lahko različne mehanizme za določanje cen (Osterwalder & Pigneur, 2010, str. 30–33).

## **Ključni viri**

Gradnik »Ključni viri« predstavlja najpomembnejši del poslovnega modela. Ključni viri so pogoj za ponudbo vrednosti, torej za izdelavo izdelkov in nudenje storitev. Še več, potrebni so za pozicioniranje na trgu, vzdrževanje odnosov s strankami in zagotavljanje prihodkov. Vrsta virov je odvisna od oblike poslovnega modela. Poznamo: materialni, intelektualni (blagovne znamke, avtorske pravice, podatki), človeški in finančni poslovni model (Osterwalder & Pigneur, 2010, str. 34–35).

## **Ključne dejavnosti**

Gradnik »Ključne dejavnosti« opisuje najpomembnejše dejavnosti, ki jih mora podjetje izvajati, da omogoči delovanje poslovnega modela. Podobno kot ključni viri so tudi ključne dejavnosti pogoj za ponudbo vrednosti, za pozicioniranje na trgu, vzdrževanje odnosov s strankami in zagotavljanje prihodkov. Prav tako so odvisne od oblike poslovnega modela. Ključne dejavnosti lahko kategoriziramo na proizvodnjo, reševanje težav in programska platformo/mrežo. Proizvodnja se nanaša na oblikovanje, izdelavo in dobavo izdelkov. Reševanje težav običajno zajema snovanje novih rešitev za težave, s katerimi se soočajo stranke. Ključne dejavnosti kategorije programske platforme/mreže pa bi bile management in promocija programske platforme ter nudenje storitev. Primer takega podjetja je na primer Microsoft (Osterwalder & Pigneur, 2010, str. 36–37).

## **Ključni partnerji**

Gradnik »Ključni partnerji« predstavlja mrežo dobaviteljev in partnerjev, ki omogočajo delovanje poslovnega modela. Podjetja s pomočjo partnerstev optimizirajo svoje poslovne modele, zmanjšajo tveganja ali pridobivajo vire. Poznamo štiri glavne kategorije partnerstev: strateške zveze med nekonkurenti, hkratno sodelovanje in konkuriranje (strateška partnerstva med konkurenti), skupna vlaganja za razvoj novih poslov ter odnos kupec – dobavitelj, ki zagotavlja surovine. Motivacija za oblikovanje partnerstva izhaja iz možnosti po: optimizaciji in ekonomičnosti, zmanjšanju tveganja in negotovosti ter pridobitvi posebnih virov in dejavnosti (Osterwalder & Pigneur, 2010, str. 38–39).

## **Stroški**

Gradnik »Stroški« opredeljuje najpomembnejše stroške, ki nastajajo pri poslovanju v okvirih določenega poslovnega modela. Za izračun le-teh moramo imeti opredeljene ključne vire, ključne dejavnosti in ključne partnerje. Načeloma je treba stremeti k čim manjšim stroškom, vendar se nekatera podjetja bolj osredotočajo na stroške kot druga. V splošnem se poslovni modeli lahko delijo na tiste, ki se osredotočajo na stroške, in na tiste, ki se osredotočajo na ponujeno vrednost. Seveda je veliko podjetij nekje vmes (Osterwalder & Pigneur, 2010, str. 40–41).

Na koncu opredelitve poslovnega modela se zastavlja vprašanje, kako se poslovni model razlikuje od strategije podjetja, ki se po Pučku (2009, str. 22): »nanaša na vprašanje ustvarjanja posebne kombinacije poslovnih aktivnosti, ki bo drugačna od kombinacije, ki so jo razvili konkurenti, in ki je konkurenti ne bodo mogli zlahka in na hitro posnemati.«

Casadesus-Masanell in Enric Ricart (2009, str. 2) poslovni model opredelita kot logiko, na podlagi katere podjetje posluje in ustvarja vrednost, strategijo pa kot izbiro poslovnega modela, na podlagi katerega bo podjetje konkuriralo na trgu.

Močnik (2010, str. 145) kot razliko navaja dejstvo, da poslovni sistem omogoča dopolnjevanje posameznih delov podjetja, strategija pa vključuje še konkurenco. Podobno zaključijo Strimbei (2005, str. 93), ko pravi, da se strategija osredotoča na tiste dele poslovnega modela, ki lahko omogočijo konkurenčno prednost.

Tudi Seddon in Lewis (2003, str. 246) pri razlikovanju obeh pojmov navajata, da je strategija podjetja vedno vezana na njegovo specifično konkurenčno okolje. Posledično je strategija določenega podjetja lastna samo temu podjetju. Na drugi strani pa lahko poslovni model smatramo kot abstrakcijo strategije podjetja. Zato lahko en poslovni model uveljavlja več podjetij.

### **1.3.2 Opredelitev poslovnih procesov**

Sherry (2011, str. 10) poslovni proces definira kot kontinuirano dejavnost, operacijo ali serijo sprememb, ki se odvijajo na poseben način. Proces je sestavljen iz opravil, vlog, virov in dejavnosti, ki jih je treba izvesti. Imeti mora lastnika, biti mora dejavnost in temeljiti na vložkih in izloških.

Po Bosilj Vukšičevi, Hernausu in Kovačiču (2008, str. 17) je poslovni proces strukturiran, analitični in medfunkcijski sklop dejavnosti, ki zahteva nenehno napredovanje. Dejavnosti imajo jasno določen začetek in konec ter v bolj ali manj podobnih intervalih ustvarjajo vrednost za potrošnike.

Tudi Kovačič, Jaklič, Indihar Štemberger in Groznik (2004, str. 78) podobno opredelijo poslovni proces. Pravijo, da gre za logično povezane izvajalske in nadzorne postopke, ki imajo za posledico nek rezultat (končni izdelek, opravljena storitev ...).

Harmon (2007, str. 198) proces opredeli kot omejen sklop aktivnosti, ki so izvedene po prožilnem dogodku z namenom, da se ustvari nek rezultat. Procese deli na enostavne in izjemno kompleksne.

Po Kovačič et al. (2004, str. 78–79) poslovne procese bolj podrobno opredeljujejo:

- prožilci procesa: prejem naročila, strankin klic ...,
- vhodi v proces: vsebina naročila,
- izhodi iz procesa: izdelki, storitve ...,
- lastnik procesa: odgovorna oseba za izvedbo in nadzor procesa,
- omejitve: pogoji obsega izvajanja procesa in omejitve pristojnosti lastnika procesa,
- stroški: viri, potrebni za izvajanje procesa (človeški, materialni ...),
- ključni dejavniki uspeha: pomembni cilji, ki stranki zagotavljajo uspešnost izvajanja procesa.

Hierarhično poslovne procese delimo na podprocese in/ali aktivnosti, pri čemer je podproces skupek opravil ali delovnih operacij, ki se v organizaciji izvajajo, aktivnost pa je tisti najmanjši sestavni del procesa, ki ga je še smiselno prikazati v diagramu poslovnega procesa (Kovačič et al., 2004, str. 78–79).

### 1.3.3 Vrste elementov grafičnega prikaza poslovnih procesov

Obstoječe poslovne procese bom predstavil z diagrami aktivnosti po standardih enotnega jezika za objektno modeliranje UML (angl. *Unified Modeling Language*), ker bom za aktivnosti analize sistemskih zahtev uporabil objektni pristop. Te standarde vzdržuje skupina OMG (angl. *Object Management Group*), ki jo sestavlja več kot 800 ponudnikov programske opreme, razvijalcev in organizacij, ki se trudijo za standardizacijo objektnih sistemov (Satzinger, Jackson, & Burd, 2009, str. 240). Tudi vse ostale modele v magistrskem delu bom modeliral po UML-standardih.

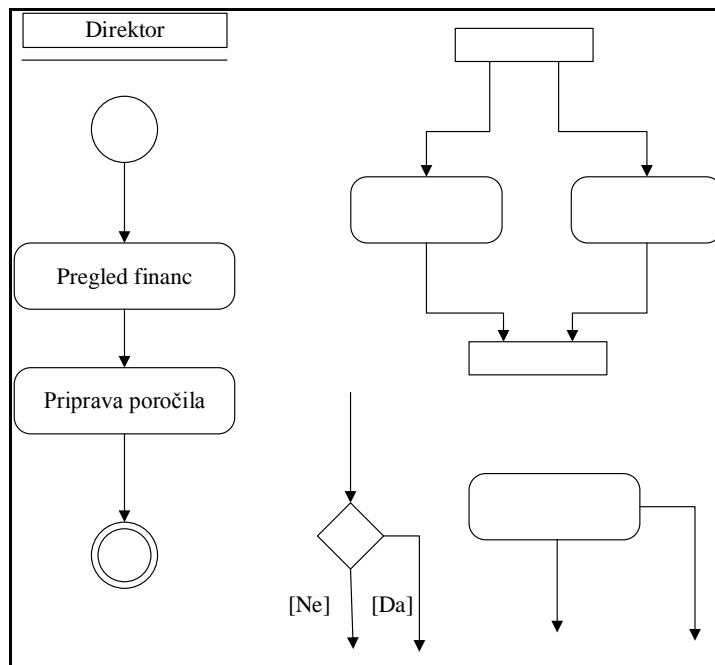
Na Sliki 1 so prikazani glavni simboli in njihov pomen, ki jih bom uporabil za modeliranje poslovnih procesov, in sicer v obliki **diagramov aktivnosti**. To so diagrami delovnih tokov, ki prikazujejo različne aktivnosti, uporabnike ali sistem, ki izvajajo te aktivnosti, in tok njihovega zaporedja. Kroga predstavljata začetek in konec delovnega toka. Aktivnosti so prikazane kot zaobljeni pravokotniki, ki jih povezujejo puščice, ki označujejo zaporedje aktivnosti. Odločitev v delovnem toku je lahko predstavljena na dva načina. Prvi način je romb, kjer se glede na odločitev delovni tok usmeri v eno ali drugo smer. Drug način pa je, da za odločitev ne uporabimo romba, ampak se delovni tok usmeri neposredno iz aktivnosti, kot je vidno na sliki spodaj desno. UML-diagram aktivnosti pozna tudi sinhronizacijsko vozlišče, ki delovni tok razdeli na dva vzporedna tokova in jih nato spet združi. Še ena pomembna lastnost pa so plavalne proge, ki glede na izvajalca aktivnosti razmejujejo celoten diagram. Primer plavalne proge je leva polovica na Sliki 1, kjer je izvajalec aktivnosti direktor (Satzinger et al., 2009, str. 141).

## 1.4 Celovite programske rešitve

Uvodoma sem dejal, da potrebujemo ERP-rešitev, ki bi bila po Sargentu (2012, str. 18–20) v fazi D na S-krivulji, kar pomeni, da bi bila rešitev oziroma funkcionalnost rešitve TMS vključena v širši ERP-sistem. Po drugi strani pa sem z nadaljnjim raziskovanjem prišel do ugotovitve, da

klasična vrsta ERP-rešitve ni primerna za prenovu informatizacije naših poslovnih procesov. V nadaljevanju bom na kratko opredelil ERP, da bo razlika med ERP in našimi potrebami preнове informatizacije bolj jasna.

Slika 1: Simboli diagrama aktivnosti



Vir: Satzinger et al., *Systems Analysis and Design in a Chnging World*, 2009, str. 142.

Kaj sploh je celovita programska rešitev ali ERP? Na spletnih straneh [erp.com](http://erp.com) (What is ERP?, 2012) je celovita programska rešitev opredeljena kot en sistem za upravljanje vseh področij poslovanja v podjetju, in sicer s samo eno programsko rešitvijo.

Kovačič, Groznik in Ribič (2009, str. 243) ERP opredelijo kot celovito programsko rešitev, torej kot rešitev, ki podpira izvajanje procesov notranje vrednostne verige v organizaciji in služi kot podpora izvajanju medorganizacijskih procesov zunanje vrednostne verige.

Wallace in Kremzar (2001, str. 11–12) ERP opredelita kot sklop orodij za celo podjetje za napovedovanje, načrtovanje in časovno razporejanje, ki uravnava ponudbo in povpraševanje, povezuje stranke in dobavitelje v celotno verigo vrednosti, uporablja preizkušene poslovne procese za odločanje in koordinira prodajo, marketing, poslovanje, logistiko, nabavo, finance, razvoj proizvodov in človeške vire.

Tudi van der Hoeven (2009, str. 7–8) namen ERP-ja opredeli kot načrtovanje in sinhronizacijo vseh virov celotnega podjetja, ki mora pokrivati vse procese v podjetju, in sicer z multilateralnimi razmerji. Vsi zaposleni v podjetju naj bi izvajali vsakodnevna opravila znotraj ERP-rešitve, ERP pa pri tem nudi en integriran informacijski sistem in eno osrednjo bazo podatkov.

Kovačič et al. (2009, str. 247) ERP razdelijo na tri ravni: podatkovno, poslovno in predstavitevno. Podatkovna raven je vir podatkov v obliki centralizirane normalizirane baze podatkov, ki je denormalizirana samo na mestih, kjer je to zaradi odzivnosti smiselno. Poslovna raven vsebuje logiko poslovnih procesov, kjer so poslovne entitete obravnavane in modelirane v obliki objektov. Gre torej za objektni pristop. Predstavitevna raven pa služi prikazovanju uporabniku prijaznih informacij. Informacije so običajno prikazane na različnih napravah s pomočjo različnih uporabniških vmesnikov.

Sami začetki ERP-ja segajo v devetdeseta leta dvajsetega stoletja in so naslednik logističnih rešitev MRP (angl. *Material Requirements Planning*) in MRP II (angl. *Manufacturing Resources Planning*).

MRP predstavlja načrtovanje materialnih potreb. Gre za metodologijo, ki načrtuje potrebe po materialu za proizvodni proces na podlagi pričakovane prodaje. Pri tem je treba upoštevati najmanjšo količino, ki jo je treba naročiti, obstoječo zalogo in seveda rok dobave (van der Hoeven, 2009, str. 13–15).

MRP II je naslednik MRP-ja. Predstavlja načrtovanje proizvodnih virov in je razširitev MRP-ja, saj upošteva dejstvo, da za proizvodnjo ne potrebujemo samo materiala, ampak tudi stroje ter ljudi. Stroji in ljudje so poimenovani kot viri. Če je treba načrtovati vse proizvodne vire, potem načrtovanje postane zelo kompleksno in vključuje veliko spremenljivk. Posledično se je povečala moč računalnikov, pojavila pa so se tudi nekatera programska orodja za načrtovanje. Slaba stran je bila ta, da so posamezni oddelki podjetja uporabljali ločene programske sisteme, kar je vodilo do nesinhroniziranih in podvojenih podatkov v različnih sistemih. To je imelo za posledico odvečno delo, večkratno vnašanje istih podatkov v sistem, nepravočasna plačila in podobno (van der Hoeven, 2009, str. 15–16).

Izvor ERP-ja je bil pogojen z zgoraj navedenimi pomanjkljivostmi rešitve MRP II, saj se je pojavila težnja po integraciji vseh vnosov podatkov v en sistem, pri čemer bi različni moduli (posamezni podsistemi) brez težav medsebojno komunicirali. Poleg tega bi tak sistem moral podpirati vse vire podjetja (van der Hoeven, 2009, str. 16).

Van der Hoeven (2009, str. 18) ERP razdeli na dva pojma. Prvi je ERP kot aplikacija, ki je standardni paket programske opreme z integrirano funkcionalnostjo, ki pokriva poslovne procese cele organizacije. ERP-sistem, ki ga upravlja taka ERP-rešitev, uporablja centralno bazo podatkov. Drugi pa je ERP kot koncept integriranega načrtovanja in upravljanja cele organizacije. Gre za interakcijo med vsemi poslovnimi procesi organizacije in interakcijo z okoljem organizacije, kar pomeni med strankami, dobavitelji in poslovnimi partnerji.

ERP pa ni bil zadnja stopnja v razvoju. Na prelomu stoletja je dobil naslednika v ERP II ali x-ERP (angl. *extended ERP*), tj. razširjeni ERP. Razširitev vključuje tesno sodelovanje z drugimi podjetji, kot so dobavitelji, in s strankami. Podjetja lažje dosežejo konkurenčno prednost, če oblikujejo neke vrste koalicio, ki vključuje celotno verigo od proizvodnih surovin do končanega

izdelka. Pri ERP gre za procese in sodelovanje znotraj podjetja, pri ERP II pa gre za medpodjetniško sodelovanje. Danes je »čistih« sistemov tipa ERP I malo, saj bolj ali manj vsi ponudniki ERP-rešitev v svoje rešitve vključujejo vidike sistema ERP II (van der Hoeven, 2009, str. 19).

Naslednja stopnja razvoja je ERP III, ki v ekosistem podjetja še bolj vključuje stranke in širši trg prek družabnih omrežij in podobnih medijev. To pa presega trenutne okvire modulov CRM, ki stranke še vedno pretežno obravnavajo v okvirih podjetja v smislu spremljanja prodajnih aktivnosti (Wood, 2010).

Sami moduli oziroma podsistemi, ki sestavljajo celovito programsko rešitev, se razlikujejo glede na panogo, v kateri podjetje posluje. Storitvena organizacija ima drugačne potrebe kot čisto proizvodna organizacija. Zato ponudniki ERP-rešitev ponujajo panožne rešitve oziroma lahko podjetje izbere tiste module, ki v celoti podpirajo njegove poslovne procese.

Tako na primer podjetje OpenERP ponuja prilagodljivo celovito programsko rešitev, ki jo sestavlja večje število podsistemov ali aplikacij, kot jih sami poimenujejo. Nudijo vse od računovodskega, nabavnega, proizvodnega, prodajnega pa do projektnega modula. Poleg glavnih 9 modulov imajo razvitih še dodatnih 200 aplikacij (Choose your application, 2013) za podporo poslovanju.

Kovačič in Bosilj Vukšić (2005, str. 278) uvajanje celovitih programskih rešitev opredelita kot enega od pomembnih pristopov k prenovi in informatizaciji poslovanja. Pripomogla naj bi k boljšemu obvladovanju podatkov ter poslovnemu odločanju. Po njunem mnenju gre za strateško pomemben in tudi nujen projekt, ki pa ima lahko pozitivne ali negativne posledice, saj je iz prakse razvidno, da takšni projekti vključujejo velika tveganja in imajo relativno nizko stopnjo uspešnosti.

Glede na naše poslovne procese bi bila klasična ERP-rešitev preveč kompleksna, saj določenih modulov (podsistemov) ne bi uporabljali v celoti, nekaterih pa sploh ne bi uporabljali (v našem konkretnem primeru imamo celotno računovodstvo v zunanjem izvajanju). Zato sem pozornost usmeril v prenovu informatizacije poslovanja na način, ki bi zagotovil enovito programsko rešitev, sestavljeno iz bistvenih modulov, ki bi informatizirali vse naše procese in bi uporabljali enotno bazo podatkov, kar je sicer značilnost ERP-rešitev. Taka rešitev bi bila po vzoru rešitve OpenERP, kar pomeni, da bi bila izdelana na modularen način, ki bi omogočal tudi kasnejše dodajanje modulov, če bi bilo potrebno. O enoviti programski rešitvi pa govorim, ker naj bi vseeno povezala vse naše ključne procese: od načrtovanja do prevajanja in zbiranja podatkov za različne namene.

## **2 INFORMATIZACIJA POSLOVANJA**

Poslovne procese informatiziramo tako, da njihovo izvajanje podpremo z ustreznim informacijskim sistemom. Satzinger et al. (2009, str. 6) informacijski sistem opredelijo kot zbirko medsebojno povezanih sestavnih delov, ki zbirajo, obdelujejo, hranijo in posredujejo informacije, ki so potrebne za izvedbo poslovnega opravila. Poslovno opravilo lahko razumemo kot aktivnost, ki jo je treba opraviti v sklopu celotnega poslovnega procesa.

Sam razvoj informacijskega sistema Satzinger et al. (2009, str. 4) delijo na analizo in oblikovanje (snovanje) sistema. Pri tem analizo opredelijo kot proces razumevanja in podrobnega opredeljevanja ciljev, ki jih mora informacijski sistem doseči. Oblikovanje sistema pa razumejo kot proces podrobnega opredeljevanja vseh sestavnih delov informacijskega sistema. Oba procesa izvaja sistemski analitik, katerega naloga je, da z informacijsko tehnologijo reši probleme poslovnih procesov.

Prej omenjeni sestavni deli informacijskega sistema so lahko:

- podsistemi, ki pokrivajo in podpirajo določeno področje poslovanja v podjetju (računovodstvo, proizvodnja in podobno),
- sestavni deli sistema v smislu strojne opreme, programske opreme, vložkov, izložkov, podatkov, ljudi in procesov.

Oba vidika sestavnih delov sta pri analizi in oblikovanju informacijskega sistema za sistemskega analitika zelo pomembna in uporabna (Satzinger et al., 2009, str. 7–8).

### **2.1 Faze razvoja informacijskega sistema**

Razvoj informacijskega sistema lahko razdelimo na pet faz, in sicer na: načrtovanje projekta, aktivnosti analize, aktivnosti oblikovanja, aktivnosti implementacije in aktivnosti podpore. V nadaljevanju so navedene aktivnosti in dodatna pojasnila posameznih faz, kot jih opredelijo Satzinger et al. (2009, str. 45–49):

#### **Načrtovanje projekta:**

- opredelitev problema,
- priprava terminskega načrta projekta,
- potrditev izvedljivosti,
- določitev izvajalcev projekta,
- začetek projekta.

Pri fazi načrtovanja projekta je ključnega pomena opredelitev projekta, saj je treba natančno določiti problem, ki ga želimo rešiti, in obseg same rešitve.



### **Aktivnosti analize:**

- zbiranje informacij,
- opredelitev sistemskih zahtev,
- izdelava prototipov za odkrivanje zahtev,
- določitev prioritet zahtev,
- izdelava in vrednotenje različnih možnosti izvedbe,
- pregled priporočenih možnosti izvedbe skupaj s poslovodstvom.

Pri analizi sta v ospredju raziskovanje oziroma odkrivanje in razumevanje. Razumeti in zapisati moramo poslovne potrebe in zahteve novega sistema. Analiza se začne z zbiranjem informacij. Cilj tega je, da se čimbolj poučimo o problemu, ki ga želimo z informacijskim sistemom rešiti. Načinov zbiranja informacij je več in vključujejo: opazovanje uporabnikov pri delu, pogovore z uporabniki, branje obstoječe dokumentacije, pregled poslovnih pravil, pregled obstoječih sistemov in pogovori z drugimi deležniki sistema.

Čeprav je zbiranje informacij zelo pomembno, pa samo to ni dovolj. Zbrane informacije moramo pregledati, analizirati in strukturirati, da pridemo do razumevanja celotnega sistema in sistemskih zahtev procesiranja. Za oblikovanje sistemskih zahtev se pogosto uporabljajo diagrami, s katerimi se nazorno pokaže zahteve procesiranja novega sistema.

Če izdelamo prototip določenega dela novega informacijskega sistema, ki analitiku pomaga razumeti in opredeliti sistemske zahteve, je le-ta lahko zelo uporaben tudi za uporabnike sistema, ki lahko ob pregledu prototipa bolj jasno opredelijo svoje potrebe v novem sistemu.

### **Aktivnosti oblikovanja:**

- oblikovanje in integracija mreže,
- oblikovanje arhitekture aplikacije,
- oblikovanje uporabniških vmesnikov,
- oblikovanje sistemskih vmesnikov,
- oblikovanje in integracija baze podatkov,
- izdelava prototipa za podrobnosti zasnove,
- oblikovanje in integracija sistemskih kontrol.

Cilj aktivnosti oblikovanja je zasnova sistema, ki temelji na opredeljenih sistemskih zahtevah, ki so bile ugotovljene med fazo analize. Oblikovanje je razdeljeno na dve ravni: visoko in nizko. Oblikovanje na visoki ravni vključuje razvoj arhitekturne strukture komponent programske opreme (podsistemov), baz podatkov, uporabniških vmesnikov in operacijskega okolja. Na nižji ravni pa gre za razvoj podrobnih algoritmov in podatkovnih struktur, ki so potrebne za razvoj programske opreme.

Izmed zgornjih sedmih aktivnosti je oblikovanje arhitekture aplikacije tisto, ki izpolnjuje zahteve končnega uporabnika v povezavi z opredeljenim problemom, saj nudi funkcije procesiranja za poslovne zahteve. Za oblikovanje se uporabljajo diagrami, ki smo jih razvili v fazi analize.

Ne glede na to, da je arhitektura aplikacije najbolj »vidna« uporabniku, pa ostale aktivnosti niso nič manj pomembne. Sistemski vmesniki na primer skrbijo za komunikacijo novega sistema z morebitnimi obstoječimi sistemi. Baza podatkov hrani podatke, sistemski kontrolniki pa skrbijo za varnost informacijskega sistema, ki je danes še posebej pomembna.

### **Aktivnosti implementacije:**

- zgradba komponent programske opreme,
- preverjanje in testiranje,
- pretvorba podatkov,
- usposabljanje uporabnikov in izdelava dokumentacije sistema,
- namestitev sistema.

V fazi implementacije je zgrajena, testirana in nameščena končna verzija sistema. Cilj te faze pa ni samo zanesljiv, v celoti delujoč informacijski sistem. Treba je tudi zagotoviti, da so uporabniki sistema ustrezno usposobljeni ter da je organizacija pripravljena na koristno uporabo sistema. Ta faza temelji na vseh prejšnjih aktivnostih.

### **Aktivnosti podpore:**

- vzdrževanje sistema,
- izboljšava sistema,
- podpora uporabnikov.

Glavni cilj aktivnosti podpore je produktivno izvajanje sistema po namestitvi in nudenje podpore uporabnikom sistema. Seveda se lahko sistem sčasoma izboljša ali se razširijo njegove funkcionalnosti.

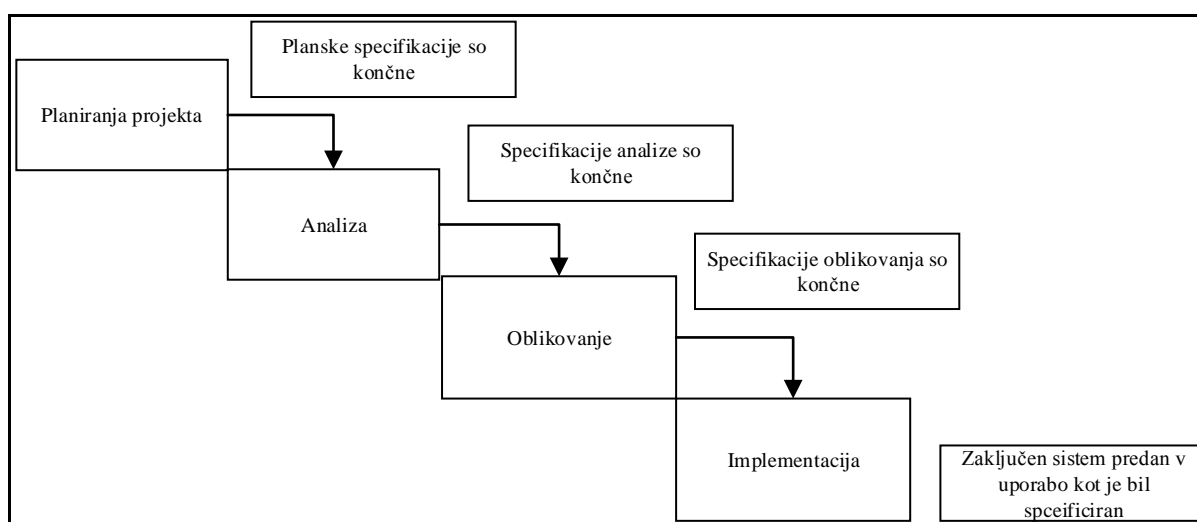
Post in Anderson (2003, str. 470–471) kot pomemben korak po zgornjih petih fazah navajata tudi vrednotenje. Treba je ugotoviti, kako učinkovit je bil določeni razvojni proces upoštevajoč primerjavo napovedanih in dejanskih stroškov, porabljeni čas, doseganje postavljenih ciljev in drugo. Dober sistem naj bi znižal stroške podjetja, povečal prihodek ali zagotovil konkurenčno prednost. Vendar to niso edini pokazatelji kakovosti novega sistema. Sistem mora biti preprost za uporabo, prilagodljiv spremembam v podjetju in zanesljiv.

Sistem vrednotenja je po Postu in Andersonu (2003, str. 471) treba določiti že na začetku projekta. Prednost tega je, da so vprašanja, ki služijo za vrednotenje sistema, lahko izredno koristna kot smernice med samim razvojem sistema.

Zadnjo oceno novega sistema pa naj bi kot za vsak drug končni izdelek podale stranke. V primeru informacijskega sistema so to uporabniki le-tega, zato je sistem dober, če so njegovi uporabniki prepričani, da izvaja koristna opravila (Post & Anderson, 2003, str. 471).

Zgoraj opisanih pet faz življenjskega cikla razvoja sistema, kot ga imenujejo Satzinger et al. (2009, str. 38), lahko izvajamo na dva različna načina. Prvi se imenuje **napovedni način** (angl. *predictive approach*), ki predvideva, da se vsaka faza življenjskega cikla razvojnega projekta izvede samo enkrat. Z drugimi besedami, razvojni projekt se izvede v skladu z načrtom. Najbolj striktna različica tega načina je tako imenovani slap, kjer se vsaka faza preliva v naslednjo do zaključka razvojnega projekta. Za napovedni način sta značilna dobro razumevanje sistemskih zahtev in majhno tehnično tveganje. Model slapa je prikazan na Sliki 2.

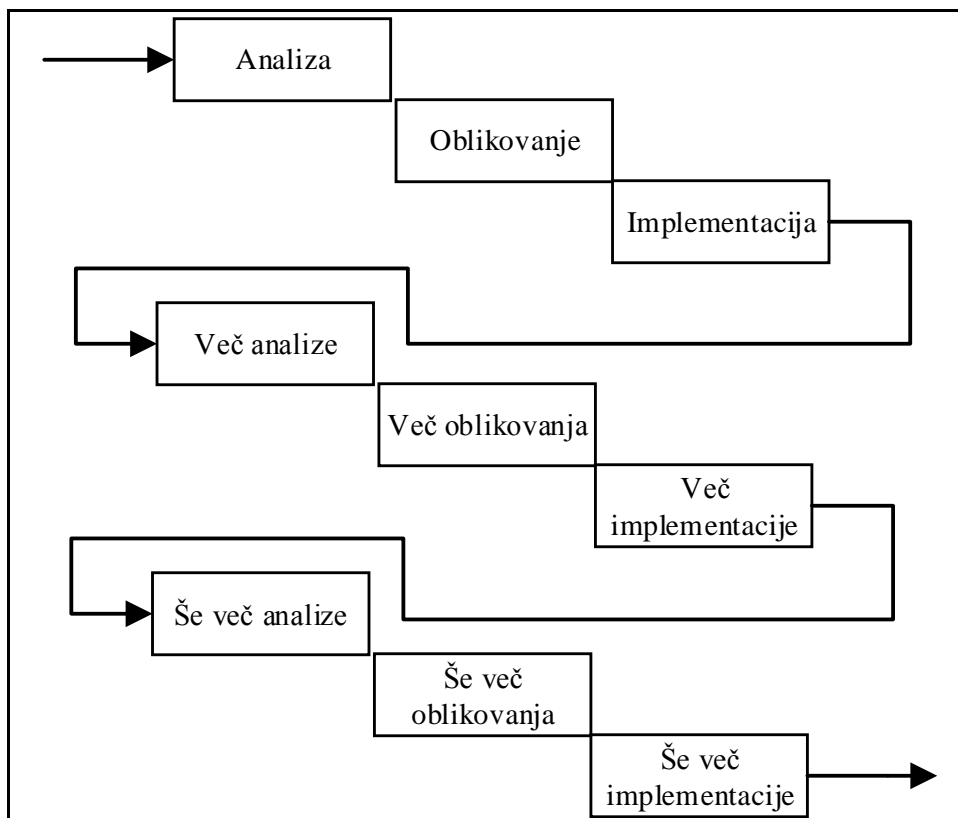
Slika 2: Model slapa življenjskega cikla razvoja sistema



Vir: Satzinger et al., *Systems Analysis and Design in a Changing World*, 2009, str. 41.

Drugi način pa se imenuje **prilagodljivi način** (angl. *adaptive approach*). Ta način je primeren, kadar sistemske zahteve vnaprej niso dobro poznane in ko razvojnega projekta ni mogoče vnaprej v celoti načrtovati. Vključuje visoko tehnično tveganje. Poleg tega se vse faze življenjskega cikla razvojnega projekta nenehno spreminjajo in prilagajajo. Najbolj razširjena različica tega načina je **spirala**. Ta način uporablja **iteracije**. To pomeni, da je celoten razvojni projekt razdeljen na množico manjših projektov oziroma problemov, ki jih je lažje rešiti. Tako se na primer najprej izvede vseh pet faz za prvi problem, nato drugi in tako do konca, dokler ni zaključen celoten razvojni projekt. Spirali z iteracijami je podobna **inkrementalna** različica, katere cilj je, da po nekaj iteracijah razvijalci predajo razviti del sistema v uporabo končnim uporabnikom. Nato po nekaj naslednjih iteracijah razvijejo drug del sistema, ga integrirajo s prvim delom in spet predajo v uporabo (Satzinger et al., 2009, str. 39–43). Ta različica je prikazana na Sliki 3.

Slika 3: Iteracija aktivnosti razvoja sistema



Vir: Satzinger et al., *Systems Analysis and Design in a Changing World*, 2009, str. 44.

V magistrskem delu bom za potrebe analize informacijskih potreb prevajalske agencije uporabil naslednje faze in aktivnosti:

- **načrtovanje projekta:** opredelitev problema,
- **analizo:** zbiranje informacij, opredelitev sistemskih zahtev, izdelava in vrednotenje različnih možnosti izvedbe.

Ker je namen magistrskega dela analiza informacijskih potreb in se ne bom opredeljeval za neko določeno strojno in programsko izvedbo, torej implementacijo, se ne bom ukvarjal z oblikovanjem in implementacijo.

V nadaljevanju bom podrobneje opredelil teoretično podlago za zgoraj navedeni fazi in aktivnosti, ki jih bom uporabil.

## 2.2 Dva pristopa k razvoju sistema

Satzinger et al. (2009, str. 53) navajajo dva različna pristopa k razvoju sistema: **tradicionalni** in **objektni**. Tradicionalnemu pristopu pravimo tudi strukturirani pristop, saj so zanj značilni strukturirana analiza, strukturirano oblikovanje in strukturiran program (programiranje). Med strukturirano analizo ugotavljamo sistemske zahteve na dva načina. Prvi je izražen z **diagramom**

**toka podatkov** (angl. *data flow diagram* – DFD), ki je grafični prikaz interakcije vložkov, procesov, hrambe in izločkov sistema. Drugi način pa se ukvarja s podatki, ki jih mora sistem shraniti in relacijami med temi podatki. Te relacije so prikazane z **diagramom relacij entitet** (angl. *entity relationship diagram* – ERD).

Za strukturirano oblikovanje je značilno, da je treba določiti nabor programov (modulov), cilje in namene vsakega programa ter njihovo hierarhično organiziranost. Ta je prikazana na diagramu strukture (angl. *structure chart*). Kakovost strukturiranega oblikovanja določamo na podlagi dveh značilnosti programov: nepovezanosti in visoke kohezivnosti. Programi morajo biti medsebojno čim bolj neodvisni, da jih lahko kasneje spreminjamo, ne da bi s tem vplivali na ostale programe. Kohezivnost pa pomeni, da vsak program služi enemu samemu jasnemu namenu (Satzinger et al., 2009, str. 55).

Za strukturirane programe je značilno naslednje (Satzinger et al., 2009, str. 53):

- imajo en začetek in en konec,
- vsak korak je sestavljen iz treh programskih sklopov:
  - zaporedje programskih stavkov,
  - odločitev, kjer se izvede en ali drug sklop stavkov,
  - ponavljanje sklopa stavkov.

Drugi pristop k razvoju sistema pa je objektni pristop. Objektni pristop se od tradicionalnega pristopa razlikuje po načinu obravnavanja informacijskega sistema. Vidi ga namreč kot zbirko objektov, ki sodelujejo pri izpolnitvi določene naloge. Gledano z vidika koncepta objektni pristop ne pozna procesov ali programov in tudi nima podatkovnih entitet ali datotek. Sestoji samo iz objektov. Objekt pa je predmet v računalniškem sistemu, ki se odziva na sporočila. Vse to zahteva drugačno analizo, oblikovanje in programiranje sistema (Satzinger et al., 2009, str. 59).

Objektna analiza opredeli vse tipe objektov, ki izvajajo opravila v sistemu, in pokaže, katere interakcije z uporabniki so potrebne za uspešno izvedbo nalog. Te interakcije imenujemo **primeri uporabe** (angl. *use cases*)<sup>1</sup>. Objektno oblikovanje nato opredeli dodatne tipe objektov, ki so potrebni za komunikacijo med uporabniki in napravami v sistemu. Poleg tega pokaže, kako objekti sodelujejo pri izpolnjevanju nalog in podrobneje opredeli tipe objektov, da so pripravljeni na implementacijo. S pomočjo objektnega programskega jezika pa opredelimo, kaj vsak tip objekta v sistemu dela. Objektni pristop klasificira podobne tipe objektov v razrede. Z diagramom razredov prikaže vse razrede v sistemu in interakcije med njimi. Posamezni razredi imajo lahko tudi podrazrede, kar pomeni, da imajo le-ti enake lastnosti kot razredi (Satzinger et al., 2009, str. 60).

---

<sup>1</sup> Primere uporabe opredelimo tudi pri tradicionalnem pristopu, ker so osnova za nadaljnje modeliranje sistemskih zahtev pri obeh pristopih, kot bomo videli v nadaljevanju.

Prednosti objektnega pristopa sta predvsem njegova naravnost in možnost ponovne uporabe. Objektni pristop z objekti odraža stvari iz realnega sveta in je zato lažje razumljiv kot zapleteni programski postopki pri strukturnem pristopu. Ponovna uporaba omogoča, da se objekti uporabijo na več različnih mestih v sistemu in jih ni treba ponovno določati (Satzinger et al., 2009, str. 60).

### 2.2.1 Načrtovanje projekta

Pri načrtovanju projekta je izrednega pomena opredelitev problema, saj so lahko v primeru, da ni natančna in pravilna, vse nadaljnje aktivnosti napačne ali pomanjkljive. Pri opredelitvi problema sledimo trem korakom. Najprej pregledamo poslovne zahteve, ki so bile povod za analizo in oblikovanje sistema. Na podlagi tega opredelimo **pričakovane poslovne koristi**. Seznam poslovnih koristi vsebuje rezultate, ki jih pričakujemo od novega sistema. Poslovne koristi se nanašajo na finančni vidik v smislu zmanjšanja stroškov ali povečanja prihodkov. Naslednji korak je opredelitev **pričakovanih zmožnosti** novega sistema. To nam pomaga razumeti obseg novega sistema in obseg samega projekta razvoja. Poslovne koristi in pričakovane zmožnosti novega sistema so povezane na način, da so prve rezultat drugih (Satzinger et al., 2009, str. 87).

### 2.2.2 Analiza

#### Zbiranje informacij

Po Satzingerju et al. (2009, str. 119–123) gre pri zbiranju informacij za razumevanje sistemskih zahtev in delovanja novega sistema, systemske zahteve pa so vse tiste zmožnosti in omejitve, ki jih mora nov sistem izpolnjevati. Shelly, Cashman, in Rosenblatt (2006, str. 101) navajajo podobno opredelitev za systemsko zahtevo, ki je karakteristika, ki mora biti vključena v informacijski sistem, da ta izpolni poslovne zahteve in je sprejemljiv za uporabnike.

Zahteve delimo na **funkcionalne** in **nefunkcionalne**. Funkcionalne izhajajo iz aktivnosti, ki jih mora sistem izvajati. Gre za zmožnosti novega sistema, ki jih opredelimo v fazi načrtovanja. Nefunkcionalne pa delimo na tehnične, zmogljivostne, uporabne, zahteve po zanesljivosti in varnostne zahteve. Te se nanašajo na lastnosti sistema, ki jih mora imeti sistem poleg tega, da izvaja prej omenjene aktivnosti. Primer varnostnih zahtev bi bil, da lahko do določenih podatkov dostopajo samo nekateri uporabniki, ne pa vsi (Satzinger et al., 2009, str. 119–123).

Funkcionalne zahteve običajno dokumentiramo z grafičnimi in opisnimi modeli, nefunkcionalne pa so samo spremljajoči opisi k funkcionalnim zahtevam. V magistrskem delu bom uporabil tako grafične kot opisne modele. Pri prvih gre za grafične prikaze v obliki diagramov, pri drugih pa za poročila, sezname ipd. (Satzinger et al., 2009, str. 126).

Modele delimo na **logične** in **fizične**. Logični model zelo podrobno prikazuje funkcionalnosti novega sistema, ne da bi vnaprej določeval tehnologijo za izvedbo sistema. S tem, ko se vnaprej ne opredelimo za določeno tehnološko izvedbo, imamo možnost, da se tem bolje posvetimo

sistemskim zahtevam. Fizični modeli pa so vezani na tehnološko obliko izvedbe. Običajno se za fazo analize uporabljajo podrobni logični modeli, za fazo oblikovanja pa fizični modeli (Satzinger et al., 2009, str. 120).

Glavni vir informacij za ugotavljanje sistemskih zahtev so različni deležniki novega sistema: končni uporabniki, naročniki razvoja sistema in tehnično osebje, ki bo skrbelo za delovanje in podporo novega sistema. V analizi se bom predvsem osredotočil na končne uporabnike kot glavne deležnike sistema (Satzinger et al., 2009, str. 122–131).

Ključnega pomena pri zbiranju informacij je razumevanje poslovnih procesov, da lahko izdelamo logični model novega sistema. Pri raziskovanju in razumevanju poslovnih procesov nas lahko vodijo spodnje smernice (Satzinger et al., 2009, str. 134):

- kateri poslovni procesi obstajajo,
- kako se poslovni procesi izvajajo,
- katere informacije so potrebne za njihovo izvajanje.

Valacich, George in Hoffer (2009, str. 164–180) metode zbiranja informacij razdelijo na tri sklope: tradicionalne, moderne in radikalne. Med tradicionalne spadajo intervjuji z ljudmi o tem, kako opravljajo svoje delo, neposredno opazovanje ljudi pri delu in analiza postopkov in dokumentacije. Med moderne metode štejeta skupni razvoj aplikacij ali JAD (angl. *Joint Application Design*) in izdelavo prototipa. Bistvo skupnega razvoja aplikacij je istočasno pridobivanje informacij od vseh ključnih deležnikov sistema, ki med intenzivnimi (lahko tudi večdnevnimi) sestanki določijo ali pregledajo systemske zahteve. Pri metodi izdelave prototipa pa gre za izdelavo osnovnega sistema na podlagi prvih zbranih informacij in zahtev. Uporabniki nato preizkusijo prvo verzijo sistema in podajo povratne informacije, ki jih analitiki nato uveljavijo v naslednji verziji. Cilj te metode je podpora določitvi specifičnih zahtev celotnega sistema, ne pa izdelava celotnega sistema. Sklop radikalnih metod predstavlja proces prenove poslovnih procesov (angl. *business process reengineering*), kjer gre za uveljavitev korenitih sprememb v poslovnih procesih, da se doseže pomembne izboljšave pri izdelkih ali storitvah.

Za zbiranje informacij bom uporabil naslednje tehnike:

- pregled poročil, obrazcev in opisov postopkov,
- opazovanje in dokumentiranje poslovnih procesov in
- preučitev obstoječih rešitev ponudnikov programske opreme.

Zadnji korak pri zbiranju informacij bo preučevanje obstoječih rešitev ponudnikov programske opreme, in sicer v smislu, kako oni rešujejo probleme, ki jih bom opredelil v fazi načrtovanja. Rešitve bom preučil s pomočjo javno dostopnih informacij na spletu.

## Opredelevanje sistemskih zahtev – splošno

V fazi opredelitve sistemskih zahtev uporabimo vse zbrane informacije in izdelamo različne logične modele sistemskih zahtev. Dva ključna koncepta te faze sta že prej omenjeni primer uporabe in stvari (angl. *things*) (Satzinger et al., 2009, str. 160).

### Primer uporabe

Primer uporabe je aktivnost, ki jo izvede sistem, in sicer običajno kot odziv na zahtevo uporabnika. Za določitev primerov uporabe poznamo več tehnik: tehnika cilja uporabnika (angl. *user goal technique*), CRUD-tehnika (CRUD je akronim za *Create, Read, Update and Delete* – **Kreiranje, Branje, Posodabljanje in Brisanje**) in tehnika razstavljanja dogodkov (angl. *event decomposition technique*) (Satzinger et al., 2009, str. 160–161).

Pri tehniki cilja uporabnika analitik opravi pogovore z vsemi uporabniki sistema, da bi ugotovil, katere cilje želijo doseči z uporabo sistema. Pri CRUD-tehniki gre za ugotavljanje, katere podatke sistem med svojim delovanjem shranjuje. Analitik nato pregleda vse tipe podatkov in primere uporabe, ki ustvarjajo, berejo, posodablajo ali brišejo podatke. Pri tretji tehniki – razstavljanju dogodkov – gre za določevanje dogodkov, na katere se mora sistem odzvati, in za določevanje načinov, na katere se sistem odzove. Pomembnost dogodkov pri ugotavljanju sistemskih zahtev je danes še toliko večja, saj večina sistemov deluje v realnem času in se v realnem času odziva na dogodke iz okolja (Satzinger et al., 2009, str. 160–161). Tehniko razstavljanja dogodkov bom uporabil za določitev primerov uporabe.

Pri določanju primerov uporabe moramo biti zelo pazljivi pri stopnji podrobnosti primera uporabe. Vnos imena stranke ne more biti primer uporabe, saj je preveč ozko usmerjen. Diametralno nasprotje temu bi bile celodnevne aktivnosti s stranko, kar pa je preveč široko in spet ni uporabno. Zlata sredina bi na primer bila dodajanje nove stranke v sistem. Temu rečemo osnovni poslovni proces (angl. *elementary business process* – EBP). To je opravilo, ki ga izvede ena oseba na enem mestu kot odgovor na poslovni dogodek in ki doprinese merljivo poslovno vrednost ter pusti sistem in njegove podatke v konsistentnem stanju (Satzinger et al., 2009, str. 161).

Dogodke delimo na zunanje (običajno jih sproži zunanji agent ali akter), časovne (ti se zgodijo v določenem trenutku) in na dogodke stanja oziroma interne dogodke (ti se zgodijo, ko nekaj v sistemu sproži procesiranje: stanje zaloge pade pod določeno mejo). Ko določamo primere uporabe, pa ne smemo vključevati dogodkov, ki so povezani s sistemskimi kontrolami (na primer pravilna prijava uporabnika), in neposredno ne zadevajo uporabnikov ali transakcij. Poleg tega moramo izločiti tudi dogodke, ki so vezani na delovanje tehnologije, kot je recimo izpad sistema in podobno (Satzinger et al., 2009, str. 163–167).



Vse dogodke zberemo v **tabeli dogodkov** in po Satzingerju et al. (2009, str. 169) za vsakega opredelimo:

- **sprožilec:** signal, ki sistemu sporoči, da se je zgodil dogodek v obliki podatkov za procesiranje ali pa časovno pogojen dogodek;
- **izvor:** zunanji agent ali akter, ki posreduje podatke sistemu;
- **primer uporabe:** aktivnost, ki jo sistem izvede;
- **odziv:** rezultat oziroma izložek, ki ga proizvede sistem in gre do cilja;
- **cilj:** zunanji agent ali akter, ki prejme podatke od sistema.

Da lahko bolj podrobno opredelimo potrebe po avtomatizaciji in informatizaciji sistema, moramo ločiti med izvorom (podatkov) in akterjem. Akter pride do izraza v UML-diagramih in je tista oseba, ki je v dejanski interakciji s sistemom. Zato izvor iz tabele dogodkov za določen primer uporabe ni vedno enak akterju v diagramu za isti primer uporabe (Satzinger et al., 2009, str. 171).

Poglejmo naslednji primer uporabe za oddajo naročila. Izvor podatkov je stranka, akter pa je uradnik, ki podatke o naročilu vnese v sistem. Če stranka sama odda naročilo prek spletnega mesta, pa je le-ta sama izvor in akter, saj sistemu sama posreduje podatke (Satzinger et al., 2009, str. 171).

Pogosto se zgodi, da ima posamezen primer uporabe več različnih možnih zaporedij korakov. Takšen je lahko zgornji primer oddaje naročila. Gre za isti primer uporabe, vendar pa so koraki različni. Take različice imenujemo scenariji ali instance primera uporabe (angl. *scenario* ali *use case instance*). Šele ko opredelimo vse možne scenarije, imamo pred seboj podrobno predstavo sistema (Satzinger et al., 2009, str. 171).

Larman (2005, str. 63) scenarije omenja že v sami opredelitvi, ko pravi, da gre pri primeru uporabe za sklop povezanih uspešnih ali neuspešnih scenarijev, ki opisujejo kako akter s pomočjo sistema doseže svoj cilj.

Opis vsakega primera uporabe podamo z opisom primera uporabe, ki je lahko samo povzetek, srednje podroben ali zelo podroben. **Povzetki** se uporabljajo v redkih primerih, ko je sistem, ki ga je treba razviti, relativno enostaven. Enostavni primeri uporabe bi na primer imeli samo en scenarij in zelo malo pogojev, ki opredeljujejo izjeme. Takšen povzetek je lahko v najbolj ekstremnem primeru tudi samo ena poved. **Srednje podroben opis** je sestavljen iz dveh delov. Prvi del vsebuje zaporedje vseh aktivnosti delovnega toka. Drugi del pa navaja morebitne pogoje izjem za posamezne aktivnosti. **Zelo podroben opis** pa vključuje: ime primera uporabe, scenarij, sprožilni dogodek, kratek opis, akterje, povezane primere uporabe, deležnike, predpogoje, zaključne pogoje, tok aktivnosti in pogoje izjem. Predpogoji določajo, kaj mora biti izpolnjeno, da se primer uporabe začne, zaključni pogoji pa določajo, kaj mora biti izpolnjeno ob zaključku primera uporabe (Satzinger et al., 2009, str. 172–175).

Prednosti rabe primerov uporabe pri opredelitvi sistemskih zahtev Larman (2005, str. 65) vidi predvsem v njihovi preprostosti, kar posledično omogoča, da se v proces lahko vključijo tudi drugi deležniki novega sistema in ne le analitiki. Kot navaja, majhna udeležba uporabnikov sistema pri razvoju le-tega predstavlja tveganje za uspeh razvoja. Druga prednost, ki jo izpostavlja pa je, da primeri uporabe v ospredje postavljajo cilje uporabnikov in njihove vidike. Poudarek je torej na uporabniku.

Ne glede na prej omenjeno preprostost primerov uporabe Larman (2005, str. 65) dodatno prednost vidi v tem, da so zelo prilagodljivi glede stopnje kompleksnosti in formalnosti.

## Stvari

Stvari predstavljajo vse tisto, o čemer mora informacijski sistem hraniti podatke. Če spet pogledamo zgornji poenostavljen primer, bi moral sistem shraniti podatke o stranki in naročilu. Pri tradicionalnem pristopu stvari postanejo podatkovne entitete, pri objektnem pristopu pa te stvari postanejo objekti, ki so v interakciji s sistemom. Ne glede na vrsto pristopa k razvoju sistema pa je izredno pomembno, da imamo opredeljene vse stvari, o katerih mora sistem hraniti podatke (Satzinger et al., 2009, str. 176).

Seznam stvari, ki jih mora sistem shraniti, naredimo v treh korakih (Satzinger et al., 2009, str. 178):

- v tabeli dogodkov poiščemo vse samostalnike,
- pregledamo razpoložljivo dokumentacijo, obstoječi sistem, obrazce in poročila ter na seznam dodamo stvari ali kategorije informacij, ki jih moramo shraniti,
- prečistimo seznam in zabeležimo zadeve, ki jih moramo še raziskati.

Pri prečiščevanju seznama si lahko pomagamo z naslednjimi vprašanji, da ugotovimo, ali moramo samostalnik (Satzinger et al., 2009, str. 178):

- vključiti:
  - ali je to enoznačna stvar, ki jo mora sistem poznati,
  - je v obsegu sistema, ki ga razvijamo,
  - si mora sistem zapomniti več kot eno teh stvari;
- izključiti:
  - je res sinonim za neko drugo stvar, ki smo jo opredelili,
  - je res samo rezultat, ki ga sistem proizvede iz drugih opredeljenih informacij,
  - je res samo vložek, ki shrani druge informacije, ki smo jih opredelili;

- podrobneje raziskati:
  - bo verjetno samo informacija (atribut) druge stvari, ki smo jo opredelili,
  - je to nekaj, kar bomo potrebovali, če se predpostavke spremenijo.

Za določevanje stvari in njihovih atributov lahko uporabimo tudi analizo CRC (angl. *Class, Responsibilities, and Collaborators*), ki pomaga pri določevanju razredov, njihovih odgovornosti in sodelovalnih razredov. Sodelovalni razredi predstavljajo dejanske povezave med razredi. Analizo CRC je treba vedno izvajati skupaj z drugimi metodami za določevanje stvari, sestavljena pa je iz dveh faz. Vsi trije elementi so predstavljeni na lističih, ki jih med prvo fazo analize (viharjenje možganov) analitiki in drugi deležniki izpolnjujejo in dopolnjujejo. Lističe lahko nalepimo na tablo, kjer lahko s črtami grafično prikažemo povezave med sodelovalnimi razredi. V drugi fazi (analiza zbranih informacij) pa analitiki in strokovnjaki problemskega področja izmed vseh navedenih stvari izluščijo tiste, ki bodo postale dejanski razredi in tiste, ki so samo atributi drugih stvari (Arlow & Neustadt, 2002, str. 136–137).

Ko določamo stvari, moramo določiti dve pomembni kategoriji: **razmerja med stvarmi** in prej omenjene **attribute stvari**. Razmerja med stvarmi so povezave med specifičnimi stvarmi, kot sta na primer stranka in naročilo. V tem primeru gre za razmerje, ko stranka odda naročilo. Razmerja lahko pokažem v različnih smereh. Razmerje, da je stranka oddala naročilo, gre v eno smer, v drugi smeri pa lahko rečemo, da je bilo naročilo oddano s strani stranke, odvisno od tega, kaj želimo poudariti. Pomembno je tudi število povezav. Stranka lahko odda večje naročilo, vsako naročilo pa ima lahko le eno stranko. V tradicionalnem pristopu temu pravimo kardinalnost (angl. *cardinality*), v objektnem pa večkratnost (angl. *multiplicity*) (Satzinger et al., 2009, str. 178–180).

Attribute stvari bi poenostavljeno lahko opredelili kot specifične informacije o določeni stvari. Ime, priimek, naslov in e-pošta so na primer vsi atributi določene stranke. Več o tem in o razmerjih med stvarmi bo govora v poglavju o bazah podatkov in podatkovnih modelih.

Tako tradicionalni kot objektni pristop k razvoju sistema dajeta velik pomen zahtevam **shranjevanja podatkov**. Kot je že omenjeno, stvari v tradicionalnem pristopu postanejo podatkovne entitete, ki jih sistem hrani. Pri tem pa je treba opredeliti posamezne podatkovne entitete, njihove attribute in razmerja med njimi. Vse to pa je prikazano v diagramu relacij entitet (Satzinger et. al, 2009, str. 182).

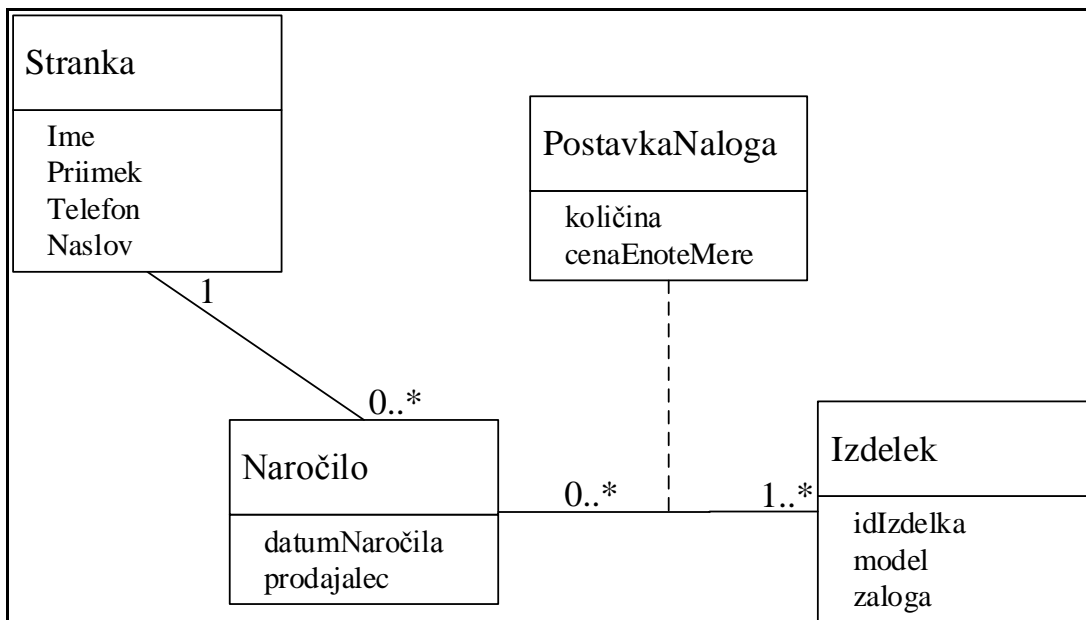
Tudi objektni pristop poudarja shranjevanje podatkov, vendar namesto o podatkovnih entitetah govori o objektih. Tudi ti imajo attribute in razmerja oziroma povezave. Pri objektnem pristopu objekte, njihove attribute in razmerja prikažemo z **diagramom razreda modela domene** (angl. *domain model class diagram*) ali krajše **diagramom razreda**. O razredih govorimo, ker z diagramom prikazujemo posamezne razrede objektov, o modelu domene pa zato, ker diagram prikazuje stvari, ki spadajo v domeno končnega uporabnika (Satzinger et al., 2009, str. 187).

Na Sliki 4 vidimo glavne značilnosti diagrama razreda modela domene. Simbol za razred domene je pravokotnik, ki je razdeljen na dva dela. V zgornjem delu je ime razreda, v spodnjem pa so atributi razreda. Povezave so ponazorjene s črtami med razredi, številke na črtah pri razredih pa določajo večkratnost povezav (Satzinger et al., 2009, str. 187).

Na Sliki 4 je prikazana še ena posebnost, ki se nanaša na povezave diagrama razreda, in sicer večkratnost vrste mnogo proti mnogo. Do te vrste večkratnosti pride, kadar je največja možna vrednost povezave na vsaki strani »mного«. V primeru na Sliki 4 to pomeni, da ima eno naročilo lahko enega ali mnogo izdelkov, en izdelek pa je lahko na nobenem ali mnogo naročilih. Vsakič, ko stranka odda naročilo, moramo vedeti, koliko enot izdelka je naročila in kakšna je cena (glede na popuste in podobno, se le-te seveda lahko razlikujejo). Ta dva atributa (količin in cena) sta rezultat povezave med razredoma naročila in izdelka. Zato jih prikažemo v **povezovalnem razredu** (angl. *association class*), ki je s črtkano črto vezan na povezavo med razredoma. Če iz povezave mnogo proti mnogo ne dobimo nobenih atributov, potem ni povezovalnega razreda (Jewett, 2006).

Po Arlow in Neustadt (2002, str. 158) je povezovalni razred povezava med dvema razredoma, ki je istočasno tudi sama razred. Poleg tega, da povezuje dva razreda, tudi opredeljuje nabor značilnosti, ki ji pripadajo. Na Sliki 4 je ta nabor značilnosti prikazan z atributoma količina in cena enote mere.

Slika 4: Diagram razreda modela domene – naročilo izdelka



Vir: T. Jewett, *Design pattern: many-to-many (order entry)*, 2013.

Pri razredih objektov je treba omeniti še hierarhijo generalizacije/specializacije, kjer gre za strukturiranje nadrejenih in podrejenih razredov, in dedovanje, kjer gre za dedovanje lastnosti nadrejenih razredov s strani podrejenih razredov (Satzinger et al., 2009, str. 189–190). To bom podrobneje obravnaval v poglavju o shranjevanju informacij in bazah podatkov.

## Opredelitev sistemskih zahtev – objektni pristop

Splošni del opredelitev sistemskih zahtev, kjer določimo vse primere uporabe, in stvari, ki jih mora sistem shranjevati v bazo podatkov, lahko nadaljujemo na že dva prej omenjena načina: tradicionalnega in objektnega. Vsak od teh načinov uporablja svoje modele za razumevanje in modeliranje sistemskih zahtev.

Objektni pristop vključuje več medsebojno povezanih modelov, kjer vsak obravnava svoj vidik novega sistema. Skupen model se imenuje **model primera uporabe** (angl. *use case model*) in je sestavljen iz naslednjih modelov (Satzinger et al., 2009, str. 242):

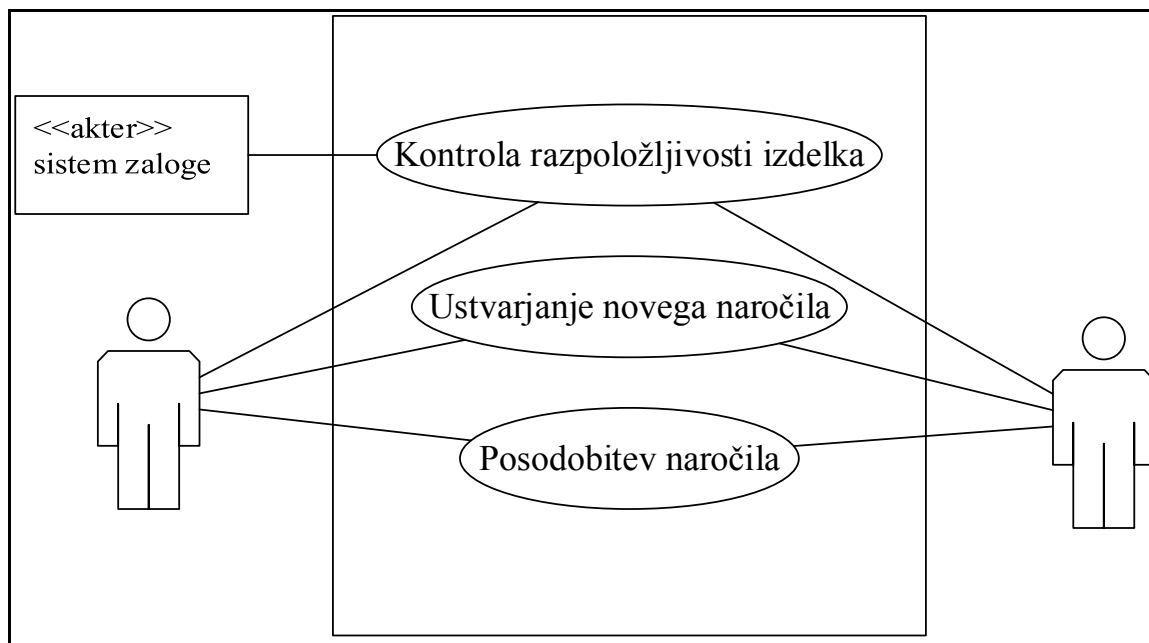
- diagram primerov uporabe (angl. *use case diagram*),
- opis primera uporabe (angl. *use case description*),
- diagram aktivnosti (angl. *activity diagram*),
- diagram zaporedja sistema (angl. *system sequence diagram*) in
- diagram stanja naprave (angl. *state machine diagram*).

Cilj modela primera uporabe je določiti in opredeliti vse osnovne poslovne procese, ki jih mora sistem podpirati. Primere uporabe lahko opredelimo na dveh ravneh: na ravni pregleda in na ravni podrobnosti. Pregled dobimo s tabelo primerov uporabe in z diagramom primerov uporabe. Podrobno pa primere uporabe opredelimo z opisom primera uporabe, z diagramom aktivnosti, diagramom zaporedja sistema ali pa s kombinacijo teh modelov.

Zgornje modele lahko razdelimo na tri dele, in sicer glede na to, katere vidike sistema pokrivajo. Diagram primerov uporabe, opis primera uporabe in diagrami aktivnosti opisujejo **vidik aktivnosti v sistemu**. Z diagramom zaporedja sistema določimo in prikažemo **tok informacij v sistemu** ter opredelimo vložke in izločke sistema. Poleg tega prikažemo tudi **interakcijo med akterji in sistemom**. Z diagramom stanja naprave pa opredelimo **obnašanje objektov** (Satzinger et al., 2009, str. 242–260).

Diagram primerov uporabe je grafični prikaz, ki ponazarja akterje in primere uporabe. Kot smo že omenili, je akter oseba, ki je v interakciji s sistemom. Poudariti pa je treba, da je akter vedno izven meje avtomatizacije sistema, kot je to prikazano na Sliki 5. Meja avtomatizacije ločuje okolje, kjer so akterji, in interne komponente sistema (Satzinger et al., 2009, str. 244).

Slika 5: Diagram primera uporabe

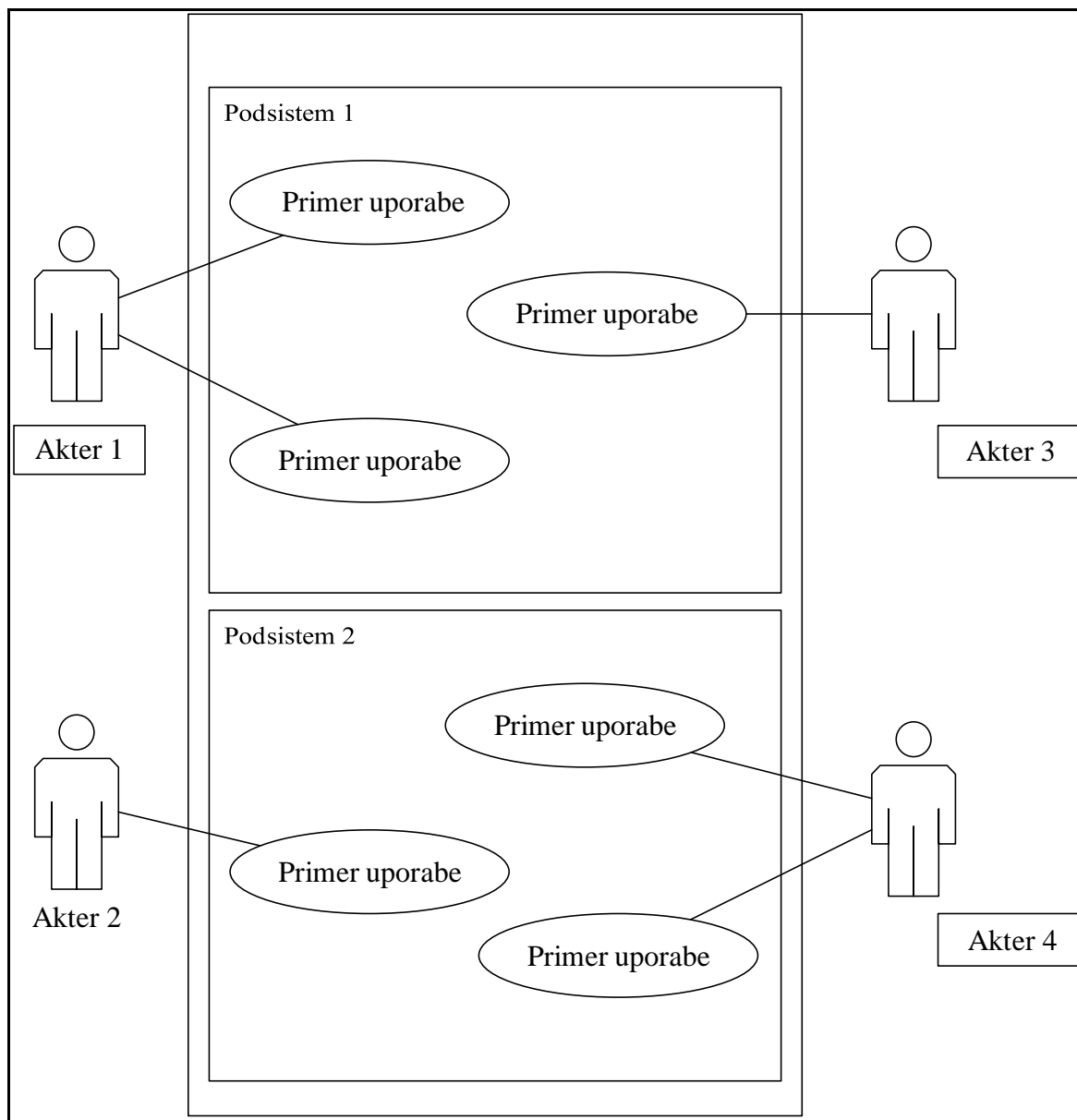


Vir: Satzinger et al., *Systems Analysis and Design in a Changing World*, 2009, str. 244.

Če želimo primer uporabe prikazati z vidika akterja, to lahko naredimo tako, da vse primere uporabe prikažemo kot paket znotraj meje avtomatizacije, kot je vidno na Sliki 6. Na sliki vidimo, da paketi predstavljajo podsisteme ali komponente sistema (Satzinger et al., 2009, str. 246).

Diagram primerov uporabe in tabela dogodkov ponujata enake informacije, vendar pa je med njima nekaj razlik. Prav tako se moramo odločiti, ali bomo za opredelitev sistemskih zahtev izdelali oba ali samo en model. Ena od razlik je ta, da se tabela dogodkov osredotoča na poslovne procese in dogodke, ki sprožajo te dogodke, torej na zunanje okolje. Diagram primerov uporabe pa daje poudarek na avtomatiziran sistem in na akterje, za katere ni nujno, da so začetniki poslovnega procesa. Vsekakor pa je pomembno, da med procesom analize stalno prečiščujemo in prenavljamo tabelo dogodkov in diagram primerov uporabe ali pa kar oboje, odvisno od tega, za kaj smo se odločili. Pogosto se na primer šele kasneje izkaže, da lahko dva primera uporabe združimo v enega ali pa moramo enega razdeliti na več delov zaradi njegove kompleksnosti (Satzinger et al., 2009, str. 247–248).

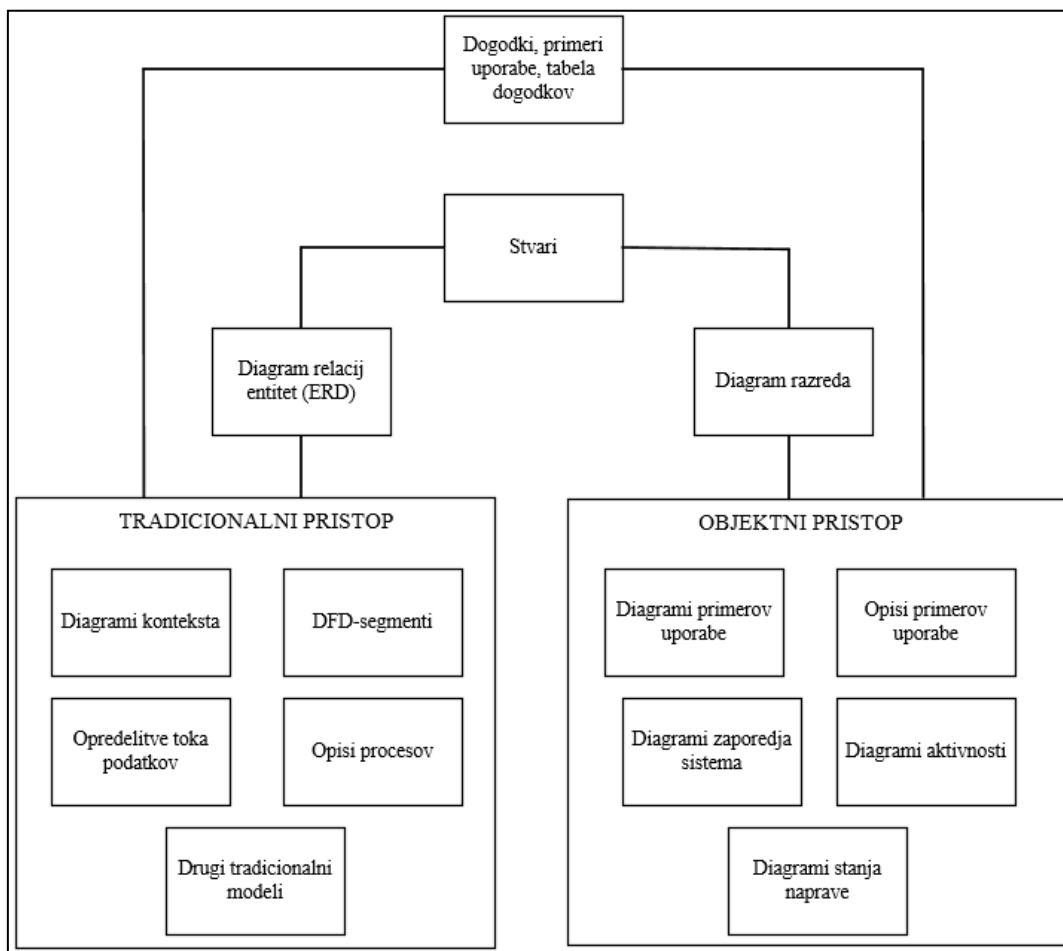
Slika 6: Diagram primera uporabe, organiziran po podsistemih



Vir: Satzinger et al., *Systems Analysis and Design in a Changing World*, 2009, str. 246.

Kot je prikazano na Sliki 7 bom najprej pripravil tabelo dogodkov, ki bo osnova za izdelavo diagrama primerov uporabe. Oba modela bosta predstavljala pregled celotnega sistema in vseh osnovnih poslovnih procesov. Primere uporabe bom opisal s podrobnimi opisi primerov uporabe in ponazoril z diagrami primerov uporabe.

Slika 7: Diagram zahtev za tradicionalni in objektni pristop



Vir: Satzinger et al., *Systems Analysis and Design in a Changing World*, 2009, str. 241.

V magistrskem delu bom uporabil naslednje aktivnosti in modele za opredelitev sistemskih zahtev:

- opredelitev sistemskih zahtev – splošno:
  - opredelitev dogodkov in primerov uporabe v tabeli dogodkov,
  - opredelitev stvari,
  - izdelava diagramov razredov;
- opredelitev sistemskih zahtev – objektni pristop:
  - opisi primerov uporabe,
  - diagrami primerov uporabe.



## **Izbira med možnostmi implementacije**

Programske rešitve lahko implementiramo na več načinov. Če je rešitev bolj ali manj standardna, potem podjetje lahko kupi računalniški program, ki podpira tako rešitev. To je možno tudi, če gre za bolj kompleksno in zapleteno rešitev. Lahko pa se podjetje odloči, da samo izdelava računalniški program ali pa da izdelavo v zunanje izvajanje. Slika 8 prikazuje matriko z različnimi možnostmi nakupa, izdelave v hiši, v zunanjem izvajanju ali kot kombinacijo teh možnosti (Saztinger et al., 2009, str. 297–300).

**Upravljanje infrastrukture** – pri tej možnosti ne gre za nakup, ampak za dolgoročno strateško odločitev podjetja, da vse procesiranje in informatizacijo odda v zunanje izvajanje. Povedano drugače, zunanje podjetje postane informacijski oddelek tega podjetja ter mu nudi strojno in programsko opremo, procesiranje podatkov in tehnično osebje (Saztinger et al., 2009, str. 298).

**Rešitve ponudnikov storitev** – tudi pri tej možnosti gre za dolgoročno strateško odločitev podjetja, vendar pa v tem primeru podjetje najame/kupi tiste tehnološke storitve, ki jih potrebuje. Manjše podjetje na primer od ponudnika storitev kupi storitve v obliki spletne trgovine z vsemi podpornimi storitvami (oddajanje naročil, procesiranje podatkov, plačila itd.) (Saztinger et al., 2009, str. 298).

**Sistemi na ključ** – za le-te je značilno, da podjetje kupi celoten sklop, torej tako strojno kot tudi programsko opremo ter takoj začne uporabljati obe. Običajno se ponudniki takih sistemov specializirajo za določeno panogo. Slaba stran takšnih sistemov je, da pogosto ne izpolnjujejo vseh zahtev podjetja, kar ima za posledico, da podjetje prilagaja svoje procese takemu sistemu. Zato večina podjetij kupi osnovni sistem, ponudnik pa nato analizira specifične sistemske zahteve podjetja in izvede ustrezne prilagoditve (Saztinger et al., 2009, str. 298–299).

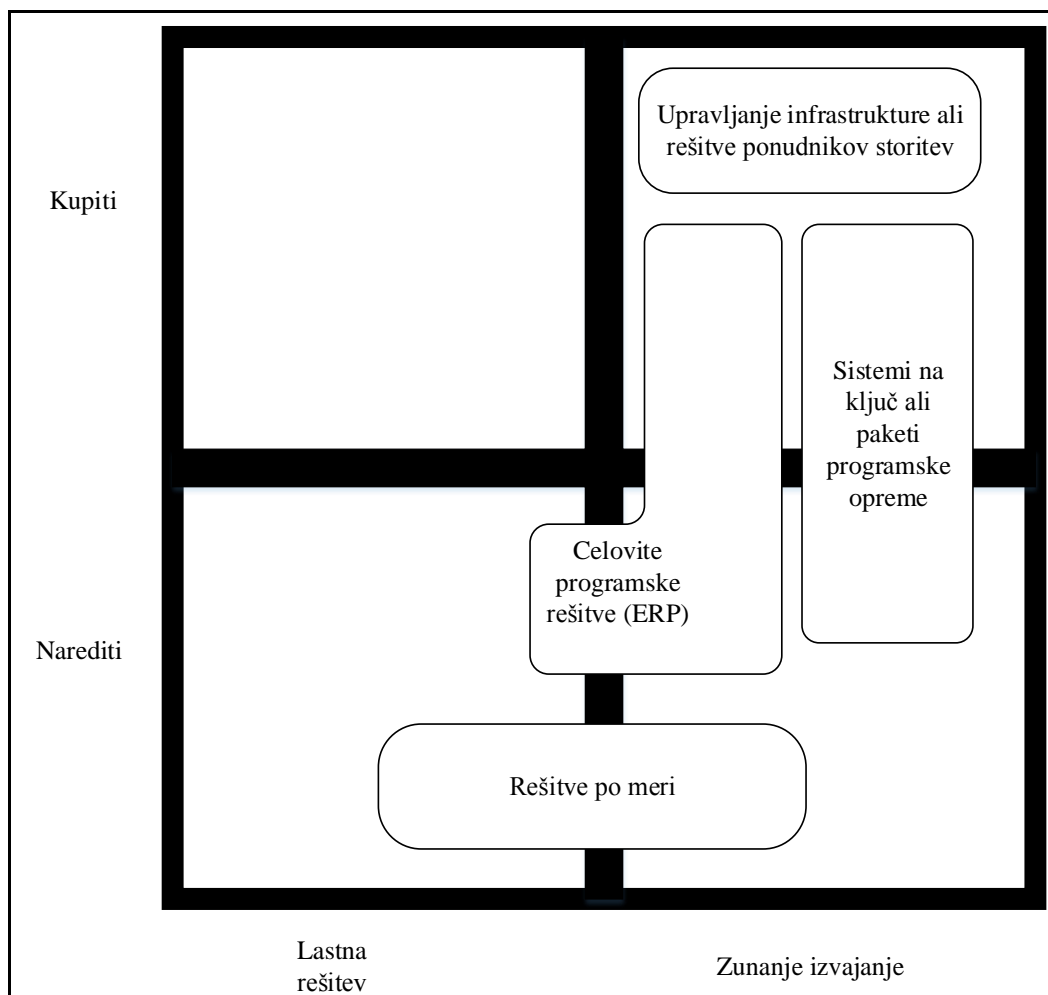
**Paket programske opreme** – le-ta se uporablja, kot je bila kupljena. Običajno je ne spreminjamo ali dodajamo novih zmožnosti uporabe. Primer takšne programske opreme so pisarniški paketi, ki jih vsakodnevno uporabljamo na naših računalnikih. Vsako podjetje ime določeno število takšnih paketov (Saztinger et al., 2009, str. 298).

**Celovite programske rešitve** – le-te podpirajo vse poslovne operacije v organizaciji. Njihova slaba stran je, da so namestitve časovno in stroškovno izredno zahtevne. Prednost celovite programske rešitve pa je vseeno ta, da organizacija pride do rešitve po nižjih stroških in z manjšim tveganjem, kot če bi rešitev razvijala sama, saj so osnovni sistemi dobro razviti in testirani. O njihovi učinkovitosti priča tudi veliko zgodb o uspehu. Slabosti so podobne kot pri sistemih na ključ. Če celovita programska rešitev ne ustreza organizaciji, se mora slednja prilagajati celoviti rešitvi, tako da spremeni svoje postopke in ustrezno usposobiti svoje zaposlene (Saztinger et al., 2009, str. 299).

**Rešitve po meri** so delno ali v celoti razvite v zunanjem izvajanju in so podrobno prikrojene potrebam podjetja. Prednost teh rešitev je ogromna količina znanja in izkušenj, ki jih ima

razvijalsko podjetje, ki je po vsej verjetnosti izdelalo že več takšnih rešitev za isto panogo. Slaba stran teh rešitev pa so nedvomno stroški. Za zunanje izvajanje se podjetja odločijo, kadar nimajo internih zmožnosti in znanja ali pa imajo na voljo kratek čas za razvoj sistema (Satzinger et al., 2009, str. 299–300).

Slika 8: Možnosti implementacije



Vir: Satzinger et al., *Systems Analysis and Design in a Chnging World*, 2009, str. 297.

Shelly et al. (2006, str. 206) navajajo nekaj dejavnikov, ki so prikazani v Tabeli 3, ki jih podjetja upoštevajo, ko se odločajo za lasten razvoj ali nakup programske rešitve.

Tabela 3: Lasten razvoj ali nakup

RAZLOGI ZA LASTEN RAZVOJ	RAZLOGI ZA NAKUP
Izpolnitev specifičnih poslovnih zahtev	Nižji stroški
Minimalne spremembe poslovnih procesov in politik	Za implementacijo je potrebnega manj časa
Skladnost z omejitvami obstoječih sistemov	Dokazana zanesljivost in ocena uspešnosti
Skladnost z omejitvami obstoječe tehnologije	Potrebna je manj tehničnega razvojnega osebja
Razvoj lastnih zaposlenih in sposobnosti	Prihodnje nadgradnje zagotavlja dobavitelj
Izpolnitev specifičnih varnostnih zahtev	Pridobitev informacij od drugih podjetij, ki uporabljajo to programsko opremo

Vir: Shelly et al., *Systems analysis and design*, 2006, str. 206.

Pri izbiri novega sistema in pri načinu implementacije ne smemo pozabiti, da se ne odločamo samo med »izdelati ali kupiti« ali »zunanje izvajanje«, ampak obstaja veliko možnih kombinacij. Zato moramo določiti nabor splošnih kriterijev, s katerimi lahko primerjamo različne alternative. Kriteriji morajo pokrivati splošne, tehnične in funkcionalne zahteve. Vsakemu kriteriju dodelimo ponderirano vrednost za pomembnost in oceno izpolnjevanja zahteve. Končno vrednost za vsak kriterij dobimo tako, da pomnožimo ponderirano vrednost z oceno. Vsota končnih ocen je končni rezultat vsake možnosti implementacije v primerjalni tabeli in služi kot osnova za izbiro možnosti. Tabela 4 je skrčen vzorec takšne primerjalne tabele (Satzinger et al., 2009, str. 304).

Tabela 4: Kriteriji za ocenitev možnosti implementacije

Kriteriji funkcionalnih zahtev	Ponder (od 1 do 5)	Možnost 1 – lastna rešitev		Možnost 2 – paket progr. opreme + sprememba	
		Osnovno	Razširjeno	Osnovno	Razširjeno
Poizvedba po izdelkih	4	5	20	4	16
Ustvarjanje naročila stranke	5	5	25	5	25
Sprememba naročila	4	5	20	5	20
Poizvedba po naročilih	4	5	20	5	20
Pakiranje naročila	5	5	25	5	25
Odprema naročila	5	5	25	5	25
Skupaj			135		131

Vir: Satzinger et al., *Systems Analysis and Design in a Changing World*, 2009, str. 302.

## Izbrane aktivnosti za izvedbo analize informacijskih potreb

V Tabeli 5 sem zbral aktivnosti, ki jih bom uporabil v sklopu magistrskega dela za izdelavo analize informacijskih potreb prevajalske agencije s stalnim prevajalskim timom.

*Tabela 5: Uporabljene aktivnosti v življenjskem ciklu razvoja informacijskega sistema*

Glavna faza	Podfaza	Aktivnosti
1. Načrtovanje projekta	1.1.Oprelitev problema	1.1.1. Opis problema
		1.1.2. Navedba pričakovanih poslovnih koristi
		1.1.3. Navedba pričakovanih zmožnosti novega sistema
2. Analiza	2.1.Zbiranje informacij	2.1.1. Pregled poročil, obrazcev in opisov postopkov
		2.1.2. Opazovanje in dokumentiranje poslovnih procesov
		2.1.3. Preučitev obstoječih rešitev ponudnikov programske opreme
	2.2.Oprelitev sistemskih zahtev – splošno	2.2.1. Oprelitev dogodkov in primerov uporabe v tabeli dogodkov
		2.2.2. Oprelitev stvari
		2.2.3. Diagrami razredov
	2.3.Oprelitev sistemskih zahtev – objektni pristop	2.3.1. Opisi primerov uporabe
		2.3.2. Diagrami primerov uporabe

### 2.2.3 Shranjevanje informacij in baze podatkov

Eden pomembnih vidikov informatizacije poslovanja je tudi shranjevanje informacij in podatkov, ki jih uporablja in posreduje informacijski sistem. Podatki so shranjeni v bazah podatkov, ki bodo v nadaljevanju podrobneje predstavljene.

Grad in Jaklič (1996, str. 1) za bazo podatkov navajata naslednjo opredelitev: »Baza podatkov (BP) je zbirka (množica) medsebojno povezanih operativnih podatkov, ki so shranjeni v računalnikovem pomnilniku brez nepotrebne podvajanja na način, ki omogoča njihovo

uporabo različnim uporabnikom z različnimi potrebami glede uporabe. Podatki so shranjeni tako, da so neodvisni od programov, ki jih uporabljajo».

Po Satzingerju et al. (2009, str. 488) je baza podatkov integrirana zbirka shranjenih podatkov, ki jo centralno upravljamo in nadzorujemo. Vsebuje tipe entitet ali razredov, njihove attribute in razmerja med njimi.

Namen oziroma cilj baze podatkov, kot jo razumemo po zgornjih opredelitvah, je, da na eni strani hrani podatke za izvajanje poslovnih operacij in na drugi strani, da poslovodstvu nudi podatke v podporo pri načrtovanju, spremljanju in odločanju. V veliko organizacijah so baze podatkov sestavni del celovitih programskih rešitev, ki nudijo podporo poslovanju (Gradišar et al., 2007, str. 88–89).

Turban, Rainer in Potter (2001, str. 138) navajajo, da baze podatkov v nasprotju s predhodnim načinom hranjenja podatkov v več ločenih podatkovnih datotekah, ki so bile vezane na posamezne računalniške programe, rešujejo naslednje probleme hrambe in rabe podatkov:

- odvečni podatki: gre za podvojene podatke, ki nastajajo v času zaradi rabe različnih podatkovnih datotek v različnih računalniških programih,
- nekonsistentni podatki: nekonsistentni podatki so lahko posledica odvečnih podatkov, ker se sčasoma podatki v različnih podatkovnih datotekah ne ujemajo več,
- izolirani podatki: do podatkov je težko dostopati iz različnih računalniških programov, če so shranjeni v podatkovnih datotekah in poleg tega namenjeni samo določenemu računalniškemu programu,
- varnost: to je še eno področje, ki ga je težko vzdrževati, če so podatki shranjeni v podatkovnih datotekah,
- celovitost (integriteta) podatkov: gre za omejitve podatkov, kot so oblika vnosa podatkov in druge omejitve, ki jih je težko uveljavljati v več podatkovnih datotekah,
- neodvisnost računalniških programov in podatkov: zasnova in razvoj računalniških programov naj ne bi smela biti pogojena z načinom hrambe podatkov. Pri podatkovnih datotekah so le-te odvisne od računalniških programov in obratno.

Jaklič (1996, str. 4–5) navaja še naslednje cilje uporabe baze podatkov:

- večstranost v prikazovanju odnosov med podatki: podatki morajo biti organizirani tako, da so vidni odnosi med njimi,
- zagotovitev večnamenske uporabe: isti podatki so na voljo različnim uporabnikom v različne namene,
- zmanjšanje stroškov z enostavno uporabo,
- preglednost: uporabniki imajo hiter pregled nad podatki, ki so jim na razpolago,
- zavarovanje podatkov pred uničenjem,
- vpeljava standardov za lažjo izmenjavo podatkov in poenostavljeno vzdrževanje,
- zvezen prehod s stare na novo sistemsko programsko opremo, ki omogoča uporabo tudi stare baze podatkov,

- zagotovitev prenašanja podatkov med nosilci,
- enostavna predstavitev logične zgradbe podatkov.

## Upravljanje z bazami podatkov

Za upravljanje in nadzor baz podatkov se uporablja računalniški program ali skupina računalniških programov, poznan kot sistem za upravljanje z bazami podatkov ali SUBP (angl. *database management system* – DBMS). SUBP omogoča hrambo podatkov na eni lokaciji, kjer jih je mogoče posodablјati in do njih dostopati prek različnih računalniških programov ali delov računalniškega programa. Poleg tega zagotavlja tudi mehanizme za vzdrževanje celovitosti shranjenih informacij, za zagotavljanje varnosti in ustreznih pravic dostopov uporabnikov, za obnavljanje informacij ob sesutju sistema idr. (Turban et al., 2001, str. 145). Zagotavlja torej doseganje zgoraj navedenih ciljev uporabe baze podatkov.

Glede na vlogo uporabnika, SUBP rešuje zahteve po prikazu istih podatkov za različne namene in po različnih pravicah dostopov z uporabo dveh različnih pogledov: s fizičnim in logičnim. Fizični pogled je namenjen upravljavcem ali administratorjem baz podatkov in predstavlja dejansko fizično lokacijo in urejenost podatkov v bazi. Logični pogled pa končnim uporabnikom skozi računalniške programe prikaže podatke na način, ki je ustrezen njihovemu namenu uporabe. Ena od prednosti SUBP-ja je, da ima lahko en fizični in veliko logičnih pogledov (Turban et al., 2001, str. 145–146).

Na Sliki 9 je prikazana interakcija med sistemom za upravljanje z bazami podatkov, bazo podatkov in njenimi uporabniki. SUBP to interakcijo uravnava s svojimi štirimi sestavnimi deli: programskim vmesnikom, procesorjem poizvedb za končnega uporabnika, administrativnim vmesnikom ter logiko nadzora in dostopa do baze podatkov. Baza podatkov je sestavljena iz fizične shrambe podatkov in sheme. Prva vsebuje grobe računalniške zapise podatkov, ki jih ustvari in uporablja informacijski sistem. Druga pa vsebuje informacije o podatkih v fizični shrambi in vključuje:

- kontrole dostopov in vsebine vključno z dovoljenimi vrednostmi za določene podatkovne elemente, odvisnostmi med temi vrednostmi in sezname vseh uporabnikov, ki lahko berejo ali posodablјajo vsebino podatkovnih elementov,
- odnose med podatkovnimi elementi ali skupinami le-teh,
- podrobnosti organiziranosti fizične shrambe podatkov (vrsta in dolžina podatkovnih elementov, lokacija podatkovnih elementov idr.) (Satzinger et al., 2009, str. 488–489).

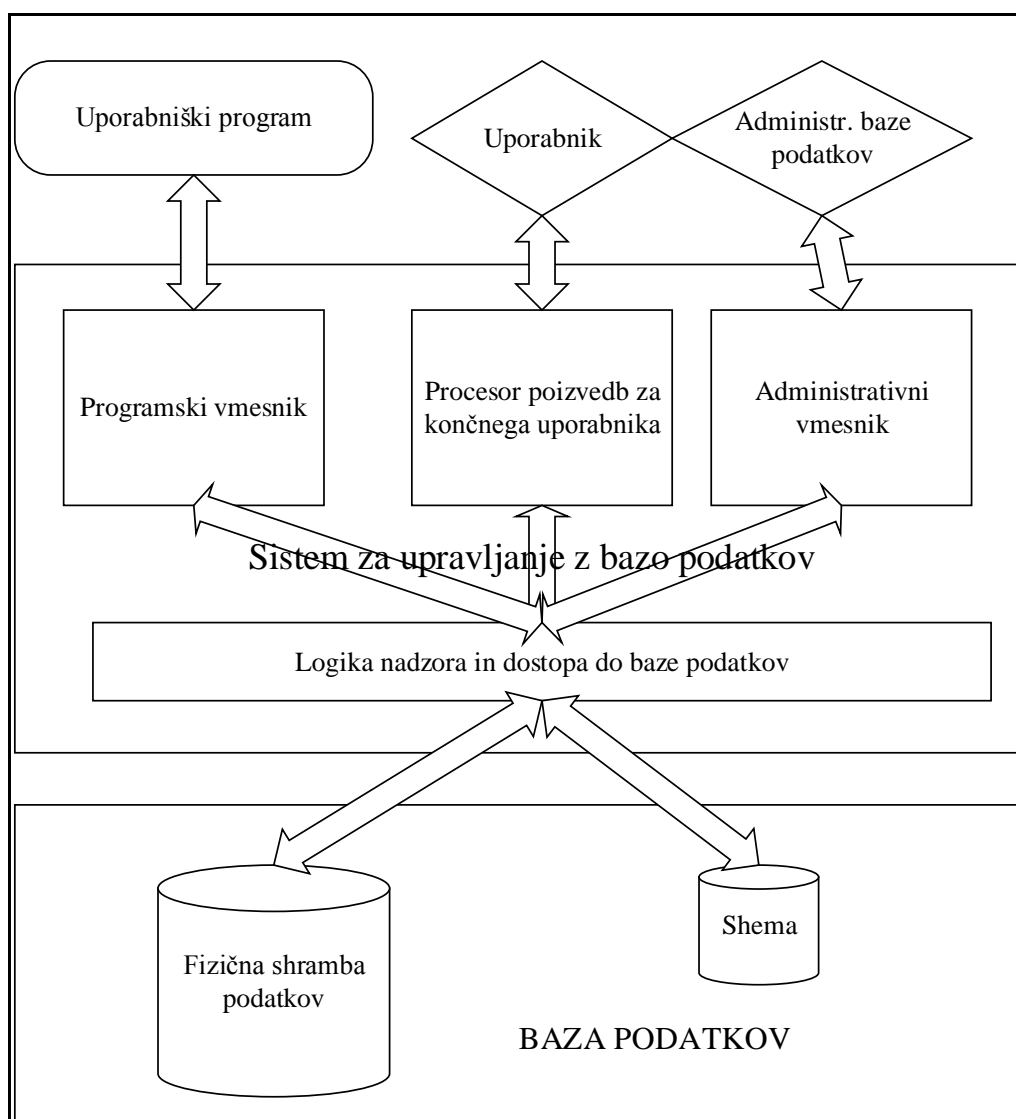
SUBP ima štiri glavne sestavne dele: programski vmesnik, procesor poizvedb za končnega uporabnika, administrativni vmesnik ter sklop programov in podrutin za dostop do podatkov. Uporabniki baze podatkov (uporabniški programi, uporabniki in administratorji baze podatkov) ne dostopajo neposredno do fizične shrambe, ampak do podatkov dostopajo prek ustreznega SUBP-vmesnika, ki preveri, ali zahtevani podatki obstajajo v bazi in ali ima uporabnik ustrezne pravice za dostop do zahtevanih podatkov. Če je zahteva veljavna, SUBP pridobi informacije o

organiziranosti zahtevanih podatkov iz sheme, nato pa do njih dostopa v fizični shrambi podatkov v imenu uporabnika (Satzinger et al., 2009, str. 489).

Baze podatkov in sistemi za upravljanje z njimi po Satzingerju et al. (2009, str. 489) nudijo več pomembnih zmožnosti pri dostopanju do podatkov in do upravljanja z njimi; med zmožnostmi so ključne naslednje:

- istočasen dostop več uporabnikov in uporabniških programov do podatkov,
- dostop do podatkov brez pisanja uporabniških programov (brez pisanja poizvedb),
- uporaba enotnih in konsistentnih kontrol dostopov in vsebine.

Slika 9: Interakcija med sistemom za upravljanje baz podatkov, bazo podatkov in njenimi uporabniki



Vir: Satzinger et al., *Systems Analysis and Design in a Changing World*, 2009, str. 489.

Danes sta najbolj uveljavljena relacijski (v nadaljevanju SURBP) in objektni (v nadaljevanju SUOBP) sistem za upravljanje z bazo podatkov. Prvi je namenjen relacijskim bazam podatkov,

drugi pa objektnim bazam podatkov. Zgornji opis velja za SURBP, glavne značilnosti za SUOBP pa bom navedel v sklopu objektnega modela podatkov in procesov kot enega izmed uveljavljenih podatkovnih modelov.

## **Podatkovni modeli**

Podatkovni modeli določajo, kako so podatki v bazi podatkov strukturirani. Dve sestavini podatkovnega modela, ki ga opredeljujeta, sta že zgoraj opisana shema in podatkovni zapis. Shema je konkretna uporaba podatkovnega modela. To pomeni, da je shema načrt neke baze podatkov, ki sledi pravilom podatkovnega modela. Podatkovni zapis pa je skupina vrednosti atributov, ki opredeljujejo isto entiteto ali povezavo v bazi podatkov in so tam shranjeni skupaj (Jaklič, 2002, str. 87).

Hoberman (2007, str. 28–29) za podatkovni model poda izredno zanimivo razlago oziroma primerjavo. Pravi, da ima podatkovni model podobno vlogo kot zemljevid. Tako podatkovni model kot zemljevid sta orodji za iskanje poti (angl. *wayfinding*). Model je torej skupek simbolov in besedil, ki omogoča lažje razumevanje zapletenih konceptov. Enako kot zemljevid poenostavlja zapleteno geografsko področje, tako podatkovni model poenostavlja zapleteno informacijsko področje.

Poleg tradicionalne rabe podatkovnega modela med fazama analize in zasnove nove aplikacije, da zagotovimo razumevanje potreb take aplikacije pred dejanskim ustvarjanjem baze podatkov, pa Hoberman (2007, str. 41–42) navaja še naslednje scenarije, kjer lahko uporabimo podatkovni model:

- za razumevanje obstoječe aplikacije,
- za izvedbo analize učinka,
- za razumevanje poslovnega področja,
- za izobraževanje novih članov projektnih timov.

Podatkovni modeli se delijo na hierarhične, mrežne, relacijske, večdimenzionalne in objektne modele podatkov. Treba je omeniti, da se danes hierarhični in mrežni ne uporabljata več, pri objektnem modelu pa so vključeni tudi procesi, kot bo razvidno iz podrobnejšega opisa.

### **Relacijski podatkovni model**

Za relacijski podatkovni model je značilno, da so podatki organizirani v relacijske podatkovne strukture, ki se odražajo v dvodimenzionalnih tabelah. V eni tabeli so shranjeni podatki o vseh entitetah enega entitetnega tipa. Vsaka vrstica v tabeli predstavlja eno entiteto, vsak stolpec pa en atribut te entitete. Za stolpec se uporablja tudi ime polja, za vrstico po podatkovni zapis. Glavni ključ tabele je tisti atribut entitetnega tipa, ki v tabeli ne more pripadati dvema entitetama. Njegova naloga je, da razlikuje posamezne entitete v tabeli (Jaklič, 2002, str. 93–95).



Dve relaciji v relacijskem podatkovnem modelu lahko povežemo na dva načina:

- s skupnim atributom, kadar gre za povezave tipa »ena proti ena« ali »ena proti mnogo«,
- z novo relacijo, kadar gre za povezave tipa »mного proti mnogo«.

Pri prvem načinu v eno relacijo dodamo drugo relacijo kot nov atribut, pri drugem pa tvorimo novo relacijo, ki predstavlja povezavo med obema obstoječima relacijama (Jaklič, 2002, str. 98–99).

Prednosti relacijskega podatkovnega modela so naslednje:

- je preprost za razumevanje,
- temelji na matematični teoriji, tj. teoriji relacij,
- za poizvedovanje uporablja standardni poizvedovalni jezik SQL.

Jaklič (2002, str. 100–105) med slabosti šteje:

- počasnejše poizvedovanje kot pri drugih podatkovnih modelih,
- velika kompleksnost relacijskih shem za relativno majhna problemska področja,
- neuporabnost pri podpori za poslovno odločanje.

Večdimenzionalni podatkovni model

Zgornje pomanjkljivosti relacijskega podatkovnega modela odpravlja večdimenzionalni podatkovni model. Omogoča organiziranje podatkov v obliki večdimenzionalne kocke. Stranice kocke so dimenzije, elementi kocke pa so mere. Na spodnjem primeru je mera vrednost prodaje za izbrane dimenzije izdelek, regija in čas (Jaklič, 2002, str. 106).

Večdimenzionalni podatkovni model omogoča, da v večdimenzionalni bazi podatkov hitro najdemo odgovore na vprašanja, kot je: »Koliko prevajalskih enot je bilo realiziranih za stranko X v letu 2012?«

Vsaki dimenziji kocke pripada ena tabela – dimenzijska tabela. Podatki o vrednostih mer za posamezno kombinacijo dimenzij pa so shranjeni v tabeli dejstev. Tabela dejstev torej povezuje dimenzijske tabele (Jaklič, 2002, str. 107).

Prednosti večdimenzionalnega podatkovnega modela so po Jakliču (2002, str. 109) predvsem:

- je preprost za razumevanje,
- primeren za podporo pri odločanju,
- iskanje podatkov je hitrejše kot pri relacijskem modelu.

## Objektni model podatkov in procesov

Za objektni podatkovni model, pravilneje objektni model podatkov in procesov, je značilen tako imenovani objektni pristop, saj poleg modeliranja podatkov vključuje tudi modeliranje procesov. Predpostavka objektnega modela je, da prikaže čimbolj realno sliko dejanskega sveta, kar pomeni, da vsak objekt realnega sveta predstavi z objektom v modelu. Ena od bistvenih lastnosti objektnega modela, ki ga razlikuje od na primer relacijskega, je, da v podatkovni model uvaja tudi obnašanje objekta (Jaklič, 1996, str. 115–117).

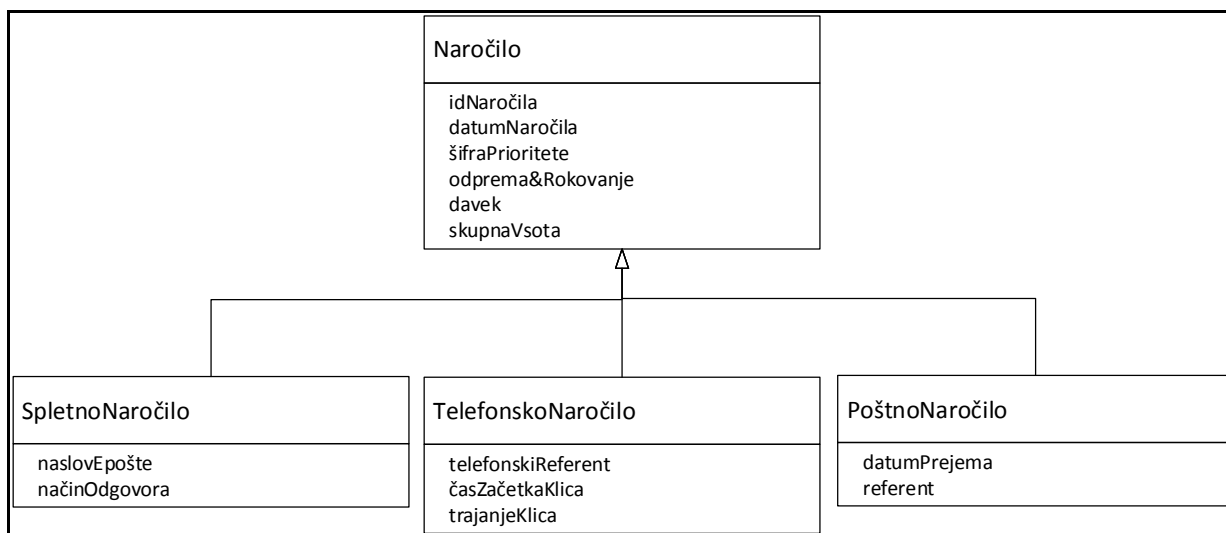
V nadaljevanju sledijo opredelitev in dodatna pojasnila glavnih sestavnih delov oziroma značilnosti objektnega modela po Jakliču (1996, str. 117–121):

- Objekt je katerakoli stvar realnega sveta, o kateri hranimo podatke in njeno obnašanje. V modelu je predstavljen z vrednostmi svojih atributov in opisom obnašanja.
- Objekte običajno kategoriziramo enako kot stvari v realnem svetu bodisi na podlagi enakih lastnosti ali podobnega obnašanja. Takšno kategorijo objekta imenujemo tip.
- Tip implementiramo z razredom. Z drugimi besedami, razred je implementacija tipa. Tudi razred je objekt, ki predstavlja tip objekta. Zato ga imenujemo metaobjekt.
- Atribut je lastnost objekta. Tudi attribute lahko obravnavamo kot objekte, kar je v skladu z objektnim pristopom. Slednji dovoljuje tudi strukturirane podatkovne tipe atributov, ki jih delimo v štiri skupine: enostavne, referenčne, zbirke in izpeljane attribute:
  - enostavni atributi so predstavljeni z vrednostmi, ki so literali (gre za vrednosti, ki so določenega podatkovnega tipa, na primer število, znak ali niz znakov),
  - referenčni atributi služijo kot povezave med objekti (en objekt je atribut drugega objekta),
  - zbirke so kompleksni atributi, ki so lahko množica, polje, seznam ali vreča (dovoljuje ponavljanje istega elementa),
  - izpeljani atributi se izračunajo iz drugih atributov, kar pomeni, da nimamo shranjenih njihovih vrednosti, ampak postopek za pridobitev te vrednosti.
- Obnašanje objektov, ki ga uvaja objektni model, je predstavljeno z operacijami. Uporabljajo se za branje in manipulacijo podatkov o objektu. Operacije, ki so kodirane v programu, imenujemo metode.
- Zahteva je tisto, kar morajo objekti poslati drugemu objektu, za dostop do njegovih podatkov.
- Ograjevanje (enkapsulacija) je lastnost objekta, ki določa, da lahko do njegovih podatkov dostopamo in manipuliramo z njimi samo prek njegovih operacij (metod).
- Objekti različnih razredov se na isto zahtevo odzovejo na različne načine, kar imenujemo mnogoličnost oziroma polimorfizem.
- Objektni podatkovni model podpira povezave tipov »ena proti ena«, »ena proti mnogo« ali »mnogo proti mnogo«.

- Hierarhija tipov pomeni, da ima določen tip objekta nadrejeni objekt, od katerega deduje attribute in/ali obnašanje. Omogočeno je tudi večkratno dedovanje, ko podrejeni tip objekta deduje od enega ali več nadrejenih tipov objekta.

Enostaven primer tipov objektov in dedovanja je prikazan na Sliki 10. Vsi trije podrejeni objekti dedujejo attribute nadrejenega objekta, poleg tega pa imajo tudi vsak svoje attribute, ki so samo njihovi.

Slika 10: Dedovanje med objekti



Vir: Satzinger et al., *Systems Analysis and Design in a Changing World*, 2009, str. 510.

Blaha in Premerlani (1998, str. 185) razlagata, da je objektni model podatkov in procesov povezan z objektno orientiranimi programskimi jeziki, objektnimi bazami podatkov in sistemi za upravljanje z objektnimi bazami podatkov. Objektna baza podatkov hrani objekte, ki jih ustvari objektno orientiran programski jezik za razliko od ostalih programskih jezikov, kjer objekti prenehajo obstajati, ko se program izvede. SUOBP pa upravlja s podatki, programsko kodo in sorodnimi strukturami, ki sestavljajo objektno bazo podatkov. Prednost SUOBP-jev je, da podpirajo objektni model podatkov in procesov ter njegove značilnosti, kot so dedovanje, atributi, metode in druge napredne lastnosti. Čeprav so se SUOBP-ji razvili iz potreb po odpravljanju pomanjkljivosti sistemov za upravljanje z relacijskimi bazami podatkov (SURBP) pri upravljanju z objektnimi bazami, pa imajo poleg svojih prednosti tudi nekaj slabosti.

Ena od slabosti, ki jo navajata Blaha in Premerlani (1998, str. 185), je nepopolna teoretska podlaga in nepopolni standardi. Če ni standardov in enotne teorije, se SUOBP-ji lahko razlikujejo in posledično pogojujejo odvisnost aplikacije.

Stanje standardizacije se je izboljšalo z oblikovanjem dveh skupin, in sicer že prej omenjena skupina OMG in skupina ODMG (angl. *Object Data Management Group*) (Jaklič, 1996, str. 140).

V skupino ODMG pa se je združilo nekaj ponudnikov rešitev SUBP z namenom standardizacije SUOBP-jev. Njihov uspeh je bil pogojen s tem, da so bili sami največji ponudniki SUOBP-jev in so tako lahko vplivali na razvoj objektnih baz podatkov in da so upoštevali standarde skupine OMG. Leta 1993 je skupina izdala prvi dokument standardov (ODMG-93), kjer so opredelili (Jaklič, 1996, str. 140–142):

- objektni model (določa osnovne pojme, kot so tip, razred, metoda, dedovanje ...),
- jezik za definiranje objektov (namenjen je definiranju tipov objektov, ki ustrezajo specifikacijam objektnega modela, istočasno pa je neodvisen od programskih jezikov),
- objektni poizvedovalni jezik (angl. *OQL – object query language*), ki omogoča tvorjenje objektov in poizvedovanje,
- povezavo z dvema objektnima programskima jezikoma (Smalltalk in C++).

Skupina ODMG je svoje delo zaključila leta 2001 in je bila nato razpuščena. Zadnja verzija njihovih standardov so ODMG 3.0. Nadaljevanje njihovega dela je prevzela skupina OMG, ki pripravlja četrto izdajo standardov (Object Data Management Group, 2013).

Danes se SUOBP vseeno ne uporablja toliko kot SURBP oziroma je to odvisno od samega namena računalniškega programa in vrste podatkov, ki jih želimo shraniti. Za računalniške programe, ki so napisani v objektnih jezikih in uporabljajo objektno modeliranje z objektnimi podatkovnimi modeli za oblikovanje baz podatkov, obstaja dodatna možnost shranjevanja podatkov. S pomočjo tehnologije **objektno relacijske preslikave** (angl. *Object relational mapping – ORM*), ki objekte, njihove attribute, relacije in obnašanje preslika oziroma pretvori v vrednosti, ki jih je mogoče shraniti v relacijsko bazo podatkov. Primer takšne tehnologije je ActiveRecord, ki se uporablja pri izdelavi spletnih aplikacij z ogrodjem Ruby on Rails. ActiveRecord poskrbi za preslikavo objektov v relacijsko bazo (običajno SQL).

## **3 PRENOVA INFORMATIZACIJE POSLOVANJA**

### **3.1 Predstavitev podjetja CeFormT d.o.o.**

Podjetje CeFormT d.o.o. je bilo ustanovljeno 10. 10. 2000. Glavna storitev podjetja je izvajanje prevajalskih storitev predvsem na področjih programske opreme in avtomobilske industrije. Podjetje ima 12 zaposlenih.

#### **3.1.1 Poslanstvo**

Poslanstvo podjetja je »omogočanje uporabe slovenskega jezika v poslovnem življenju tudi pri najobsežnejših poslovnih besedilih in informacijskih tehnologijah iz tujine.« (Poslanstvo, 2013)

Poleg tega tujim podjetjem in mednarodnim institucijam omogoča, »da hitreje in lažje posredujejo uporabnikom slovenskega jezika tudi tiste prevode, ki zaradi velikega obsega

zahtevajo sodelovanje večjega števila prevajalcev s tem, da je poleg ostalih meril prevajalske stroke zagotovljena tudi terminološka in slogovna konsistentnost prevoda« (Poslanstvo, 2013).

### **3.1.2 Vizija**

Podjetje uresničuje »vizijo uglednega in prodornega podjetja, ki bo kar najhitreje razvijalo ponudbo kakovostnih prevodov in omogočalo uporabo najzahtevnejših strokovnih tujih del v slovenskem jeziku« (Vizija, 2013).

### **3.1.3 Poslovni koncept**

CeFormT d.o.o. želi za znane stranke nuditi prevajalske storitve večjih obsegov na področju tehničnih besedil in spremne dokumentacije. V očeh stranke želi biti njen lastni prevajalski oddelek, ki samo zanjo opravlja vse dejavnosti na področju prevajanja in lokalizacije. Ciljne stranke so srednje velika in velika mednarodna podjetja.

### 3.1.4 Poslovni model

Slika 11 prikazuje model podjetja CeFormT d.o.o. z vsemi gradniki, opisanimi v poglavju 1.3.1.

Slika 11: Poslovni model podjetja CeFormT d.o.o.

<b>Ključni partnerji</b> <ul style="list-style-type: none"> <li>• podjetje za informacijsko podporo</li> <li>• ponudnik internetnih storitev</li> <li>• partnerska prevajalska podjetja</li> <li>• zunanji sodelavci prevajalci in redaktorji</li> </ul>	<b>Ključne dejavnosti</b> <ul style="list-style-type: none"> <li>• prevod besedil</li> <li>• oblikovanje besedil</li> <li>• projektni management</li> <li>• vzdrževanje baz prevodov</li> <li>• redakcija besedil</li> <li>• nenehna optimizacija in razvoj prevajalskih procesov</li> </ul>	<b>Ponudba vrednosti</b> <ul style="list-style-type: none"> <li>• kakovostni prevodi tehničnih besedil in spremne dokumentacije</li> <li>• skrb za vse strankine potrebe na področju prevajanja in lokalizacije</li> </ul>	<b>Odnosi s strankami</b> <ul style="list-style-type: none"> <li>• vloga zunanjega prevajalskega oddelka za vsako stranko z eno kontaktno točko</li> <li>• dolgoročno sodelovanje na pogodbenih osnovah</li> </ul>	<b>Stranke</b> <ul style="list-style-type: none"> <li>• slovenska in tuja mednarodna podjetja vseh velikosti, ki dolgoročno potrebujejo storitve prevajanje in lokalizacije</li> </ul>
	<b>Ključni viri</b> <ul style="list-style-type: none"> <li>• zaposleni prevajalci</li> <li>• zaposleni področni strokovnjaki</li> <li>• strojna in programska oprema za prevajanje</li> <li>• razviti procesi in znanje na področju prevajanja</li> <li>• baze prevodov</li> <li>• terminološke baze</li> </ul>		<b>Kanali</b> <ul style="list-style-type: none"> <li>• e-pošta</li> <li>• spletna stran</li> <li>• telefon</li> </ul>	
<b>Stroški</b> <ul style="list-style-type: none"> <li>• strojna in programska oprema</li> <li>• prevajalci in strokovnjaki</li> <li>• osredotočenje na ustvarjanje vrednosti</li> </ul>		<b>Prihodki</b> <ul style="list-style-type: none"> <li>• pogodbene cene za storitve prevajanja in podporne storitve</li> </ul>		

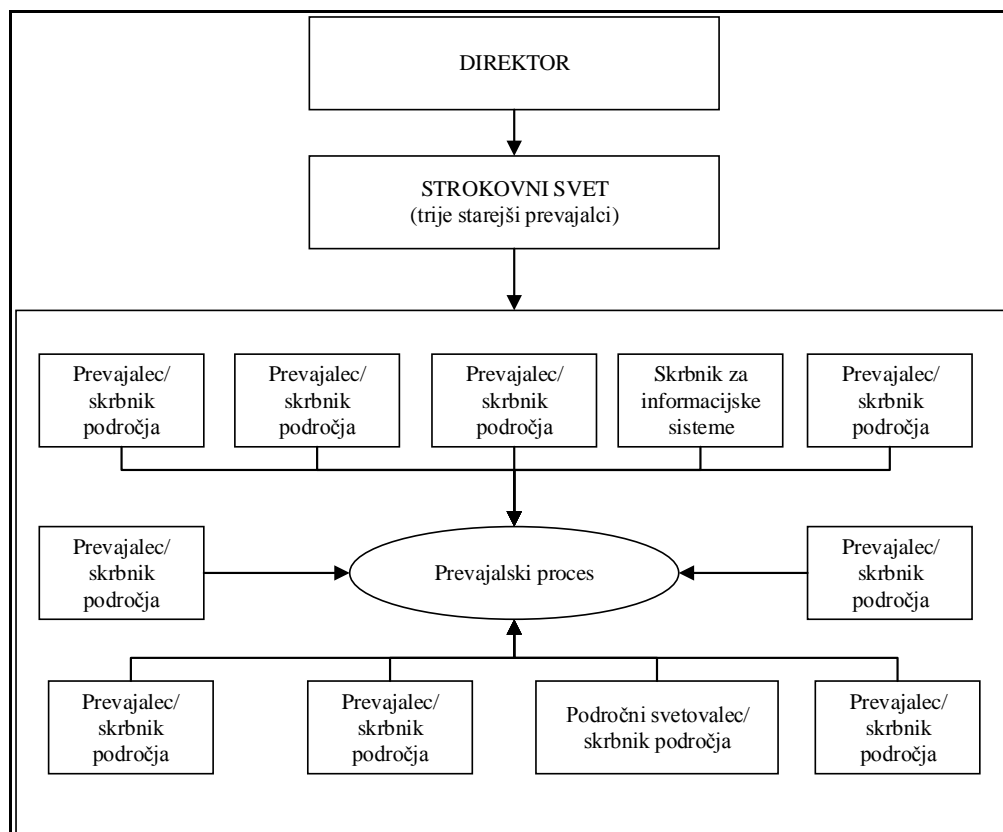
### 3.1.5 Organizacijska struktura

#### Različica izmenične ravnalne strukture v podjetju CeFormT d.o.o.

Najprej je treba opredeliti pojem »različica« izmenične ravnalne strukture, ki je prikazana na Sliki 12. Lipovec (1987, str. 174) navaja, da je v izmenični ravnalni strukturi zaposleni enkrat v

vlogi izvajalca, drugič v vlogi ravnalca. Ko zaključi določen projekt, je zopet izvajalec. V podjetju CeFormT d.o.o. se področja prevajalskega procesa ne spreminjajo in zato ravnalci<sup>2</sup> ne menjajo projektov, ampak nenehno skrbijo za eno področje – en projekt. Odstopanje se na prvi pogled zdi nepomembno, vendar igra dosti večjo vlogo, kot se zdi. Skrbnik se za svoje področje v času specializira in pridobiva avtoriteto v odnosu do sodelavcev. Dodana vrednost zaposlenega se večja, saj se prispevek k izboljšanju prevajalskega procesa v času povečuje in je vedno bolj kakovosten.

Slika 12: Različica izmenične ravnalne strukture



Vsak od prevajalcev ima dodatno vlogo skrbnika za posamezno področje prevajalskega procesa, ki ureja določen vidik le-tega (na primer: baze prevodov, projektni management itd.). Za nadzor nad prevajalskim procesom in posameznimi skrbniki je bil ustanovljen strokovni svet, ki ga sestavljajo trije starejši prevajalci. Tudi oni so v vlogi prevajalcev in skrbnikov posameznih področij prevajalskega procesa. Strokovni svet je neposredno podrejen direktorju.

Delovanje poteka na naslednji način. Vsak prevajalec/skrbnik ima torej dve vlogi. Ko kot skrbnik ugotovi potrebe po spremembi določene stvari na svojem področju, zasnuje delni projekt (celotni projekt je namreč ravnanje področja) in ga predstavi strokovnemu svetu. Poleg idejnega koncepta določi tudi vse stvari, ki jih bo potreboval za izvedbo, vključno z oceno potrebnih ur. S strokovnim svetom se nato dogovori o načinu in roku izvedbe. Ko je projekt zaključen, se po potrebi organizira usposabljanje za ostale zaposlene in uveljavi spremembe. S tem je delni

<sup>2</sup> V podjetju CeFormT d.o.o. uporabljamo izraz »skrbnik področja prevajalskega procesa. Ta izraz bom uporabljal tudi v nadaljevanju.

projekt zaključen in zopet je v vlogi prevajalca. Na ta način deluje različica izmenične ravnalne strukture v našem podjetju (CeFormT d.o.o., 2012j, str. 4).

### **Prednosti in pomanjkljivosti**

Razgibano delo zaposlenih je nedvomno najpomembnejša prednost. Monotonost dela se v celoti izloči, saj imajo zaposleni z dodatno vlogo skrbnikov zelo popestreno delo. Ker vsi delajo v izvedbeni in ravnalni funkciji ter imajo enakomerne pristojnosti, odgovornosti in oblast, je bolj prisotno vzdušje soodločanja, timskega dela in želja po skupnem razvoju in napredovanju. Prav tako se ustvari demokratično in sproščeno delovno okolje. Enakega pomena je dejstvo, da ima vsak zaposleni poleg novih zadolžitev tudi možnost, da se znotraj nove vloge izobražuje in ustrezno specializira. Ker nenehno pogloblja znanja na svojem področju in razvija izboljšave, znatno prispeva k temu, da je prevajalski proces v vedno večji meri rutinsko programiran.

Zgoraj navedene prednosti omogočajo hitrejši razvoj podjetja in prevajalskega procesa. Vsem so jasni in predvsem bližji skupni cilji podjetja, s tem pa je izpolnjen eden od pomembnejših pogojev za smotrno doseganje teh ciljev.

Pri pomanjkljivostih je treba izpostaviti čas in napor, ki ga porabi strokovni svet skupaj s posameznimi skrbniki. Za nadzor in usmerjanje takšne ravnalne strukture na prvi instanci je potrebno kar nekaj napora znotraj strokovnega sveta. Če namreč hočemo uspešno delovati in se razvijati, je treba celoten proces nadzorovano usmerjati. To pomeni koordinacijo vseh skrbnikov, da se zagotovi jasen in enoten razvoj na vseh področjih. Posledično to vodi do več sestankov s posameznimi skrbniki in s celotnim timom, vendar to ne preprečuje smotrnosti delovanja, saj s kakovostno koordinacijo lahko zmanjšamo porabo časa.

## **3.2 Opredelitev problema – dokument obsega sistema**

### **3.2.1 Opis problema**

V podjetju nudimo prevajalske storitve za projekte večjih obsegov. Vse storitve v pretežnem delu izvaja stalen prevajalski tim, ki ga vodi en projektni manager. To pomeni, da prevajalskih projektov ne dajemo v zunanje izvajanje. Glavni razlog za to je visoka standardizirana raven kakovosti prevodov, ki jo lahko zagotavlja stalen prevajalski tim. Tak tim deluje kot enovita celota z enim ciljem in ni razdeljen na podskupine. V ta namen smo tudi razvili že opisano organizacijsko strukturo in prilagodili poslovne procese.

Trenutno so poslovni procesi informatizirani s pisarniškim programom MS Excel. Avtomatizacija procesov je do neke mere zagotovljena s pomočjo VBA-makrov. S tem smo v največji možni meri izločili vse nepotrebne ročne obdelave in vnose podatkov, ki smo jih s to obliko informatizacije lahko. Vseeno pa je še vedno potrebno veliko ročnih opravil za izvajanje aktivnosti iz spodnjih problemskih področij. Posledično obstaja relativno visoka stopnja tveganja človeških napak in zamud pri poročanju. Na to je vezano tudi shranjevanje podatkov v enotno

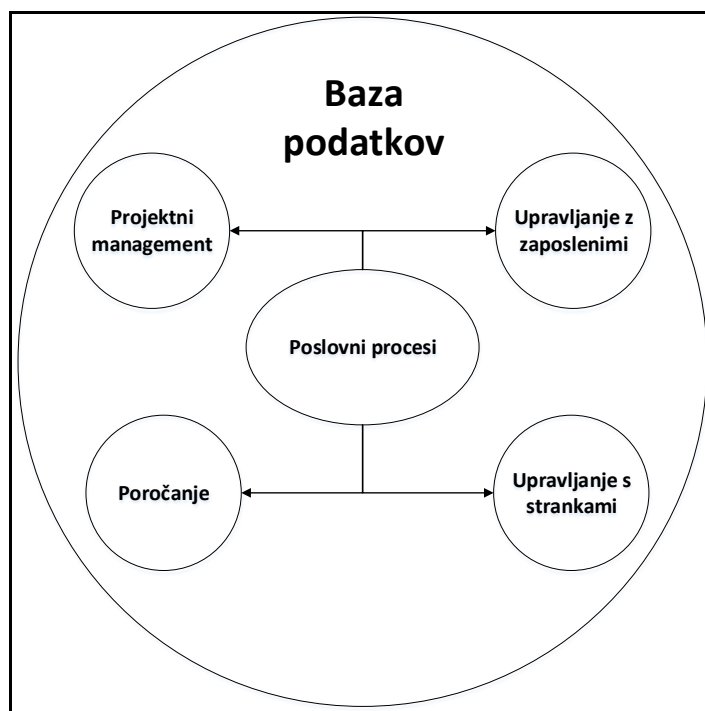


bazo, ki je predpogoj za učinkovito poročanje v realnem času in tudi za ostala poročila. Trenutno imamo enotno bazo samo za shranjevanje matičnih podatkov strank, podatki o naročilih in projektih pa so shranjeni v različnih datotekah.

Poleg makrov avtomatizacijo zagotavljata še dva portala, ki sta zasnovana v jezikih html in javascript, in do katerih se dostopa prek spletnega brskalnika. Portala sta dostopna samo znotraj mreže podjetja (intraneta). En portal je namenjen projektnemu managementu, drug pa prevajalcem. Glavne funkcionalnosti obeh portalov so: dostop do vse relevantne dokumentacije in urejanje ter dostop do vseh spletnih povezav na enem mestu. Portala nista integrirana v sistem projektnega managementa ali v računalniški prevajalski program.

Naslednji korak, ki ga je treba narediti, je prenova informatizacije poslovnih procesov. Glede na poslovne procese, lahko potrebe informatizacije poslovanja razdelim na naslednje module: upravljanje z zaposlenimi, upravljanje s strankami, poročanje in projektni management. V nadaljevanju bom za vsakega opredelil posamezna problemska področja, ki zahtevajo prenovu informatizacije poslovanja. Slika 13 prikazuje predlagane module, ki izhajajo iz poslovnih procesov in uporabljajo enotno bazo podatkov, v Tabeli 6 pa so opredeljena problemska področja za vsak modul.

*Slika 13: Moduli predlagane celovite programske rešitve*



Analiza potreb za te module bo sledila poslovnim procesom in organizaciji našega tipa agencije, zato se bodo ti moduli razlikovali od standardnih ali pa panožno specifičnih modulov, saj naj bi vključevali le stvari, ki jih potrebujemo.

Tabela 6: Moduli in problemska področja

Modul	Problemsko področje
Projektni management	<ul style="list-style-type: none"> <li>• dinamično načrtovanje projektov in razporejanje prevajalcev</li> <li>• dodeljevanje verige opravil prevajalskega projekta</li> <li>• integracija s prevajalskimi računalniškimi programi</li> <li>• evidentiranje opravljenega dela</li> <li>• priprava podatkov za računovodstvo</li> </ul>
Poročanje	<ul style="list-style-type: none"> <li>• napredek projektov</li> <li>• razporejenost in razpoložljivost prevajalcev</li> <li>• uspešnost realizacije projekta po prevajalcih</li> <li>• finančna uspešnost projekta</li> </ul>
Upravljanje s človeškimi viri	<ul style="list-style-type: none"> <li>• vzdrževanje matičnih podatkov zaposlenih</li> <li>• spremljanje izobraževanj in usposabljanj zaposlenih</li> <li>• evidentiranje in spremljanje izrabe delovnega časa zaposlenih</li> <li>• vzdrževanje matičnih podatkov zunanjih izvajalcev</li> </ul>
Upravljanje s strankami	<ul style="list-style-type: none"> <li>• vzdrževanje in posodabljanje matičnih podatkov strank</li> <li>• spremljanje dejavnosti strank</li> </ul>

### Projektni management

Pri **dinamičnem načrtovanju projektov in razporejanju prevajalcev** gre za umestitev novega projekta med obstoječe projekte, pri čemer moramo upoštevati potrebe projekta na eni in razpoložljive vire na drugi strani. Potrebe projekta so opredeljene kot: število ur prevajanja, jezikovna kombinacija, področje prevajanja (pravo, ekonomija itd.), besedilna vrsta (pogodba, aneks itd.). Vire pa predstavljajo prevajalci, ki so usposobljeni za projekte s takimi potrebami. To predstavlja večji problem, ker vse prevajalske projekte izvaja stalen prevajalski tim, kar pomeni, da so viri vedno omejeni. Na drugi strani pa so potrebe dela (vsota potreb vseh odprtih projektov) vnaprej neznane in v stalnem nihanju, zato je treba nenehno uravnavati potrebe dela in razpoložljive vire. Če bi prevajalske projekte oddajali v zunanje izvajanje in bi v ta namen imeli veliko bazo izvajalcev, se s tem problemom ne bi soočali v takšni razsežnosti. V našem primeru mora projektni manager vsak nov projekt umestiti med obstoječe prevajalce in zagotoviti, da bo izveden do dogovorjenega roka. V idealnem primeru začetek projekta (projekt B) lahko določi za enim od obstoječih projektov. Pogosto pa se zgodi, da je treba projekt zaključiti vzporedno z ostalimi ali pa celo pred ostalimi projekti. V tem primeru mora projektni manager razporediti prevajalce z drugega projekta (projekt A) na projekt B. Seveda pa je potem treba vse odvzete ure s projekta A nadomestiti in zagotoviti dokončanje do roka. To dinamično načrtovanje je zelo zamudno, saj ga je treba izvajati ročno na več mestih v sistemu.

**Dodeljevanje verige opravil prevajalskega projekta** pomeni, da se vsa opravila prevajalskega projekta in njihovi izvajalci določijo na začetku projekta. Ob sprožitvi dodelitve dobi prvi izvajalec obvestilo o dodeljenem opravilu. Ko zaključi dodeljeno opravilo, obvestilo o dodelitvi dobi naslednji izvajalec in tako naprej do zadnjega izvajalca. To funkcionalnost že vsebujejo

nekateri prevajalski računalniški programi, vendar bi bilo bolje imeti to funkcionalnost implementirano neodvisno in bi se lahko uporabljala tudi s programi, ki te funkcionalnosti še nimajo. Trenutno v našem podjetju nimamo takšne neodvisne funkcionalnosti.

Trenutno uporabljamo več različnih prevajalskih računalniških programov, kar lahko pričakujemo tudi v bodoče. Zato bi bila nujno potrebna **integracija s temi prevajalskimi računalniškimi programi** na način, da se priprava projekta in dodelitev verige opravil izvede v našem informacijskem sistemu enkrat in je ni treba ponavljati še v specifičnem računalniškem programu za prevajanje. Druga prednost integracije pa bi bila avtomatično pridobivanje podatkov za vsa poročila.

**Evidentiranje opravljenega dela** na projektih je izrednega pomena, saj je to osnova za različna poročila in obračun plač. Trenutno opravljeno delo evidentiramo v orodju v okviru aplikacije MS Excel, vendar ima le-to številne pomanjkljivosti. Ob vnosu podatkov obstaja razmeroma visoka verjetnost človeške napake. Poleg tega je priprava različnih poročil zamudna in zahteva velik napor pri kontroliranju podatkov.

Tudi **priprava podatkov za računovodstvo** je ključnega pomena in mora biti izvedena takoj po zaključku projekta oziroma takoj po strankini potrditvi prevzema dokumenta. Projektni manager zapre delovni nalog in ga pošlje v računovodstvo v obračun. Ker ni sistemskih kontrol, obstaja tveganje, da obračun zamuja ali pa sploh ne pride do računovodstva.

## **Poročanje**

**Poročanje (tudi v realnem času)** je dandanes izrednega pomena v vseh panogah in podjetjih, saj omogoča podlago za sprejemanje poslovnih odločitev. V našem primeru potrebujemo poročanje v realnem času predvsem za namene dinamičnega načrtovanja projektov in razporejanja prevajalcev ter napredku odprtih projektov. Informacije o zasedenosti prevajalcev in napredku odprtih projektov so ključnega pomena za projektne managerja, ko se odloča o izvedbi in pripravlja načrt za nov projekt. Trenutno poročanje ni avtomatizirano in je vezano na pridobivanje informacij iz različnih virov in njihovo naknadno obdelavo, ki vključuje tudi ročne korake, ki jih ni mogoče avtomatizirati z VBA-jem. Poleg tega potrebujemo tudi avtomatizacijo priprave mesečnih poročil o realizaciji načrta in finančni uspešnosti po projektih in prevajalcih.

## **Upravljanje s človeškimi viri**

Pri upravljanju s človeškimi viri je največja pomanjkljivost decentralizirano in neenotno **vzdrževanje matičnih podatkov zaposlenih, spremljanje in evidentiranje izobraževanj in usposabljanj zaposlenih, evidentiranje in spremljanje izrabe delovnega časa zaposlenih in vzdrževanje matičnih podatkov zunanjih izvajalcev**. Podatki o zaposlenih in zunanjih izvajalcih so vodeni ločeno in na različne načine. Posledično priprava določenih poročil zahteva veliko ročnega dela in napora, kar je treba avtomatizirati.

## Upravljanje s strankami

Čeprav sodelujemo samo z znanimi strankami na dolgoročno obdobje, moramo tudi za to področje prenoviti informatizacijo procesov. Podatke strank moramo konsolidirati na enem mestu in omogočiti njihovo **enostavno vzdrževanje in posodabljanje**. Poleg tega bi bilo smiselno pri prenovi informatizacije v ta modul vključiti tudi spremljanje dejavnosti strank, saj delujemo kot njihov zunanji prevajalski oddelek. S tem bomo lažje, hitreje in boljše odreagirali na njihove potrebe ali spremembe v njihovem poslovanju.

### 3.2.2 Pričakovane poslovne koristi

S prenovi informatizacije poslovanja zgoraj navedenih modulov in problemskih področij pričakujem naslednje poslovne koristi:

- zmanjšanje napak ob vnosu in obdelavi podatkov,
- hitrejša in točnejša poročila o napredku in finančna poročila,
- hitrejša in bolj avtomatizirano dinamično načrtovanje prevajalskih projektov,
- ohranitev enake organizacijske strukture (en stalen prevajalski tim z enim projektnim managerjem) tudi ob znatnem povečanju obsega dela,
- zanesljivejša in bolj uporabna hramba podatkov,
- znaten prihranek časa zaradi manjšega števila ročnih opravil,
- prihranek časa zaradi integracije s prevajalskimi računalniškimi programi,
- učinkovitejše prilagajanje potrebam strank in hitrejša izvajanje storitev.

### 3.2.3 Pričakovane zmožnosti novega sistema

Za uresničitev pričakovanih poslovnih koristi mora nov informacijski sistem:

- uporabljati enotno bazo podatkov,
- biti integriran s prevajalskimi računalniškimi programi,
- omogočati poročanje v realnem času,
- omogočati ustvarjanje mesečnih poročil o realizaciji načrta in finančni uspešnosti po projektih in prevajalcih,
- omogočati dinamično načrtovanje v obliki »peskovnika«, kjer se lahko primerja različne različice načrtov in izvedb projektov,
- omogočati evidentiranje opravljenega dela po prevajalcih in projektih,
- omogočati avtomatizirano pripravo obračunskih podatkov za računovodstvo,
- omogočati učinkovito upravljanje s strankami.

### **3.3 Zbiranje informacij**

Med zbiranjem informacij bom pregledal opise postopkov, poročila in obrazce, ki se trenutno uporabljajo v podjetju. Poleg tega bom tudi opazoval izvajanje poslovnih procesov, ki jih bom predstavil z opisnim modelom in diagrami aktivnosti. Informacij bom črpal tudi iz več kot desetletnih izkušenj, ki sem jih pridobil v prevajalskem procesu v različnih vlogah (prevajalec, terminolog, projektni manager, skrbnik področja prevajalskega procesa). Zadnji korak zbiranja informacij bo preučitev nekaterih obstoječih rešitev na trgu, ki omogočajo informatizacijo procesov prevajalskih agencij.

Z logičnimi modeli v obliki diagramov aktivnosti bom predstavil poslovni proces na najvišji ravni v obliki pregleda, nato pa podrobneje vse njegove sestavne dele, kot se trenutno izvajajo.

#### **3.3.1 Poslovni procesi v podjetju CeFormT d.o.o.**

Glavna storitev v podjetju CeFormT d.o.o. je prevajanje tehničnih in strokovnih besedil večjih obsegov. Prevod izdeluje stalen prevajalski tim, ki je sestavljen iz redno zaposlenih prevajalcev. Pri tem je najbolj pomembno, da se prevodu ne pozna, da ga je soustvarjalo večje število prevajalcev. Za to je bistvenega pomena dobro definiran prevajalski proces in komunikacija med člani tima.

Prevajalski proces je razdeljen na enajst področij, ki zagotavljajo smotrno izvajanje prevajalskih storitev in visoko kakovost prevoda. Ta področja so: projektni management, področja prevajanja, slogovni vodnik, izobraževanje in izpopolnjevanje, zagotavljanje kakovosti, terminologija in terminološke baze, baze prevodov, področno svetovanje, organizacija informacijskih sistemov, obdelava dokumentov ter uporabniški vidik prevajalskih računalniških programov. Poleg prevajalcev sta zaposlena še področni svetovalec, katerega naloga je, da vsebinsko svetuje in pomaga prevajalcem na tehnično zapletenih področjih prevajanja, in organizator informacijskih sistemov.

Vsak prevajalski projekt je izveden v skladu s standardi prevajalske panoge in internimi pravili, specifičnimi za vsako stranko.

#### **Projektni management**

Področje projektnega managementa ureja celoten življenjski cikel prevajalskega projekta v podjetju CeFormT d.o.o. od prevzema naročila do obračuna storitev. Določa posebnosti prevzema naročil, oddaje prevajalskih projektov stranki in obračunskega sistema za posamezne stranke (CeFormT d.o.o., 2012g, str. 6).

Učinkovit projektni management z dinamičnim načrtovanjem projektov je ključnega pomena za brezhibno izvedbo prevajalskih projektov in izpolnitev strankinih zahtev glede posameznih naročil.

## **Področja prevajanja**

Tu so določena pravila in postopki za vzdrževanje baze s področji prevajanja in besedilnimi vrstami v podjetju CeFormT d.o.o. Poleg tega opredeljuje pravila za določitev nosilcev posameznih področij, preference prevajalcev glede prevajanja besedil s posameznih področij ter njihove prevajalske izkušnje na posameznih področjih (CeFormT d.o.o., 2012d, str. 3).

Bazo s področji prevajanja vzdržuje njen skrbnik, in sicer tako, da v primeru novega področja prevajanja določi nosilca tega področja in od prevajalcev pridobi informacije o njihovih preferencah in izkušnjah glede tega področja. Nosilci so nato odgovorni za vsa terminološka vprašanja tega področja.

Projektni manager bazo uporablja, ko dodeljuje prevajalske projekte v prevod. Pri tem upošteva preference in izkušnje prevajalcev na določenem področju.

## **Slogovni vodnik**

Slogovni vodnik za posamezni projekt (lahko je enak za vse projekte ene stranke) vsebuje pravila za uporabo jezika pri prevajanju določenega projekta, ki zagotavljajo konsistentnost prevodov.

Slogovne vodnike delimo na interne (ki smo jih oblikovali sami) in strankine. Na enem projektu se lahko uporablja več slogovnih vodnikov, pri čemer je na začetku projekta določena njihova hierarhična uporaba.

## **Izobraževanje in izpopolnjevanje**

Področje izobraževanja in izpopolnjevanja določa interne postopke, s katerimi v podjetju CeFormT d.o.o. zagotavljamo visoko usposobljenost prevajalcev in posledično visoko kakovost prevodov. To vključuje: vodenje evidenc izobraževanja, preverjanje znanja in napredka prevajalcev, ugotavljanje potreb po dodatnem usposabljanju, dodelavo in stalno posodabljanje projektnih in slogovnih priročnikov ter navodila za usposabljanje novih prevajalcev (CeFormT d.o.o., 2012c, str. 3).

## **Zagotavljanje kakovosti**

Področje zagotavljanja kakovosti določa pravila za izvajanje pregledov, izpolnjevanje in shranjevanje obrazcev za preglede ter postopek vzdrževanja in uporabe baze teh obrazcev. Podrobno ureja tudi kategorizacijo napak, ki se spremljajo po projektih in prevajalcih, kar omogoča učinkovit nadzor nad kakovostjo prevodov (CeFormT d.o.o., 2012k, str. 4-16). Slednje služi kot osnova za nenehno usposabljanje, izobraževanje in izpopolnjevanje prevajalcev.

## **Terminologija in terminološke baze**

Področje terminologija in terminološke baze ureja in določa postopke za iskanje ustreznih prevodov terminov, slogovna pravila za oblikovanje terminov in model klasificiranja terminov. Prav tako določa pravila za ustvarjanje in poimenovanje terminoloških baz. Glavna namena tega področja sta zagotoviti terminološko ustreznost in usklajenost v prevodih ter enotno obliko poimenovanja terminoloških baz in določitev kriterijev za ustvarjanje terminoloških baz (CeFormT d.o.o., 2012h, str. 4).

## **Baze prevodov**

Področje baze prevodov ureja in določa pravila za ustvarjanje, poimenovanje in vzdrževanje baz prevodov. Namen teh pravil je, da zagotovijo enotno obliko poimenovanja baz prevodov ter določijo kriterije, kdaj se ustvari nova baza prevodov (CeFormT d.o.o., 2012a, str. 3).

Baze prevodov so v prevajalskem procesu ključnega pomena, saj zagotavljajo konsistentne prevode za določena področja in besedilne vrste, s tem pa posledično zagotavljajo tudi visoko kakovost prevodov.

## **Področno svetovanje**

Področje za področno svetovanje ureja metodologijo in naloge področnega svetovalca. Področni svetovalec je strokovnjak za določeno področje prevajanja in igra vlogo končnega uporabnika ter specialista za terminološka vprašanja. Njegove osnovne naloge so pomoč prevajalcem pri razumevanju tehničnih besedil, priprava in izvedba strokovnih usposabljanj za prevajalce, pomoč pri prevodu terminov in drugo (CeFormT d.o.o., 2012e, str. 3, CeFormT d.o.o., 2012j, str. 9-10).

## **Organizacija informacijskih sistemov**

Področje organizacije informacijskih sistemov nudi informacijsko podporo prevajalskemu procesu. Določa pravila za upravljanje in vzdrževanje računalniških programov ter postopke za odpravljanje težav (CeFormT d.o.o., 2012b, str. 3-12).

## **Obdelava dokumentov**

Področje obdelave dokumentov določa pravila in navodila za začetno obdelavo dokumentov, da ustrezajo normativom in zahtevam prevajalskih računalniških programov, in končno obdelavo dokumentov, da ustrezajo zahtevam stranke (CeFormT d.o.o., 2012č, str. 3).

## **Uporabniški vidik prevajalskih računalniških programov**

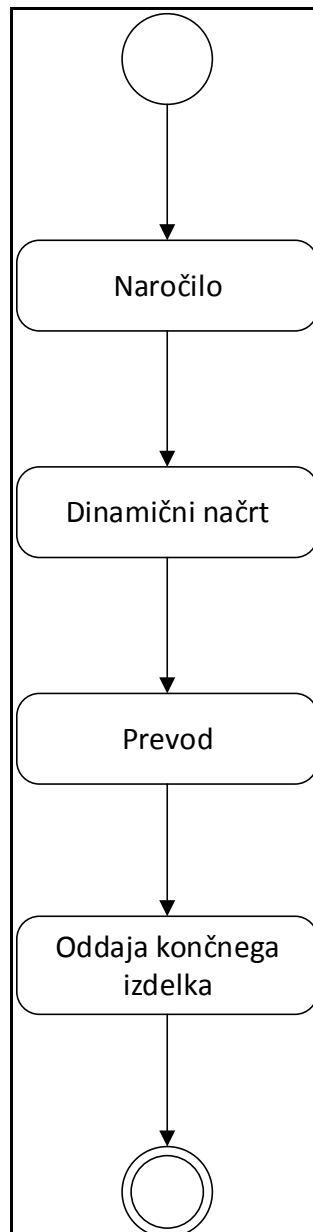
To področje določa metodologijo, po kateri skrbnik pripravlja navodila za računalniške programe, ki prevajalcem olajšajo rabo teh programov (CeFormT d.o.o., 2012i, str. 4). Enotna

metodologija zagotavlja, da vsa navodila služijo enakemu namenu in cilju ter so enako strukturirana.

### Poslovni proces – pregled

Poslovni proces je sestavljen iz strankinega naročila in obdelave le-tega, iz dinamičnega načrta izvedbe naročila, prevoda ter oddaje končnega izdelka stranki kot je vidno na Sliki 14. V nadaljevanju bom podrobneje predstavil vsakega od teh sestavnih delov.

*Slika 14: Poslovni proces – pregled*





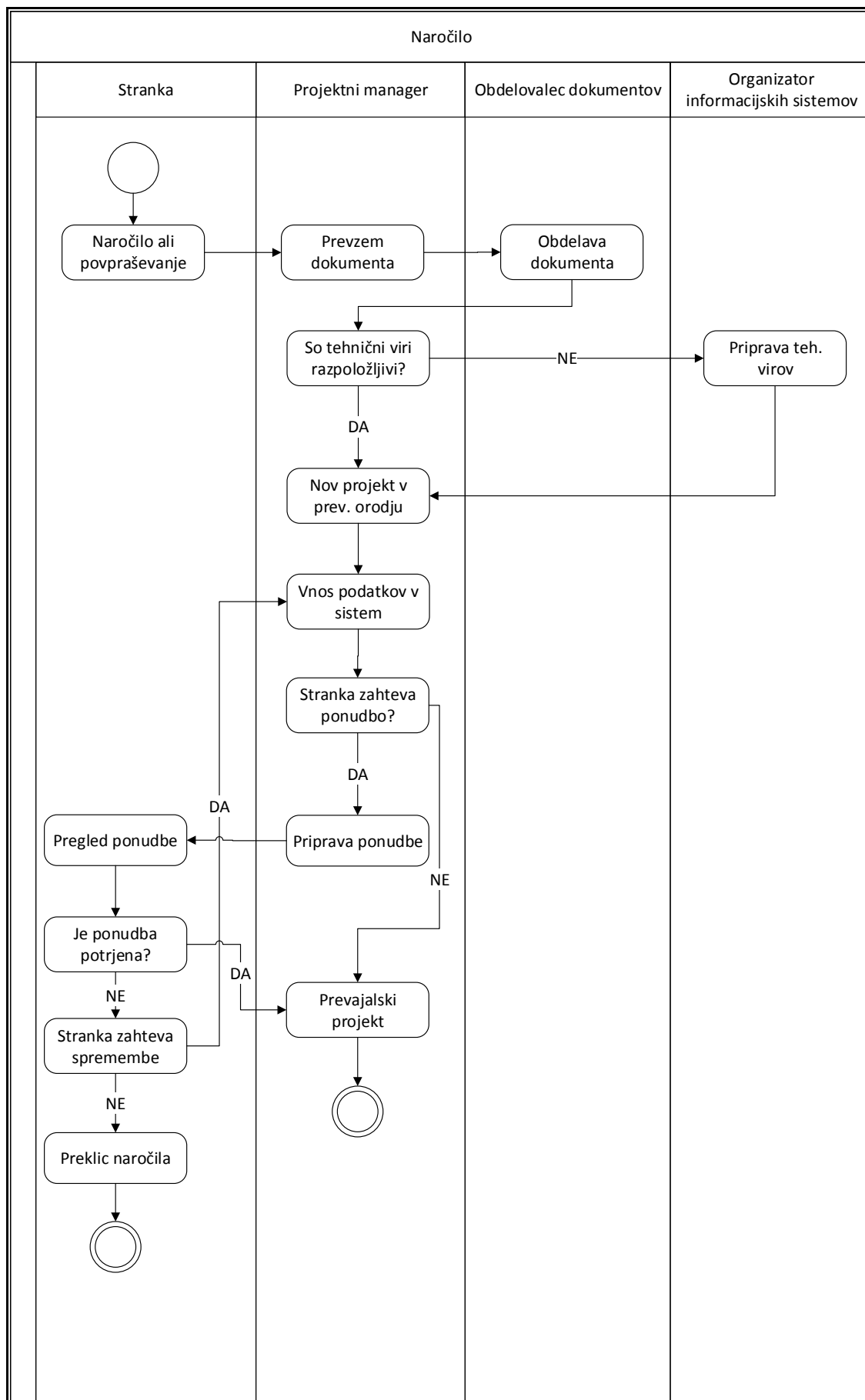
## Poslovni proces – naročilo

Poslovni proces se začne s strankinim naročilom za prevod dokumenta ali s strankinim povpraševanjem in zahtevo za ponudbo. Projektni manager shrani prejeti dokument v ustrezno mapo stranke na strežniku (če gre za fizično datoteko), kamor se kronološko shranjujejo prejeti dokumenti. Nato evidentira naročilo ali povpraševanje v evidenci prejetih naročil (z vsemi potrebnimi atributi: datum prevzema, naročene storitve, rok oddaje idr.). Vsako naročilo ali povpraševanje dobi svoj delovni nalog z enoznačno zaporedno identifikacijsko številko. Z delovnim nalogom se spremlja celotni življenjski cikel naročila in vsebuje naslednje sklope: specifikacije naročila, načrt izvedbe naročila, poročilo o izvedbi in obračun storitev. Obdelovalec dokumentov tehnično obdela dokument, da ustreza strankinim zahtevam in zahtevam računalniškega prevajalskega orodja. Projektni manager preveri razpoložljivost tehničnih virov, glede na strankine zahteve, in značilnosti projekta: baza prevodov in terminološka baza. Če ustreznih virov ni, projektni manager pri organizatorju informacijskih sistemov zahteva vzpostavitev baz.

Ko je besedilo pripravljeno za analizo s prevajalskim računalniškim programom (izpolnjuje njegove tehnične zahteve o obliki in formatu besedila), projektni manager v računalniškem prevajalskem orodju ustvari nov projekt in izvede analizo besedila za dokument. Analiza besedila pove, kakšno je ujemanje posameznih segmentov (stavkov ali odstavkov) z bazo prevodov. Segmenti so razvrščeni v različne kategorije po stopnji ujemanja s segmenti v bazi prevodov od 0 % do 100 %. Takšna analiza količine in strukture besedila za prevod služi za izdajo ponudbe in načrtovanje izvedbe naročila.

Če je stranka zahtevala ponudbo, jo projektni manager izda po izvedeni analizi besedila.. Če ponudba ni potrjena in stranka zahteva spremembo ponudbe, projektni manager pripravi drugo ponudbo. Lahko pa stranka ne zahteva druge ponudbe in prekliče naročilo. Če je ponudba potrjena, projektni manager nadaljuje s prevajalskim projektom, tako da mu v računalniškem prevajalskem orodju doda navodila in referenčna besedila. Slika 15 prikazuje naročilo prevoda.

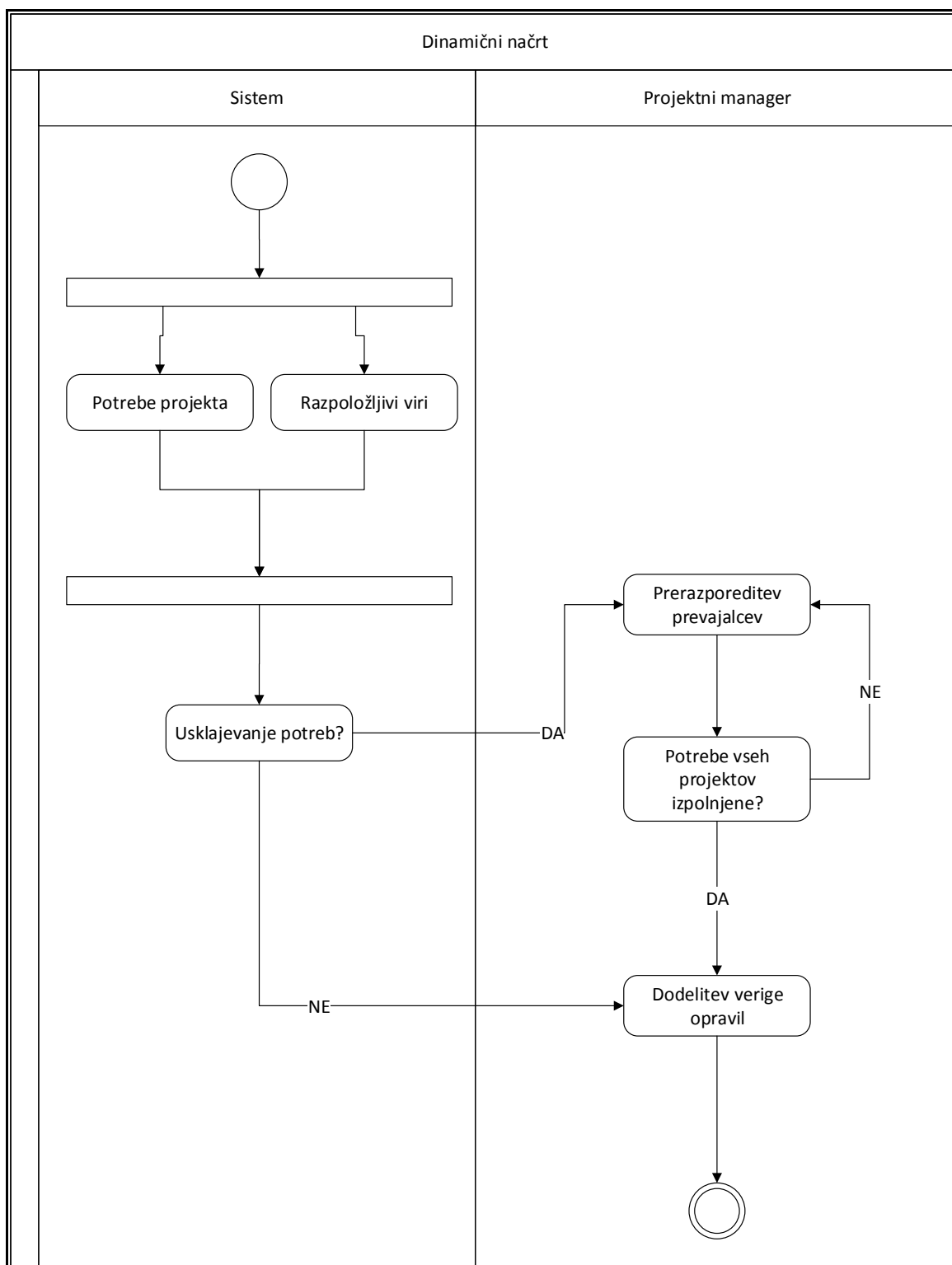
Slika 15: Naročilo



## **Poslovni proces – dinamični načrt**

Projektni manager od sistema pridobi podatke o potrebah projekta in razpoložljivih virih za izvedbo projekta. Prvo pravilo izvedbe je, da projekt izvaja en prevajalec, če ga lahko izvede do roka. Če projekta ne more izvesti samo en prevajalec ali pa je potrebno usklajevanje potreb, projektni manager izvede potrebne ukrepe, da izpolni potrebe projekta. To lahko pomeni razdelitev dela med več prevajalcev, prerazporeditev prevajalcev z drugih projektov, nadurno delo ipd. Po vsaki uskladitvi preveri, da so izpolnjene in usklajene vse potrebe vseh projektov. Dokler temu ni tako, ponavlja ta korak. Nato v delovni nalog vnese končni načrt izvedbe projekta in dodeli prevajalce, pregledovalce ter po potrebi druge člane prevajalskega tima (redaktorji, oblikovalci besedil idr.) na prevajalski projekt. Enake dodelitve izvede v računalniškem prevajalskem orodju. Potek dinamičnega načrtovanja je prikazan na Sliki 16.

Slika 16: Dinamični načrt



## **Poslovni proces – prevod**

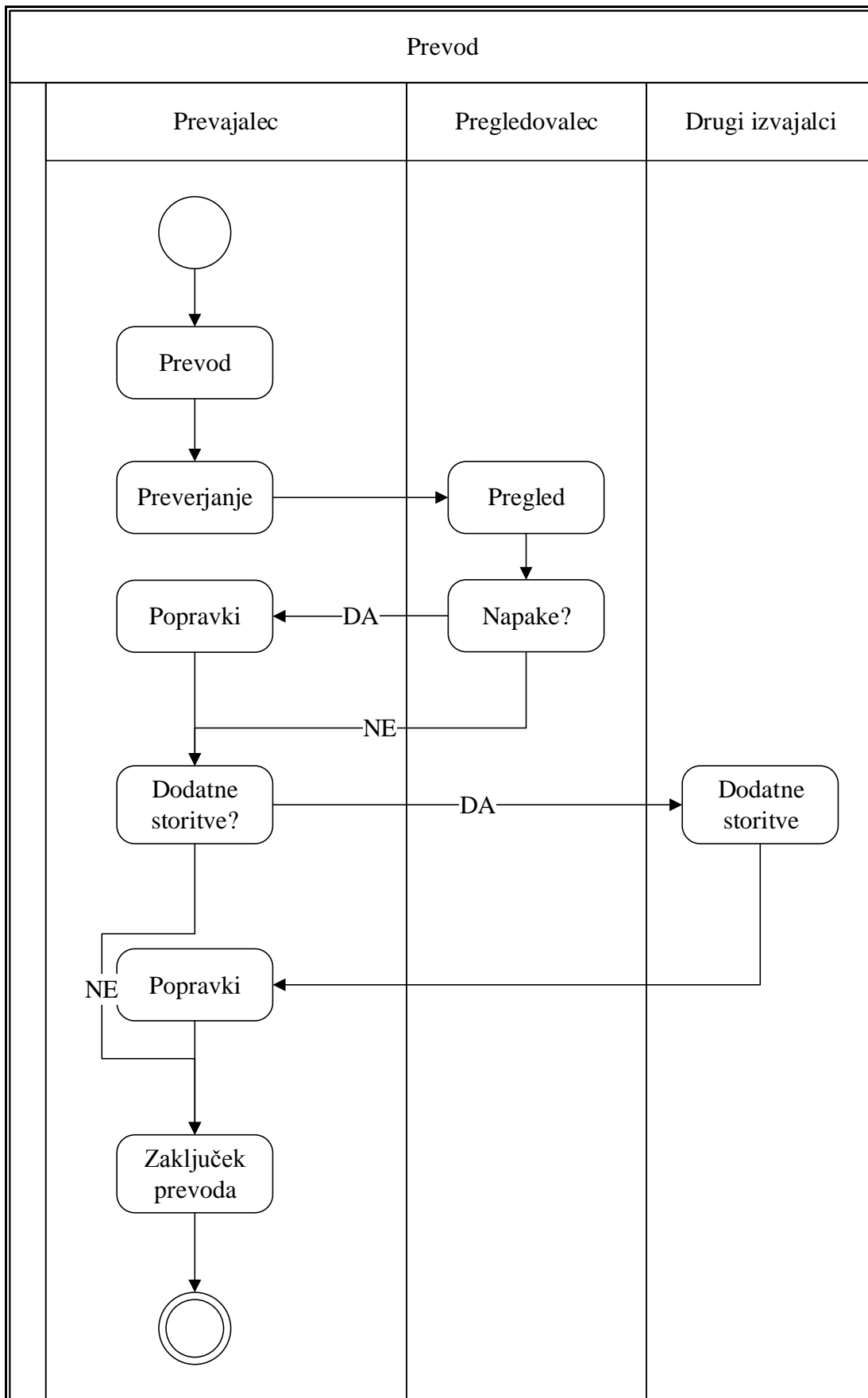
Proces prevoda, ki ga prikazuje Slika 17, se začne, ko prevajalec v računalniškem prevajalskem orodju sprejme opravilo, ki mu ga dodeli projektni manager, in začne s prevodom. Najprej prebere besedilo za prevod, pregleda referenčna besedila in projektna navodila. Nato začne s prevodom. Tu gre za prenos pomena iz izvirnega v ciljni jezik v skladu z jezikovnimi pravili ciljnega jezika, in sicer tako, da izpolnjuje zahteve iz dodelitve projekta. Pri prevodu upošteva ustrezno terminologijo in zagotavlja, da je skladna z določenim področjem in strankino terminologijo. Pri prevodu upošteva tudi slog, ki je določen v internem ali strankinem slogovnem vodniku. Eno od glavnih vodil, ki mu mora slediti, je ciljno občinstvo in namen prevoda. Ko konča prevod, prevajalec preveri svoj prevod in zaključi opravilo, da se veriga opravil pomakne do pregledovalca.

Pregledovalec, ki ni ista oseba kot prevajalec, sprejme opravilo in s primerjavo izvirnega in ciljnega besedila preveri, ali prevod ustreza namenu. Pri tem upošteva projektna navodila, referenčna besedila, ustrezno terminologijo in slog. Pregledovalec ugotovljene napake zabeleži v obrazec za pregled, ki služi za oceno kakovosti prevoda. Ko prevajalec od pregledovalca dobi besedilo nazaj, vstavi ustrezne (tiste, s katerimi se strinja) popravke.

Če so naročene še kakšne dodatne storitve (redakcija, korektura, priprava za objavo), jih opravijo ustrezni izvajalci (redaktor, korektor, obdelovalec dokumentov).

Ko redaktor ali korektor vrneta besedilo prevajalcu, le-ta vstavi ustrezne popravke (tiste, s katerimi se strinja).

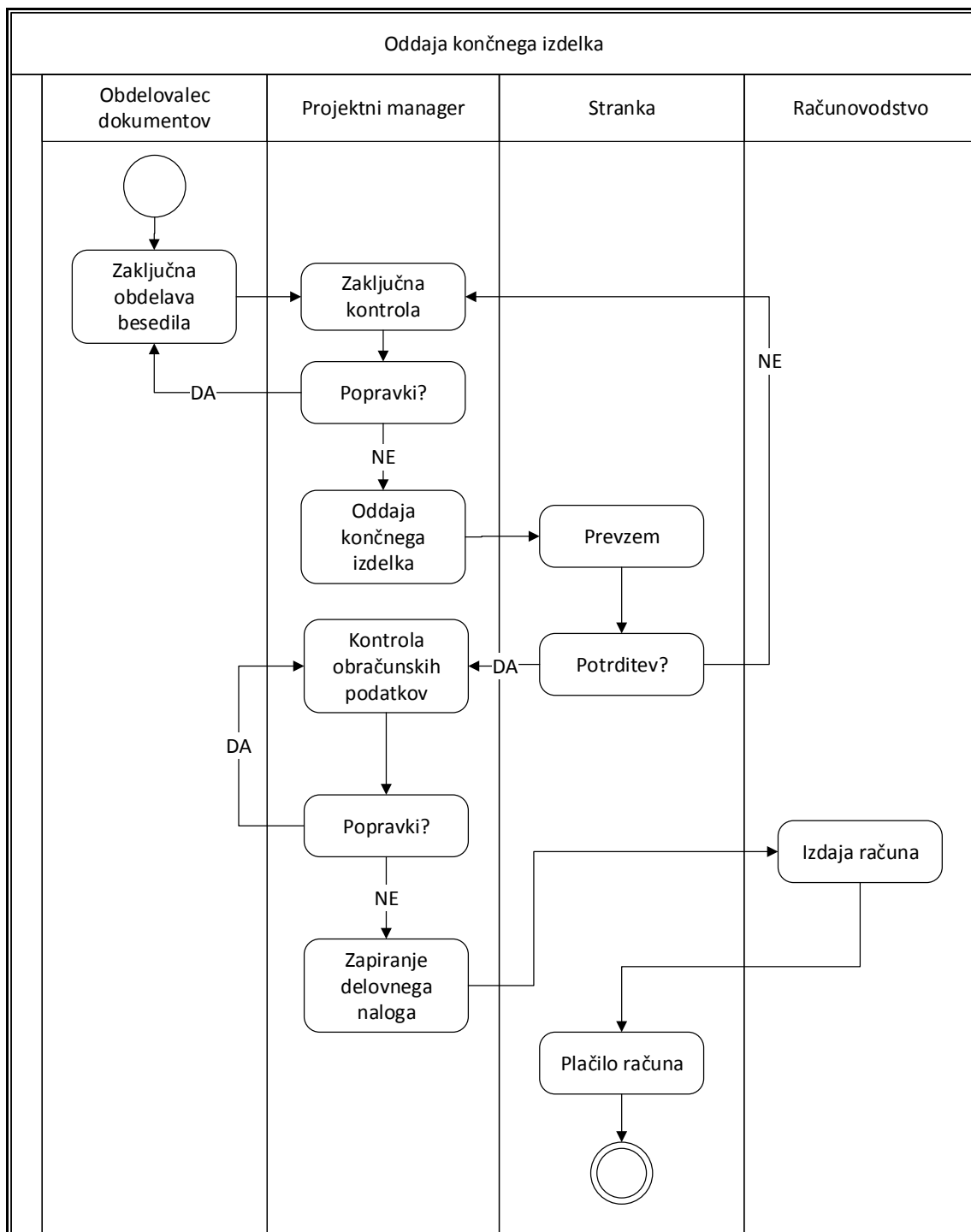
Slika 17: Prevod



## **Poslovni proces – oddaja končnega izdelka**

Slika 18 prikazuje proces oddaje končnega izdelka stranki, ki se začne, ko obdelovalec dokumentov dobi dodeljeno opravilo za zaključno obdelavo dokumenta. Dokument pripravi za oddajo, tako da ustreza strankinim tehničnim in oblikovnim zahtevam. Projektni manager opravi zaključno kontrolo pred oddajo stranki. Če so potrebni popravki, dokument vrne obdelovalcu dokumentov, drugače pa odda projekt stranki po dogovorjenem komunikacijskem kanalu. Stranka prevzame dokument in ga pregleda. Če je vse v redu, potrdi prevzem, drugače pa ga vrne projektnemu managerju v ponovni pregled. Po strankini potrditvi projektni manager preveri obračunske podatke. Če ni potrebnih popravkov, zapre delovni nalog in obvesti računovodstvo, naj izda račun. Opravljene storitve se obračunajo po ponudbi, če je bila izdana, ali po pogodbenih cenah. Proces se zaključi s strankinim plačilom računa.

Slika 18: Oddaja končnega izdelka



### 3.3.2 Preučitev obstoječih rešitev ponudnikov programske opreme

Med zbiranje informacij sem vključil tudi preučitev obstoječih rešitev vodilnih ponudnikov programske opreme za prevajalske agencije. Informacije sem zbiral na njihovih spletnih straneh ter na njihovih multimedijskih kanalih na spletnem portalu YouTube, kjer so objavljeni videoposnetki posameznih funkcionalnosti teh rešitev. Poleg tega sem se udeležil tudi nekaterih spletnih seminarjev, kjer so bile predstavljene rešitve.



Vse rešitve so namenjene managementu prevajalskih projektov in posledično vključujejo tudi module, ki jih predlagam: projektni management, poročanje, upravljanje s človeškimi viri in upravljanje s strankami. Avtomatizacija poslovnih procesov, ki jo ponujajo, v veliki meri temelji na dejstvu, da agencija prevajalske projekte v večji meri ali pa kar v celoti daje v zunanje izvajanje. To pomeni, da ima agencija veliko bazo zunanjih sodelavcev in posledično nima težav pri pridobivanju ustreznih izvajalcev za prevajalske projekte. Avtomatizacija se v najbolj splošnem zreducira na primerjavo potreb projekta z lastnostmi izvajalcev v bazi izvajalcev.

Na drugi strani, pa sem ugotovil nekaj neskladij med našimi informacijskimi potrebami in funkcionalnostmi, ki jih te rešitve ponujajo. Nekatero rešitve sicer podpirajo integracijo s prevajalskimi računalniškimi programi, vendar je ta omejena. En primer take omejitve je avtomatizacija verige opravil prevajalskega projekta. Druge rešitve pa integracijo s prevajalskimi računalniškimi programi ponuja le v obliki uvoza analize besedila prevajalskega računalniškega programa v njihov sistem, ki se nato uporablja za različne obračune in določitev obsega dela. Nekatero pa sploh ne podpirajo integracije s prevajalskimi računalniškimi programi, ampak nudijo zgolj funkcije projektnega managementa, kjer pa je tudi manjkala avtomatizirana veriga opravil: ta je določena enkrat, na začetku, za nadaljnje obveščanje in komunikacijo pa poskrbi sistem.

Prav tako sem ugotovil neskladje na področju dinamičnega načrtovanja prevajalskih projektov (predvsem v zvezi s popravki načrtov prevajalskih projektov in s tem povezanimi potrebnimi predodelitvami opravil članom prevajalskih projektov) in razpoložljivimi vrstami poročil, ki so osredotočena bolj na finančne vidike prevajalskih projektov kot pa na analize uspešnosti realizacije posameznih izvajalcev in projektov.

Močne prednosti teh rešitev so avtomatizacija, enotna baza podatkov, poročanje ter upravljanje s strankami in s človeškimi viri, ki pa se seveda razlikujejo od rešitve do rešitve.

Zgoraj ugotovljena neskladja še niso osnova za odločitev izbire ene od rešitev. Lahko pa se na tej podlagi odloči, katero ali katere rešitve se pogleda bolj podrobno v primerjavi z analizo informacijskih potreb, še posebej glede na hiter, konkurenčen in strm trend razvoja na tem področju v zadnjem času. Obseg take primerjave pa je prevelik za vključitev v magistrsko nalogo in tudi ni prvotni namen naloge. Poleg tega taka primerjava zahteva bolj tesno in daljše sodelovanje s ponudnikom take rešitve, saj je zaradi ugotovljenih neskladij zelo verjetno, da bi jo bilo treba do neke mere prilagajati.

Zelo obetavna je bila novica, objavljena 29. oktobra 2013, kjer lahko beremo o podobni stopnji integracije, kot jo predlagam sam, in sicer med rešitvijo XTRF, ki pokriva projektni management, in rešitvijo MemSource Cloud (Memsources, 2013b).

MemSource Cloud je rešitev podjetja MemSource in ponuja celovito prevajalsko okolje, namenjeno prevajalcem, prevajalskim agencijam in prevajalskim oddelkom v podjetjih.

Vključuje funkcionalnosti baze prevodov, terminološke baze in prevajalskega namizja. Poleg tega ima tudi integrirano strojno prevajanje (Memsources, 2013a).

Najbolj pomembni napovedi članka (Memsources, 2013b) sta nedvomno:

- prevajalske agencije in prevajalski oddelki bodo lahko prevajalske in lokalizacijske projekte izvajali s pomočjo optimalnega delovnega toka med obema sistemoma,
- odstranjenih bo veliko ponavljajočih se opravil, ki jih morajo projektni managerji izvajati tako v svojem sistemu za projektni manager in nato še v računalniškem prevajalskem orodju.

### 3.4 Opredelitev sistemskih zahtev – splošno

#### 3.4.1 Seznam dogodkov in primerov uporabe v tabeli dogodkov

##### Projektni management

Modul projektni management je najbolj obsežen. Posamezni dogodki in primeri uporabe so opredeljeni v Tabeli 7.

*Tabela 7: Seznam dogodkov in primerov uporabe za projektni management*

	<b>Dogodek</b>	<b>Sprožilec</b>	<b>Izvor</b>	<b>Primer uporabe</b>	<b>Odgovor</b>	<b>Cilj</b>	<b>Vrsta</b>
1.	Stranka odda naročilo.	Novo naročilo	Stranka	Odpiranje novega projekta	Nov projekt Potrdilo naročila	Projektni manager Stranka	Z
2.	Projektni manager izda ponudbo.	Zahteva za ponudbo	Projektni manager	Izdaja ponudbe	Ponudba	Stranka	Z
3.	Stranka potrdi ponudbo.	Potrditev ponudbe	Stranka	Potrditev ponudbe	Potrditev projekta	Projektni manager	Z
4.	Stranka zahteva spremembo ponudbe.	Sprememba ponudbe	Stranka	Sprememba ponudbe	Nova ponudba	Stranka	Z
5.	Stranka prekliče naročilo.	Zavrnitev ponudbe	Stranka	Preklic naročila	Arhiviran projekt	Projektni manager	Z
6.	Priprava časovnega načrta projekta	Nov projekt	Projektni manager	Časovni načrt projekta	Časovni načrt projekta	Projektni manager	
7.	Prerazporeditev prevajalcev	Neusklajene potrebe		Prerazporeditev prevajalcev	Nov časovni načrt projekta	Projektni manager	I

se nadaljuje

nadaljevanje

	<b>Dogodek</b>	<b>Sprožilec</b>	<b>Izvor</b>	<b>Primer uporabe</b>	<b>Odgovor</b>	<b>Cilj</b>	<b>Vrsta</b>
8.	Projektni manager dodeli verigo opravil.	Izdelan časovni načrt	Projektni manager	Dodelitev verige opravil	Dodeljena opravila	Vsi člani projekta	Z
9.	Projektni manager izda naročilnico zunanjemu izvajalcu.	Naročene storitve pri zunanjem izvajalcu	Projektni manager	Izdaja naročilnice zunanjemu izdajalcu	Naročilnica za zunanjega izvajalca	Zunanji izvajalec	Z
10.	Uporabnik sprejme opravilo.	Dodeljeno opravilo	Uporabnik	Izvedba opravila	Potrdilo o sprejetem opravilu	Projektni manager	Z
11.	Uporabnik zaključi opravilo.	Izvedeno opravilo	Uporabnik	Zaključitev opravila	Potrdilo o zaključenem opravilu  Obvestilo naslednjemu uporabniku	Projektni manager  Naslednji uporabnik na projektu	Z
12.	Uporabnik v sistem vnese podatke o opravljenem delu.	Opravljen delo	Uporabnik	Vnos podatkov o opravljenem delu	Podatki opravljenega dela	Projektni manager	Z
13.	Sistem izvede opravilo.	Dodeljeno opravilo		Izvedba systemskega opravila	Potrdilo o izvedenem systemskem opravilu	Projektni manager  Drugi uporabniki na projektu	I
14.	Projektni manager zaključi projekt.	Oddani prevedeni dokumenti	Projektni manager	Zapiranje projekta	Obračun storitev	Računovodstvo	Z
15.	Stranka plača račun.	Plačan račun	Računovodstvo	Plačilo računa	Potrdilo plačila	Direktor	Z
16.	Stranka reklamira opravljene storitve.	Pomanjkljivo opravljene storitve	Stranka	Reklamacija naročila	Reklamacija	Direktor Projektni manager	Z
17.	Zunanji izvajalec izda račun za opravljene storitve-	Opravljene storitve z naročilnice	Zunanji izvajalec	Prejem računa zunanjega izvajalca	Račun zunanjega izvajalca	Projektni manager	Z

**Legenda:**

- Z – zunanji dogodek
- Č – časovni dogodek
- I – interni dogodek (tudi dogodek stanja)

## Poročanje

Tabela 8 prikazuje seznam dogodkov in primerov uporabe za modul poročanja.

*Tabela 8: Seznam dogodkov in primerov uporabe za poročanje*

	<b>Dogodek</b>	<b>Sprožilec</b>	<b>Izvor</b>	<b>Primer uporabe</b>	<b>Odgovor</b>	<b>Cilj</b>	<b>Vrsta</b>
1.	Projektni manager priključuje poročilo napredka projekta.	Posodobitev stanja projekta	Računalniški program za prevajanje ali projektni manager	Poročilo o napredku projekta	Priprava poročila	Projektni manager	Z
2.	Projektni manager priključuje poročilo o razporejenosti in razpoložljivosti prevajalcev.	Zahteva za poročilo	Projektni manager	Poročilo o razporejenosti in razpoložljivosti prevajalcev	Priprava poročila	Projektni manager	Z
3.	Direktor priključuje finančno poročilo projekta.	Zahteva za poročilo	Direktor	Finančno poročilo	Priprava poročila	Direktor Projektni manager	Z
4.	Čas za mesečno poročilo o realizaciji projektov	Konec meseca		Priklic mesečnega poročila	Priprava poročila	Direktor Projektni manager	I

### Legenda:

- Z – zunanji dogodek
- Č – časovni dogodek
- I – interni dogodek (tudi dogodek stanja)

## Upravljanje s človeškimi viri

Tabela 9 prikazuje seznam dogodkov in primerov uporabe za modul upravljanja s človeškimi viri. Prva dva dogodka sta povezana z ustvarjanjem novega človeškega vira v sistemu in posodabljanjem matičnih podatkov obstoječega vira, pri zadnjem pa gre za pripravo obračuna plač.

*Tabela 9: Seznam dogodkov in primerov uporabe za upravljanje s človeškimi viri*

	<b>Dogodek</b>	<b>Sprožilec</b>	<b>Izvor</b>	<b>Primer uporabe</b>	<b>Odgovor</b>	<b>Cilj</b>	<b>Vrsta</b>
1.	Projektni manager ustvari nov človeški vir v sistemu.	Nov človeški vir	Direktor, projektni manager	Nov človeški vir			Z
2.	Projektni manager vzdržuje matične podatke človeškega vira.	Sprememba matičnih podatkov človeškega vira	Projektni manager	Posodobitev matičnih podatkov človeškega vira			Z
3.	Direktor pripravi obračun plač za zaposlene.	Obračun plač	Direktor	Priprava obračuna plač	Obračun plan	Računovodstvo	Z

**Legenda:**

- Z – zunanji dogodek
- Č – časovni dogodek
- I – interni dogodek (tudi dogodek stanja)

**Upravljanje s strankami**

Modul upravljanja s strankami vsebuje dva dogodka in primera uporabe, prikazana v Tabeli 10. Podobno kot pri upravljanju s človeškimi viri, sta povezana z ustvarjanjem nove stranke v sistemu in posodabljanjem matičnih podatkov obstoječe stranke.

*Tabela 10: Seznam dogodkov in primerov uporabe za upravljanje s strankami*

	<b>Dogodek</b>	<b>Sprožilec</b>	<b>Izvor</b>	<b>Primer uporabe</b>	<b>Odgovor</b>	<b>Cilj</b>	<b>Vrsta</b>
1.	Projektni manager ustvari novo stranko v sistemu.	Nova stranka	Projektni manager	Nova stranka			Z
2.	Projektni manager vzdržuje matične podatke stranke.	Sprememba matičnih podatkov stranke	Projektni manager	Posodobitev matičnih podatkov stranke			Z

**Legenda:**

- Z – zunanji dogodek
- Č – časovni dogodek
- I – interni dogodek (tudi dogodek stanja)

### 3.4.2 Opredelitev stvari

V Tabeli 11 sem opredelil stvari, o katerih mora informacijski sistem hraniti informacije. Vsaka stvar je opremljena s kratico modula in podrobnejšo opredelitvijo.

Tabela 11: Opredelitev stvari

	<b>Samostalnik</b>	<b>Modul</b>	<b>Opredelitev</b>
1.	Projekt	PM	Rozman in Stare (2008, str. 7) projekt opredelita kot »podjem (širšo dejavnost, delo) med seboj povezanih zaposlenih, sredstev in aktivnosti, za katerega so značilni neponovljivost projektnega procesa in enkratnost proizvoda ali storitve, s tem časovna omejenost celotne dejavnosti in sodelovanje različnih sodelavcev in sredstev v projektu.«
2.	Dokument	PM	Dokument pomeni nosilni medij prevajalskega projekta (Slovenski inštitut za standardizacijo, 2007, str. 5). V našem primeru je dokument splošna oznaka za besedilo, ki je predmet prevoda, ni pa vezano na fizično obliko. Dokument je lahko fizična datoteka ali pa besedilo za prevod v strankinem lastnem sistemu.
3.	Ponudba	PM	Ponudba pomeni dokument, ki ga izda izvajalec prevajalskih storitev naročniku, in vsebuje naročene storitve, njihovo ceno in rok dobave.
4.	Opravilo	PM	Opravilo pomeni nalogo, ki jo mora izvesti izvajalec opravila, in je določena s pripadajočim projektom, vrsto opravila, obsegom dela in rokom izvedbe.
5.	Opravljen delo	PM	Opravljen delo je določeno s pripadajočim projektom, porabljenim časom za določeno opravilo in količino prevedenega besedila.
6.	Račun	PM	Račun je izdan za opravljene storitve, ki jih je naročila stranka, in izkazuje ceno storitve, količino in datum opravljene storitve.
7.	Storitev	PM	Storitev pomeni delo, ki ga stranka naroči izvajalcu prevajalskih storitev. Predmet storitve je dokument.
8.	Izvirni jezik	PM	Jezik, v katerem je napisano izvirno besedilo dokumenta (Slovenski inštitut za standardizacijo, 2007, str. 6).
9.	Ciljni jezik	PM	Jezik, v katerega se prenaša izvirno besedilo dokumenta (Slovenski inštitut za standardizacijo, 2007, str. 6).

se nadaljuje

nadaljevanje

	<b>Samostalnik</b>	<b>Modul</b>	<b>Opredelitev</b>
10.	Obrazec za dvojezični pregled	PM	Obrazec za dvojezični pregled se uporablja za beleženje in kategorizacijo napak med pregledovanjem prevoda besedila. Obrazec poleg napak vsebuje podatke o pripadajočem projektu in dokumentu. Uporablja se za zagotavljanje kakovosti prevoda in ocenjevanje prevajalcev.
11.	Naročilnica za zunanjšega izvajalca	PM	Naročilnica za zunanjšega izvajalca se uporablja za obračun storitev, ki jih projektni manager naroči pri zunanjem izvajalcu in je osnova za račun zunanjšega izvajalca.
12.	Račun zunanjšega izvajalca	PM	Račun zunanjšega izvajalca vsebuje obračun storitev, ki jih zunanji izvajalec izvede na podlagi naročilnice.
13.	Merska enota	PM	Merska enota je vrsta količine, ki se uporablja pri načrtovanju prevajalskih projektov, razporejanju dela in merjenju opravljenega prevajalskega dela. Primeri merskih enot so: beseda, termin, vrstica, ura in podobno (CeFormT d.o.o., 2012f, str. 1).
14.	Stopnja ujemanja	PM	Analiza besedila pove, kakšno je ujemanje posameznih segmentov (stavkov ali odstavkov) z bazo prevodov. Segmenti so razvrščeni v različne kategorije, glede na stopnjo ujemanja s segmenti v bazi prevodov od 0 % do 100 %.
15.	Cena	PM	Cena določa vrednost posamezne storitve.
16.	Norma	PM	Norma je pričakovana količina ustvarjenega prevoda v eni uri. Norma je izražena v količini ustvarjenih (prevedenih) merskih enot v eni uri prevajalskega dela.
17.	Človeški vir	UČV	Izvajalec prevajalskih storitev in drugih storitev, ki je lahko redno zaposlen ali zunanji izvajalec.
18.	Zunanji izvajalec	UČV	Izvajalec prevajalskih storitev, ki ni redno zaposlen v podjetju in občasno izvaja naročene storitve.
19.	Zaposleni	UČV	Izvajalec prevajalskih storitev, ki ima status redno zaposlenega.
20.	Direktor	UČV	Najvišji poslovodja, ki vodi podjetje.
21.	Projektni manager	UČV	Poslovodja, ki vodi prevajalski tim in skrbi za izvajanje prevajalskih projektov.
22.	Prevajalec	UČV	Prevajalec je izvajalec prevajalskih storitev, ki ima status redno zaposlenega.

se nadaljuje

nadaljevanje

	<b>Samostalnik</b>	<b>Modul</b>	<b>Opredelitev</b>
23.	Področni svetovalec	UČV	Področni svetovalec je redno zaposleni, ki zagotavlja strokovno pomoč pri prevajanju, predvsem pri tehnično zahtevnih in specifičnih besedilih (CeFormT d.o.o., 2012j, str. 9).
24.	Organizator informacijskih sistemov	UČV	Organizator informacijskih sistemov je redno zaposleni, ki zagotavlja nemoteno delovanje informacijskih sistemov, po potrebi pa sodeluje z zunanjim podjetjem, ki upravlja določene dele informacijskega sistema (CeFormT d.o.o., 2012j, str. 10).
25.	Redaktor	UČV	Redaktor je oseba, ki revidira preveden dokument (Slovenski inštitut za standardizacijo, 2007, str. 6).
26.	Stranka	US	Stranka je pravna oseba, ki naroči storitve pri izvajalcu prevajalskih storitev.

**Legenda:**

- PM – projektni management
- UČV – upravljanje s človeškimi viri
- US – upravljanje s strankami

### 3.4.3 Diagrami razredov

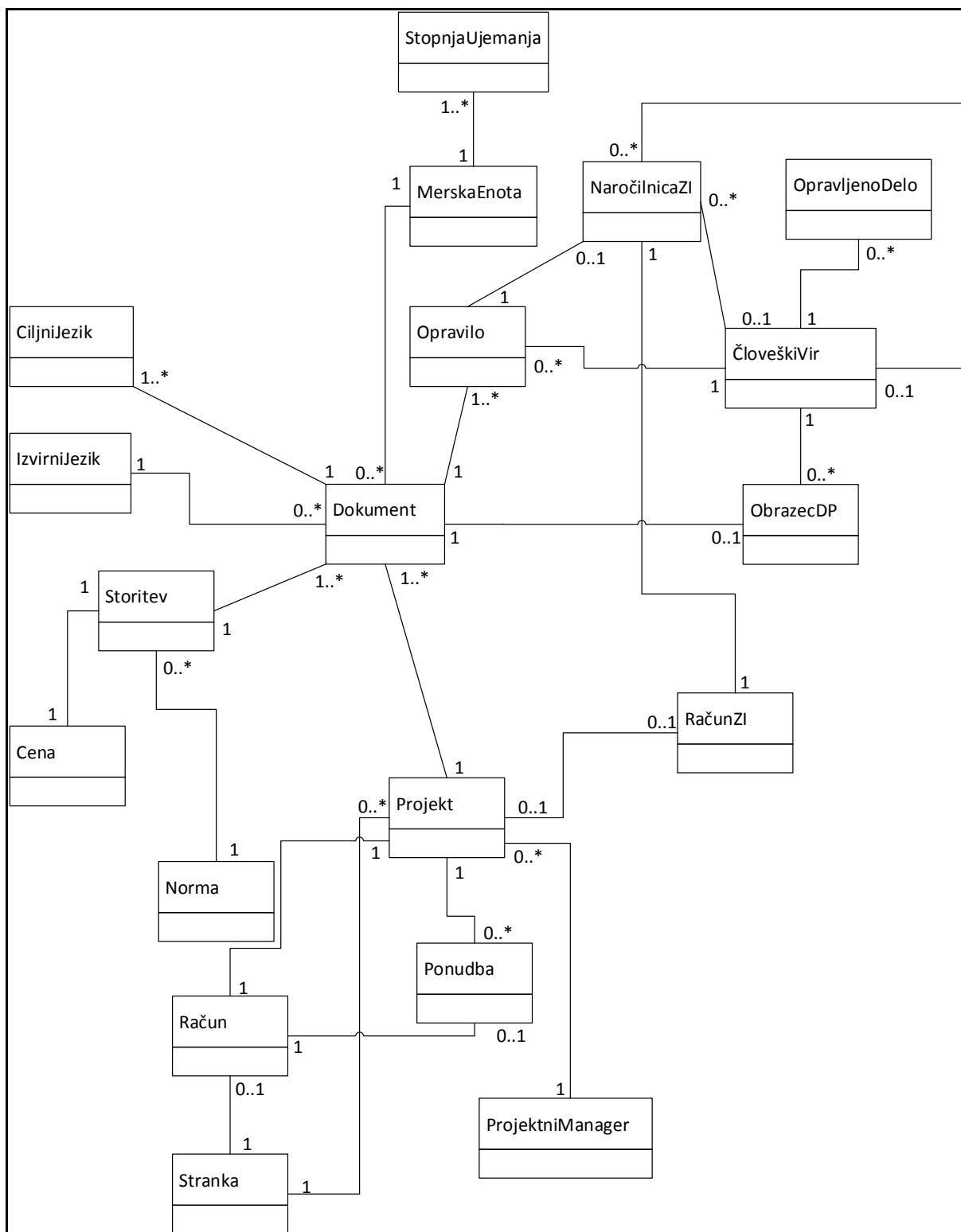
V nadaljevanju sta prikazana dva diagrama razredov in sicer za projektni management in za upravljanje človeških virov. Za poročanje diagram razredov ni relevanten, za upravljanje s strankami pa ni potreben, ker gre dejansko samo za razred stranke, ki je vključen v diagram razredov za projektni management.



## Projektni management

Slika 19 prikazuje diagram razredov za vse stvari iz modula projektni management. Vključen je tudi razred stranke iz modula upravljanje s strankami.

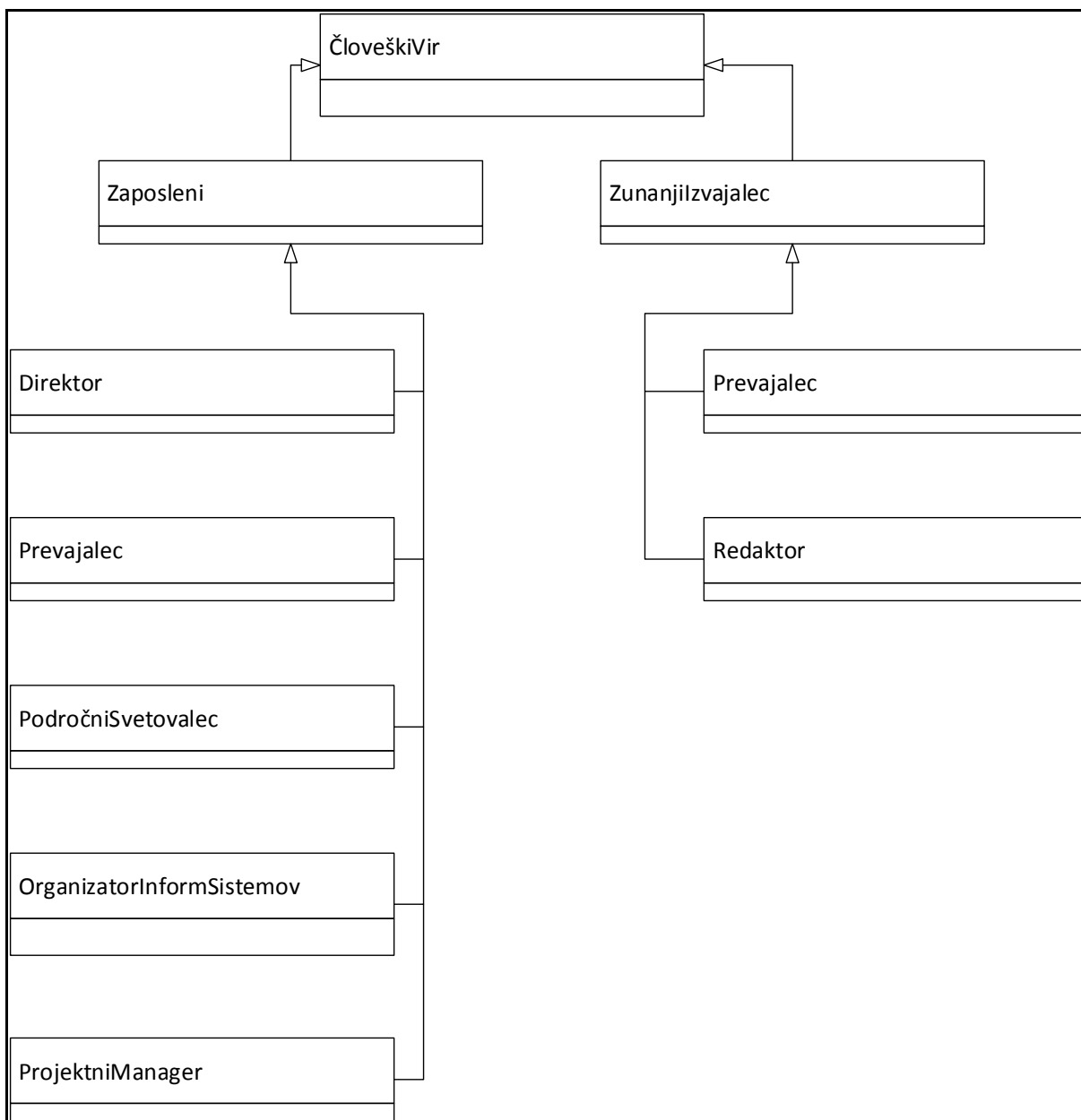
Slika 19: Diagram razredov za projektni management



## Upravljanje s človeškimi viri

Razredi objektov modula upravljanja s človeškimi viri so prikazani na Sliki 20, kjer je vidno tudi dedovanje med posameznimi razredi objektov.

Slika 20: Diagram razredov za upravljanje s človeškimi viri



### 3.5 Opredelitev sistemskih zahtev – objektni pristop

#### 3.5.1 Opisi primerov uporabe

##### Projektni management

V tabelah od 12 do 16 so navedeni podrobni opisi primerov uporabe za modul projektnega managementa. Primer uporabe odpiranje novega projekta vključuje dva scenarija in sicer odpiranje novega projekta z in brez zahteve za ponudbo.

Tabela 12: Opis primera uporabe – odpiranje novega projekta

Ime primera uporabe:	Odpiranje novega projekta	
Scenarij:	Naročilo brez zahteve za ponudbo	
Prožilni dogodek:	E-sporočilo kontaktne osebe stranke z naročilom prevoda z zahtevo za ponudbo	
Kratek opis:	Kontaktna oseba stranke z e-sporočilom naroči prevod dokumenta. Pred začetkom projekta kontaktna oseba stranke zahteva ponudbo.	
Akterji:	Projektni manager	
Povezani primeri uporabe:	/	
Deležniki:	Projektni manager, direktor, organizator informacijskih sistemov	
Predpogoji:	<ul style="list-style-type: none"><li>• Stranka, njena kontaktna oseba in specifikacija storitev s cenikom morajo obstajati v sistemu.</li><li>• Obdelovalec dokumentov mora na zahtevo projektnega manager pregledati in po potrebi obdelati dokument.</li></ul>	
Zaključni pogoji:	<ul style="list-style-type: none"><li>• Odprt mora biti projekt z naslednjimi specifikacijami: naročene storitve, podatki o stranki in kontaktni osebi, rok oddaje, relevantni dokumenti za prevod.</li></ul>	
Tok aktivnosti:	Akter	Sistem
	1. Stranka odda naročilo.  2. Projektni manager odpre projekt, vnese dokument za prevod in izbere naročene storitve za dokument.	2. a. Sistem ponudi obstoječe storitve.

se nadaljuje

nadaljevanje

Tok aktivnosti:	Akteur	Sistem
	<p>3. Za dokument navede tudi vse ostale specifikacije: rok prevoda, kontaktno osebo stranke, jezikovno smer in področje prevajanja.</p> <p>4. Projektni manager določi bazo prevodov in terminološko bazo.</p> <p>5. Projektni manager zažene proces ustvarjanja projekta v računalniškem prevajalskem programu.</p> <p>6. Projektni manager nadaljuje s projektom in pripravo časovnega načrta.</p>	<p>3. a. Sistem ponudi na izbiro obstoječe kontaktne osebe stranke, jezikovne smeri, področje prevajanja in koledar za izbiro roka za vsak dokument.</p> <p>4. a. Sistem ponudi na izbiro obstoječe baze prevodov in terminološke baze podatkov ter prikaže potrdilo o tem.</p> <p>4. b. Sistem določi status projekta na »Odprt« in shrani projekt v bazo</p> <p>5. a. Sistem ustvari projekt v računalniškem prevajalskem programu in tam pridobi podatke o obsegu prevoda (analiza strukture besedila glede na bazo prevodov) in jih shrani v projekt.</p> <p>5. b. Na podlagi podatkov obsega projekta sistem za vsak dokument izračuna ceno prevoda in pričakovani obseg dela po danih normativih.</p>

se nadaljuje

nadaljevanje

Pogoji izjem:	<p>3. En projekt lahko vsebuje več dokumentov, vendar morajo biti v sklopu istega naročila ter imeti isto stranko in isti rok oddaje.</p> <p>4. Če ustreznih baz ni na razpolago, jih projektni manager zahteva pri organizatorju informacijskih sistemov.</p> <p>5. Samo če gre za fizično datoteko, drugače korak 5 ni relevanten in so podatki obsega pridobljeni iz drugih virov (na primer strankinega lastnega sistema ali orodja).</p>
---------------	---

Tabela 13: Opis primera uporabe – odpiranje novega projekta z zahtevo za ponudbo

Ime primera uporabe:	Odpiranje novega projekta	
Scenarij:	Naročilo z zahtevo za ponudbo	
Prožilni dogodek:	E-sporočilo kontaktne osebe stranke z naročilom prevoda z zahtevo za ponudbo	
Kratek opis:	Kontaktna oseba stranke z e-sporočilom naroči prevod dokumenta. Pred začetkom projekta kontaktna oseba stranke zahteva ponudbo.	
Akterji:	Projektni manager	
Povezani primeri uporabe:	Vključuje: <i>Izdaja ponudbe, Potrditev ponudbe, Sprememba ponudbe, Preklic naročila.</i>	
Deležniki:	Projektni manager, direktor, organizator informacijskih sistemov, obdelovalec dokumentov	
Predpogoji:	<ul style="list-style-type: none"> <li>Stranka, njena kontaktna oseba in specifikacija storitev s cenikom morajo obstajati v sistemu.</li> </ul>	
Zaključni pogoji:	<ul style="list-style-type: none"> <li>Odprt mora biti projekt z naslednjimi specifikacijami: naročene storitve, podatki o stranki in kontaktni osebi, rok oddaje, relevantni dokumenti za prevod.</li> <li>Ponudba in naročilo morata biti potrjena ali preklicana.</li> </ul>	
Tok aktivnosti:	Akter	Sistem
	<p>1. Stranka odda povpraševanje.</p> <p>2. Projektni manager odpre projekt in vnese dokument za prevod in izbere naročene storitve za dokument.</p>	<p>2. a. Sistem ponudi obstoječe storitve.</p>

se nadaljuje

nadaljevanje

Tok aktivnosti:	Akteur	Sistem
	<p>3. Za dokument navede tudi vse ostale specifikacije: rok prevoda, kontaktno osebo stranke, jezikovno smer in področje prevajanja.</p> <p>4. Projektni manager določi bazo prevodov in terminološko bazo.</p> <p>5. Projektni manager zažene proces ustvarjanja projekta v računalniškem prevajalskem programu.</p> <p>6. Projektni manager sproži pripravo ponudbe za stranko.</p> <p>7. Po strankini potrditvi ponudbe projektni manager nadaljuje s projektom in pripravo časovnega načrta.</p>	<p>3. a. Sistem ponudi na izbiro obstoječe kontaktne osebe stranke, jezikovne smeri, področje prevajanja in koledar za izbiro roka za vsak dokument.</p> <p>4. a. Sistem ponudi na izbiro obstoječe baze prevodov in terminološke baze.</p> <p>4. b. Sistem določi status projekta na »Odprt« in shrani projekt v bazo podatkov in prikaže potrdilo o tem.</p> <p>5. a. Sistem ustvari projekt v računalniškem prevajalskem programu, tam pridobi podatke o obsegu prevoda (analiza strukture besedila glede na bazo prevodov) in jih shrani v projekt.</p> <p>5. b. Na podlagi podatkov obsega projekta sistem za vsak dokument izračuna ceno prevoda in pričakovani obseg dela glede na dane normative.</p> <p>6. a. Sistem pripravi ponudbo v PDF-obliki in jo pošlje kontaktni osebi stranke.</p>

se nadaljuje

nadaljevanje

Pogoji izjem:	<p>3. En projekt lahko vsebuje več dokumentov, vendar morajo biti v sklopu istega naročila ter imeti isto stranko in isti rok oddaje.</p> <p>4. Če ustreznih baz ni na razpolago, jih projektni manager zahteva pri organizatorju informacijskih sistemov.</p> <p>5. Samo če gre za fizično datoteko, drugače korak 5 ni relevanten in so podatki obsega pridobljeni iz drugih virov (na primer strankinega lastnega sistema ali orodja).</p> <p>7. Če stranka zahteva spremembo ponudbe (povpraševanje po popustu in podobno), projektni manager izvede ustrezne spremembe v sistemu in sproži ponovno pošiljanje ponudbe, lahko:</p> <p>a) stranka potrdi ponudbo in projektni manager nadaljuje s projektom,</p> <p>b) stranka zavrne ponudbo in prekliče naročilo.</p>
---------------	--

Tabela 14: Opis primera uporabe – časovni načrt projekta

Ime primera uporabe:	Časovni načrt projekta
Scenarij:	/
Prožilni dogodek:	Nov projekt, ustvarjen v sistemu, za katerega je treba pripraviti načrt izvedbe.
Kratek opis:	Projektni manager z vsemi člani projektnega tima pripravi časovni načrt izvedbe projekta in ga umesti v celoten časovni načrt izvedbe vseh projektov.
Aktejri:	Projektni manager
Povezani primeri uporabe:	Vključuje: <i>Prerazporeditev prevajalcev, Dodelitev verige opravil, Izdaja naročilnice zunanjemu izdajalcu.</i>
Deležniki:	Projektni manager, člani projektnega tima
Predpogoji:	<ul style="list-style-type: none"> <li>• Projekt mora biti ustvarjen v sistemu in računalniškem prevajalskem programu.</li> <li>• V sistemu morajo biti na razpolago podatki obsega projekta (dokumentov), iz katerih se izračuna predviden potreben čas izvedbe projekta (potrebne ure prevajanja).</li> <li>• V sistemu morajo biti podatki o razpoložljivosti prevajalcev.</li> </ul>

se nadaljuje

nadaljevanje

Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Izdelan mora biti časovni načrt izvedbe projekta in umeščen v celotni časovni načrt.</li> <li>• Izpolnjene morajo biti potrebe aktualnega projekta in zagotovljena neokrnjenost ostalih projektov (v primeru prerazporejanja prevajalcev z drugih projektov na aktualni projekt).</li> <li>• V sistemu mora biti dodeljena veriga opravil za izvedbo projekta, ki se mora odražati tudi v računalniškem prevajalskem programu.</li> <li>• O dodelitvah morajo biti obveščeni vsi člani projektnega tima po e-pošti.</li> <li>• Če na projektu sodelujejo zunanji izvajalci, morajo dobiti naročilnice za naročene storitve.</li> </ul>	
Tok aktivnosti:	Akteur	Sistem
	<p>1. Projektni manager določi izvajalce posameznih opravil. Vsako opravilo lahko izvaja eden ali več prevajalcev.</p> <p>2. Projektni manager določi datum začetka prvega opravila v verigi opravil.</p> <p>3. Projektni manager izda naročilnice vsem zunanjim izvajalcem, če so člani projektnega tima.</p>	<p>1. a. Sistem ponudi na izbiro samo prevajalce, ki ustrezajo potrebam projekta (področje prevajanja, besedilne vrste in jezikovne kombinacije, časovna razpoložljivost).</p> <p>1. b. Sistem prikaže razporejenost vseh prevajalcev na projekte v koledarski matriki (prevajalci, projekti in dnevi).</p> <p>2. a. V verigi opravil sistem na podlagi potrebnega števila ur in datuma začetka prvega opravila izračuna datum konca vsakega opravila.</p> <p>2. b. Sistem shrani časovni načrt izvedbe v projekta.</p>

se nadaljuje



nadaljevanje

Pogoji izjem:	<p>1. a. Če sistem javi, da glede na potrebe projekta ni ustreznega prevajalca, ki bi projekt lahko zaključil do roka, mora projektni manager izvesti naslednje ukrepe:</p> <ul style="list-style-type: none"><li>• Če je časovno prekrivanje z drugimi projekti relativno majhno, lahko izvedbo zagotovi z nadurnim delom in temu ustrezno prilagodi časovni načrt izvedbe.</li><li>• Začasno lahko prerazporedi prevajalca z drugega projekta, pri čemer mora takoj korigirati načrt izvedbe projekta, kateremu je odvzel razporejenega prevajalca.</li><li>• Začasno prerazporedi več prevajalcev z drugega (-ih) projekta(-ov), pri čemer mora takoj korigirati načrt izvedbe projekta(-ov), kateremu(-im) je odvzel razporejenega(-e) prevajalca(-e).</li></ul> <p>Vse zgoraj navedene spremembe projektni manager izvede na enem mestu v sistemu (koledarski prikaz vseh projektov in razpoložljivih človeških virov, ki služi tudi kot »peskovnik« za preverjanje različnih možnosti), nato pa sproži korekcijo časovnega načrta izvedbe za vse prizadete projekte. Sistem uveljavi spremembe in obvesti vse člane projektnega tima.</p>
---------------	---

Tabela 15: Opis primera uporabe – izvedba opravila

Ime primera uporabe:	Izvedba opravila	
Scenarij:	/	
Prožilni dogodek:	Dodeljeno opravilo	
Kratek opis:	Vsi člani projektnega tima izvedejo dodeljena opravila in to zabeležijo v sistemu, ki skrbi za obveščanje o poteku projekta.	
Akterji:	Človeški vir	
Povezani primeri uporabe:	Vključuje: <i>Zaključitev opravila, Vnos podatkov o opravljenem delu, Izvedba systemskega opravila.</i>	
Deležniki:	Izvajalec opravila, projektni manager	
Predpogoji:	<ul style="list-style-type: none"> <li>• Projekt mora biti ustvarjen v računalniškem prevajalskem programu.</li> <li>• V sistemu in računalniškem prevajalskem programu morajo biti dodeljena vsa opravila.</li> </ul>	
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Opravilo je sprejeto, izvedeno in zaključeno.</li> <li>• Izvedena so vsa potrebna systemska opravila.</li> <li>• Projektni manager je obveščen o izvedenem opravilu.</li> </ul>	
Tok aktivnosti:	Akter	Sistem
	<p>1. Izvajalec sprejme dodeljeno opravilo.</p> <p>2. Po končani izvedbi izvajalec zaključi opravilo.</p> <p>3. Koraka 1 in 2 se izvajata, dokler niso vsa opravila v verigi opravil zaključena.</p> <p>4. Prevajalec v svojo evidenco delovnega časa vnese opravljeno opravilo, vrsto dela in porabljen čas.</p>	<p>1. a. Sistem pri opravilu v projektu označi, da je izvajalec začel izvajati opravilo.</p> <p>2. a. Sistem pri opravilu v projektu označi, da je bilo opravilo izvedeno.</p> <p>2. b. Sistem o končanem opravilu obvesti projektne managerja in naslednjega člana projektnega tima, ki mora izvesti naslednje opravilo.</p> <p>3. a. Izvedba morebitnih systemskih avtomatičnih opravil v računalniškem prevajalskem orodju.</p> <p>4. a. Sistem shrani vnesene podatke.</p>
Pogoji izjem:	/	

Tabela 16: Opis primera uporabe – zapiranje projekta

Ime primera uporabe:	Zapiranje projekta	
Scenarij:	/	
Prožilni dogodek:	Oddani vsi prevedeni dokumenti	
Kratek opis:	Ko je prevod dokumentov končan, projektni manager odda dokumente stranki in v projekt vnese datum oddaje in kontaktno osebo, ki je prevzela dokumente. Če stranka reklamira opravljene storitve, se izvedejo popravki. Projektni manager preveri obračunske podatke in sproži posredovanje obračuna v računovodstvo. Računovodstvo v sistem vnese podatke o plačilu računa. Po prejemu računa zunanjšega izvajalca ga direktor posreduje v računovodstvo v plačilo in to označi v sistemu.	
Akterji:	Projektni manager, računovodstvo, direktor	
Povezani primeri uporabe:	Vključuje: <i>Plačilo računa, Reklamacija naročila, Prejem računa zunanjšega izvajalca.</i>	
Deležniki:	Projektni manager, direktor, računovodstvo, stranka, zunanji izvajalec	
Predpogoji:	<ul style="list-style-type: none"> <li>• Prevedeni dokumenti so oddani.</li> </ul>	
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Stranka potrdi prevzem prevedenih dokumentov.</li> <li>• Stranka plača izdan račun.</li> <li>• Računovodstvo plača račun zunanjšega izvajalca.</li> <li>• Vse zaključne aktivnosti so zavedene v sistemu in projekt zaprt.</li> </ul>	
Tok aktivnosti:	Akter	Sistem
	<p>1. Projektni manager vnese podatke o oddaji dokumentov (datum in čas, prevzemnika dokumentov).</p> <p>2. Projektni manager izvede kontrolo obračunskih podatkov in sproži posredovanje v računovodstvo.</p> <p>3. Računovodstvo v sistemu označi datum izdaje in plačila računa.</p>	<p>2. a. Sistem posreduje obračun projekta v računovodstvo za izdajo računa.</p> <p>3. a. Sistem o izdaji in plačilu računa obvesti direktorja.</p>

se nadaljuje

nadaljevanje

Tok aktivnosti:	Akter	Sistem
	<p>4. Projektni manager v sistem zavede račun zunanjega izvajalca.</p> <p>5. Direktor sproži posredovanje računa zunanjega izvajalca v plačilo v računovodstvo.</p> <p>6. Računovodstvo v sistemu označi datum plačila računa.</p>	<p>4. a. Sistem izvede validacijo računa in obvesti direktorja o prispelem računu zunanjega izvajalca.</p> <p>5. a. Sistem posreduje račun zunanjega izvajalca v plačilo v računovodstvo.</p>
Pogoji izjem:	1. Če stranka reklamira opravljene storitve, projektni manager na projektu označi reklamacijo in po popravkih ter ponovni oddaji spet označi datum ponovne oddaje dokumentov.	

## Poročanje

V tabelah od 17 do 19 so navedeni podrobni opisi primerov uporabe za modul poročanja.

*Tabela 17: Opis primera uporabe – poročilo o napredku projekta*

Ime primera uporabe:	Poročilo o napredku projekta
Scenarij:	/
Prožilni dogodek:	Posodobitev stanja projekta
Kratek opis:	Projektni manager med potekom projekta prikliče poročilo o napredku projekta. To pomeni priklic podatkov o opravljenih urah vseh članov projektnega tima ter količine opravljenega dela (prevedenega besedila). Prav tako je treba ugotoviti še preostalo količino za prevod in predvideno število potrebnih ur. Člani projektnega tima dobijo vpogled v poročilo.
Akterji:	Projektni manager
Povezani primeri uporabe:	/
Deležniki:	Projektni manager, člani projektnega tima
Predpogoji:	<ul style="list-style-type: none"> <li>• Prevajalski projekt je v teku.</li> </ul>
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Izdelano poročilo napredka projekta, v katerega imajo vpogled vsi člani projektnega tima.</li> </ul>

se nadaljuje

nadaljevanje

Tok aktivnosti:	Akter	Sistem
	1. Projektni manager v sistemu izbere projekt in časovno obdobje, za katero želi priklicati poročilo o napredku.	1. a. Sistem pridobi podatke o opravljenem delu (čas in količina) vseh članov projektnega tima. 1. b. Sistem izračuna realizacijo glede na pridobljene podatke in planske podatke. 1. c. Sistem prikaže preostalo količino za prevod. 1. d. Sistem obvesti vse člane projektnega tima o pripravljenem poročilu.
Pogoji izjem:	1. a. Če je vir podatkov integriran prevajalski računalniški program, sistem podatke o količini pridobi sam, drugače jih mora vnesti projektni manager (če je vir zunanji strankin sistem).	

*Tabela 18: Opis primera uporabe – finančno poročilo projekta*

Ime primera uporabe:	Finančno poročilo projekta	
Scenarij:	/	
Prožilni dogodek:	Zahteva za poročilo	
Kratek opis:	Direktor prikliče finančno poročilo projekta, ki je lahko še v teku ali pa je že zaključen. Osnova finančnega poročila so porabljene ure človeških virov in ustreznih cen ter vrednosti zaračunanih storitev.	
Akterji:	Direktor	
Povezani primeri uporabe:	/	
Deležniki:	Direktor	
Predpogoji:	<ul style="list-style-type: none"> <li>• Projekt je v teku ali zaključen.</li> </ul>	
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Pripravljeno je delno ali končno finančno poročilo projekta.</li> </ul>	
Tok aktivnosti:	Akter	Sistem
	1. Direktor izbere projekt in sproži pripravo finančnega poročila projekta.	1. a. Sistem pridobi podatke o opravljenih urah vseh članov projektnega tima na projektu.  1. b. Sistem izračuna izdatke in prihodke projekta.  1. c. Sistem prikaže finančno poročilo.
Pogoji izjem:	/	

Tabela 19: Opis primera uporabe – priklic mesečnega poročila

Ime primera uporabe:	Priklic mesečnega poročila	
Scenarij:		
Prožilni dogodek:	Konec meseca	
Kratek opis:	Sistem na začetku meseca ustvari poročilo o realizaciji prevodov preteklega meseca. Poročilo prikaže realizacijo po prevajalcih in projektih, tako da primerja planski čas z dejansko porabljenim časom v povezavi s prevedeno količino.	
Akterji:		
Povezani primeri uporabe:		
Deležniki:	Direktor, projektni manager, prevajalci	
Predpogoji:	<ul style="list-style-type: none"> <li>• Mesec, za katerega se pripravi poročilo, je zaključen.</li> <li>• Če kakšen projekt ni zaključen, je treba ustrezno razmejiti njegove količine za prevod med dvema mesecema.</li> </ul>	
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Mesečno poročilo je pripravljeno in vsi člani o tem obveščeni.</li> </ul>	
Tok aktivnosti:	Akter	Sistem
		<ol style="list-style-type: none"> <li>1. Sistem pridobi podatke za vse projekte o: planskem času, dejansko porabljenem času in količinske podatke.</li> <li>2. Sistem izračuna uspešnost realizacije in pripravi poročilo.</li> <li>3. Sistem obvesti vse deležnike o pripravljenem poročilu.</li> </ol>
Pogoji izjem:	<ol style="list-style-type: none"> <li>1. Če kakšen projekt še ni zaključen, sistem pošlje sporočilo projektnemu managerju, da izvede ustrezne postopke za razmejitev količinskih podatkov med dva meseca.</li> </ol>	

## Upravljanje s človeškimi viri

V tabelah od 20 do 22 so navedeni podrobni opisi primerov uporabe za modul upravljanja s človeškimi viri.

Tabela 20: Opis primera uporabe – nov človeški vir

Ime primera uporabe:	Nov človeški vir	
Scenarij:	/	
Prožilni dogodek:	Nov človeški vir	
Kratek opis:	Ob novi zaposlitvi ali ob novi pogodbi o sodelovanju z zunanjim izvajalcem projektni manager v sistemu ustvari nov človeški vir.	
Akterji:	Direktor, projektni manager	
Povezani primeri uporabe:	/	
Deležniki:	Direktor, projektni manager, vsi človeški viri	
Predpogoji:	<ul style="list-style-type: none"> <li>• Človeški vir še ne obstaja v sistemu.</li> </ul>	
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Človeški vir je ustvarjen v sistemu.</li> </ul>	
Tok aktivnosti:	Akter	Sistem
	1. Direktor začne postopek ustvarjanja novega človeškega vira v sistemu.  2. Direktor vnese matične podatke človeškega vira v sistem: ime in priimek, izobrazbo, spol, datum rojstva, datum sklenitve delovnega razmerja, delovno mesto.  3. Projektni manager v sistem vnese dodatne matične podatke novega človeškega vira: področja prevajanja, jezike, službeni e-naslov.	1. a. Sistem prikaže obrazec za vnos podatkov novega človeškega vira.  2. a. Sistem ustvari nov človeški vir.  3. a. Sistem shrani človeški vir.
Pogoji izjem:	2. Če gre za zunanjega izvajalca, se namesto datuma sklenitve delovnega razmerja vnese datum sklenitve pogodbe o podizvajanju.	

Tabela 21: Opis primera uporabe – posodobitev matičnih podatkov človeškega vira

Ime primera uporabe:	Posodobitev matičnih podatkov človeškega vira	
Scenarij:	/	
Prožilni dogodek:	Sprememba matičnih podatkov človeškega vira	
Kratek opis:	Kadar pride do spremembe matičnih podatkov človeškega vira (drugo delovno mesto, druga vloga v prevajalskem procesu, nove zmožnosti ipd.), direktor ali projektni manager posodobita matične podatke v sistemu.	
Akterji:	Direktor, projektni manager	
Povezani primeri uporabe:	/	
Deležniki:	Direktor, projektni manager, vsi človeški viri	
Predpogoji:	<ul style="list-style-type: none"> <li>• Človeški vir je ustvarjen v sistemu.</li> </ul>	
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Matični podatki človeškega vira so posodobljeni in shranjeni v sistemu.</li> </ul>	
Tok aktivnosti:	Akter	Sistem
	1. Direktor ali projektni manager prikličeta prikaz matičnih podatkov človeškega vira.  2. Direktor ali projektni manager posodobita matične podatke človeškega vira.	1. a. Sistem prikaže obrazec matičnih podatkov človeškega vira.  2. a. Sistem shrani posodobljene podatke človeškega vira.
Pogoji izjem:		

Tabela 22: Opis primera uporabe – priprava obračuna plač

Ime primera uporabe:	Priprava obračuna plač
Scenarij:	/
Prožilni dogodek:	Obračun plač
Kratek opis:	Direktor na začetku meseca pripravi obračun plač na podlagi opravljenega dela v preteklem mesecu in ga pošlje v računovodstvo.
Akterji:	Direktor
Povezani primeri uporabe:	/
Deležniki:	Direktor, zaposleni
Predpogoji:	<ul style="list-style-type: none"> <li>• Mesečno poročilo o realizaciji je pripravljeno.</li> </ul>
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Obračun plač je posredovan v računovodstvo.</li> </ul>

se nadaljuje



nadaljevanje

Tok aktivnosti:	Akter	Sistem
	1. Direktor odpre mesečno poročilo o realizaciji, ki ga je pripravil sistem.  2. Direktor za vsakega zaposlenega izpolni obrazec za obračun plače (plačilna lista).	1. a. Sistem prikaže mesečno poročilo o realizaciji.  2. a. Sistem prikaže posamezne obračune plač.  2. b. Sistem pripravljene obračune plač pošlje v računovodstvo.
Pogoji izjem:	/	

### Upravljanje s strankami

Tabeli 23 in 24 vsebujeta podrobna opisa primerov uporabe za modul upravljanja s strankami.

*Tabela 23: Opis primera uporabe – nova stranka*

Ime primera uporabe:	Nova stranka	
Scenarij:	/	
Prožilni dogodek:	Nova stranka	
Kratek opis:	Ob pridobitvi nove stranke direktor ustvari stranko v sistemu in vnese osnovne matične podatke. Projektni manager vnese podatke o storitvah in kontaktnih osebah.	
Akterji:	Direktor, projektni manager	
Povezani primeri uporabe:	/	
Deležniki:	Direktor, projektni manager	
Predpogoji:	<ul style="list-style-type: none"> <li>• Stranka še ne obstaja v sistemu.</li> </ul>	
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Stranka je ustvarjena v sistemu.</li> </ul>	
Tok aktivnosti:	Akter	Sistem
	1. Direktor začne postopek ustvarjanja nove stranke v sistemu.  2. Direktor vnese matične podatke stranke v sistem: naziv, naslov, datum sklenitve pogodbe idr.	1. a. Sistem prikaže obrazec za vnos podatkov nove stranke.  2. a. Sistem ustvari novo stranko.

se nadaljuje

nadaljevanje

Tok aktivnosti:	Akter	Sistem
	3. Projektni manager v sistem vnese dodatne matične podatke nove stranke: storitve, področja prevajanja, jezike, službeni e-naslov.	3. a. Sistem shrani stranko.
Pogoji izjem:	/	

*Tabela 24: Opis primera uporabe – posodobitev matičnih podatkov stranke*

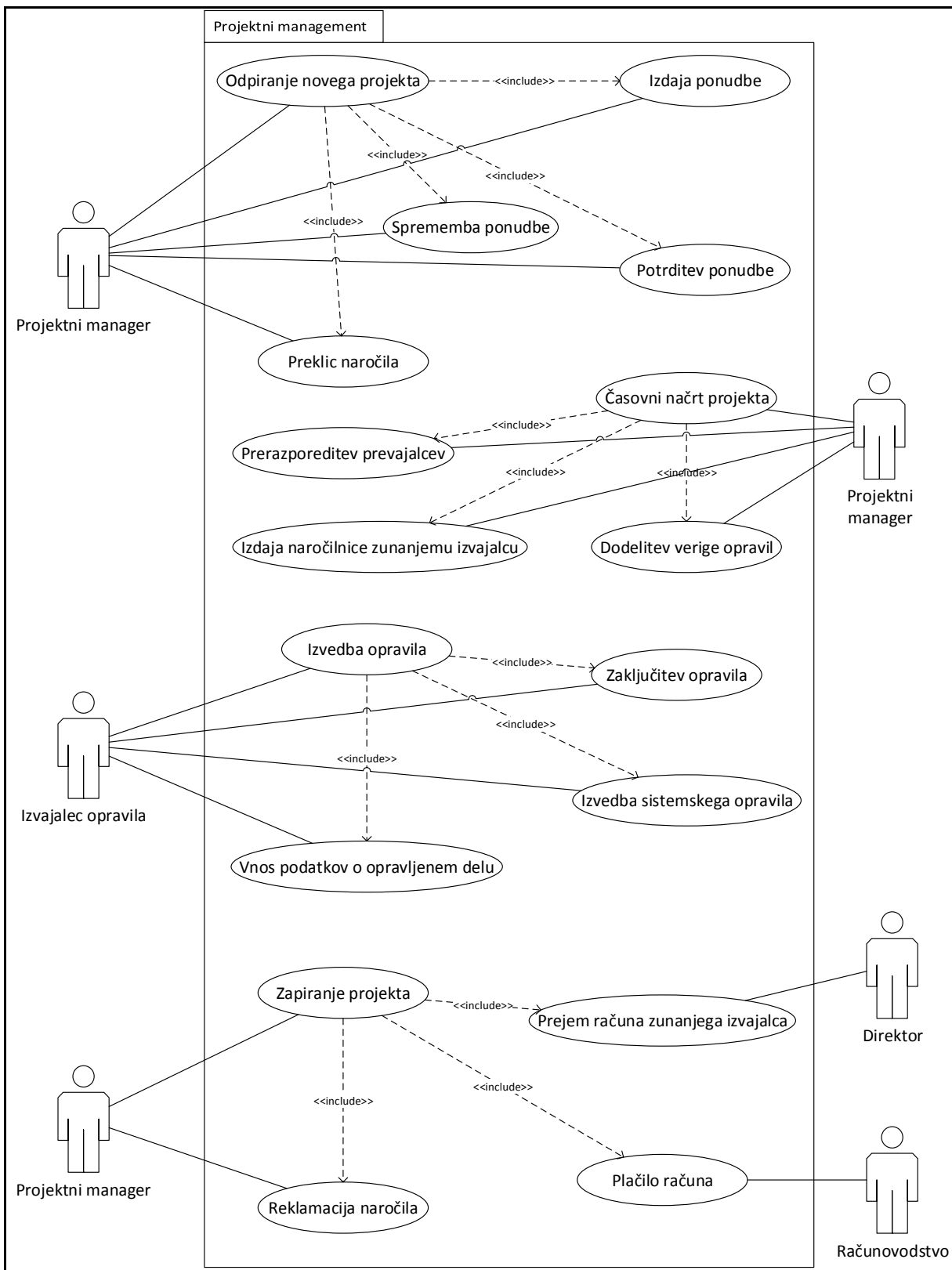
Ime primera uporabe:	Posodobitev matičnih podatkov stranke	
Scenarij:	/	
Prožilni dogodek:	Sprememba matičnih podatkov stranke	
Kratek opis:	Ob spremembi podatkov stranke, ki so lahko posledica spremembe dejavnosti, storitev, kontaktnih oseb stranke ipd., direktor ali projektni manager posodobita matične podatke stranke.	
Akterji:	Direktor, projektni manager	
Povezani primeri uporabe:	/	
Deležniki:	Direktor, projektni manager	
Predpogoji:	<ul style="list-style-type: none"> <li>• Stranka je ustvarjena v sistemu.</li> </ul>	
Zaključni pogoji:	<ul style="list-style-type: none"> <li>• Matični podatki stranke so posodobljeni.</li> </ul>	
Tok aktivnosti:	Akter	Sistem
	1. Direktor ali projektni manager priključeta prikaz matičnih podatkov stranke.	1. a. Sistem prikaže obrazec matičnih podatkov stranke.
	2. Direktor ali projektni manager posodobita matične podatke stranke.	2. a. Sistem shrani posodobljene podatke stranke.
Pogoji izjem:	/	

### 3.5.2 Diagrami primerov uporabe

Na slikah od 21 do 24 so prikazani diagrami primerov uporabe za posamezne module. Diagrami prikazujejo interakcijo med akterji in sistemom.

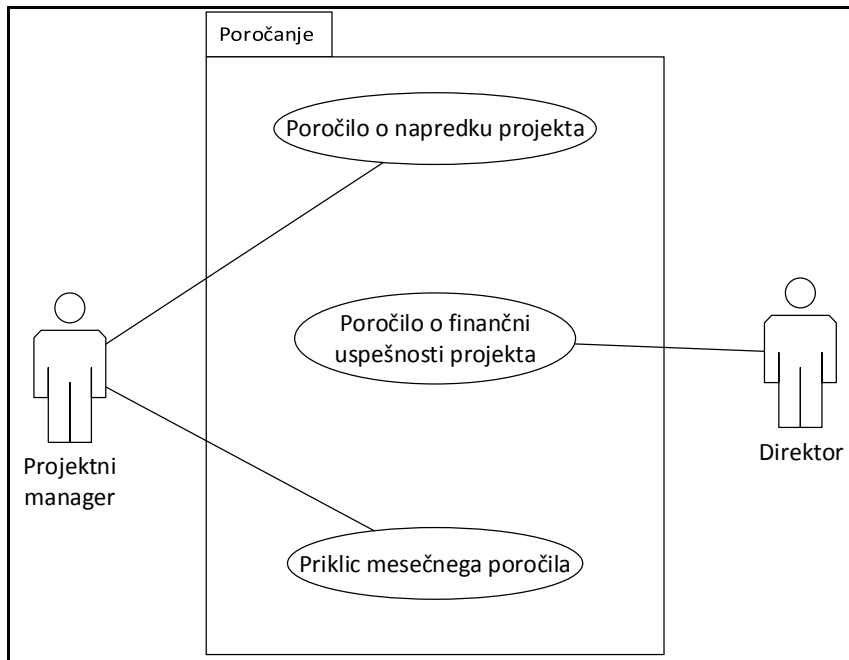
# Projektni management

Slika 21: Diagram primerov uporabe za projektni management



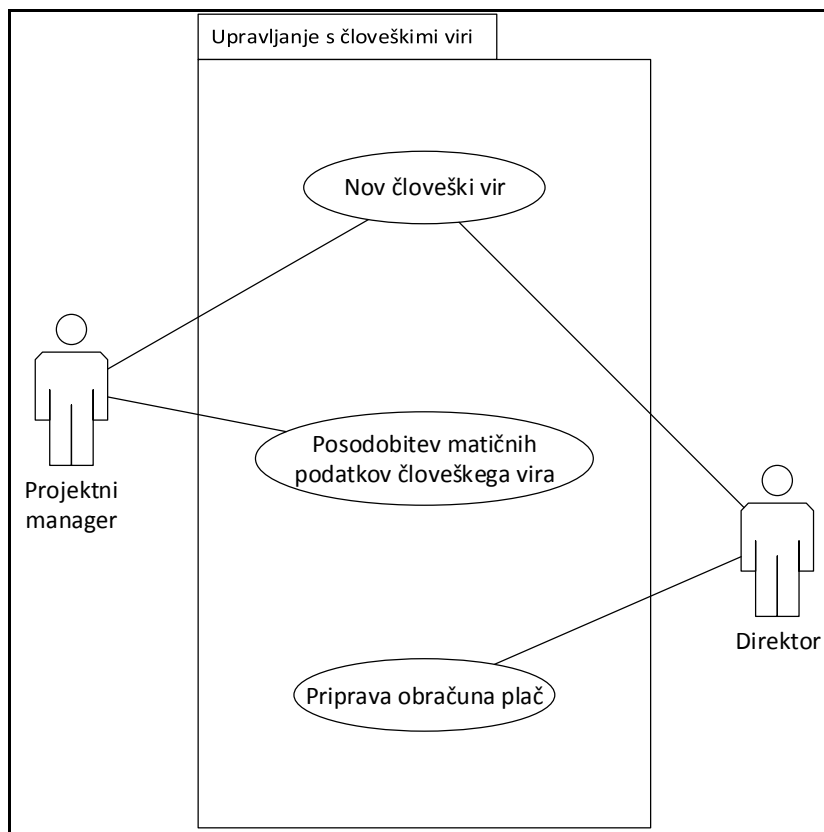
## Poročanje

Slika 22: Diagram primerov uporabe za poročanje



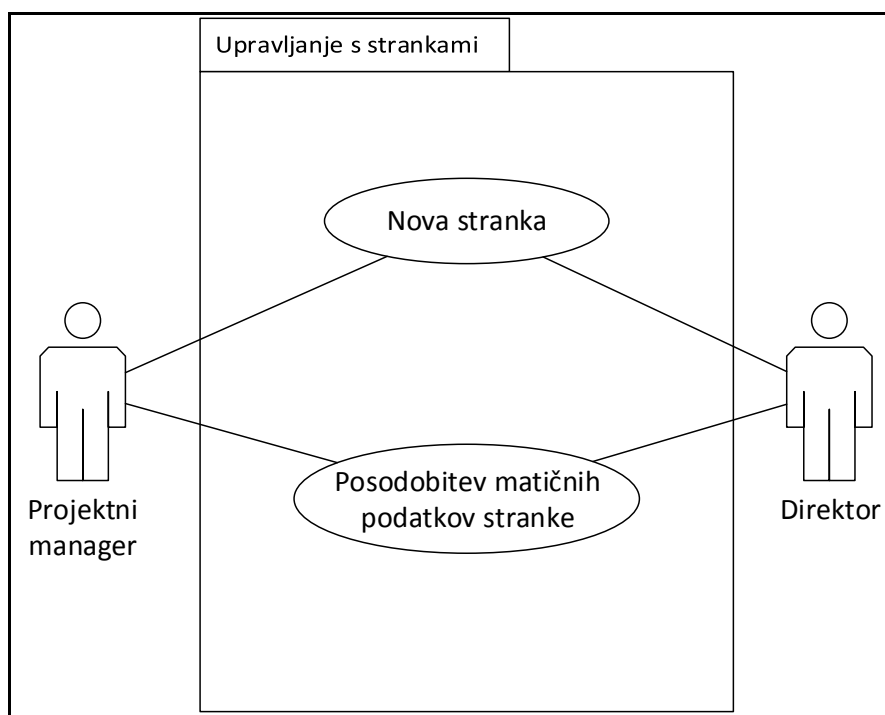
## Upravljanje s človeškimi viri

Slika 23: Diagram primerov uporabe za upravljanje s človeškimi viri



## Upravljanje s strankami

Slika 24: Diagram primerov uporabe za upravljanje s strankami



## 4 MOŽNOSTI IMPLEMENTACIJE

Pri implementaciji vidim naslednje tri možnosti.

### **Prenova informatizacije v okvirih trenutno uporabljene tehnologije.**

Trenutna informatizacija in uporabljena tehnologija imata prednosti v tem, da ju enostavno in hitro prilagajamo našim procesom. Kot sem že omenil, je avtomatizacija zagotovljena s pomočjo makrov VBA. Slaba stran je pomanjkanje integracije s prevajalskimi računalniškimi programi, poročanje pa lahko po želji prilagajamo. Visoka stopnja prilagodljivosti, ki ne zahteva visokih stroškov in jo lahko izvedemo z lastnim znanjem, je tudi razlog, zakaj se še nismo odločili za nakup rešitve »s police«.

Ne glede na to pa bi morali, v kolikor bi se odločili za to možnost, informatizacijo prenoviti na način, da bi za shranjevanje podatkov začeli uporabljati MS Access, za njihovo obdelavo in prikaz pa bi še vedno uporabljali MS Excel.

## **Lasten razvoj rešitve s pomočjo nove tehnologije.**

Če bi želeli razviti lastno rešitev z uporabo druge tehnologije, kot je opredeljena pri prvi možnosti, potem bi morali razvoj dati v zunanje izvajanje, saj sami nimamo niti znanja niti časa. Prednost te možnosti vidim predvsem v treh stvareh: v prilagodljivosti rešitve našim procesom in ne obratno, v uporabi prave baze podatkov in v možnosti trženja take rešitve.

## **Nakup že obstoječe rešitve s prilagajanjem.**

V primeru, da podrobnejša primerjava opravljene analize informacijskih potreb z obstoječimi rešitvami na trgu pokaže, da je ena izmed teh rešitev v veliki meri primerna za opredeljen tip podjetja in da bi bilo treba izvajati prilaganje obstoječe rešitve v majhni meri, potem bi to bila optimalna izbira.

## **SKLEP**

Za cilj magistrskega dela sem si zastavil teoretični model celovite programske rešitve, ki bo prilagojen prevajalski agenciji s stalnim prevajalskim timom, do česar sem prišel z analizo informacijskih potreb. Takšna programska rešitev naj bi bila podobna oziroma enaka tisti, ki je v fazi D življenjskega cikla na S-krivulji po Sargentu. Na podlagi opravljene analize se bomo lažje odločili za eno izmed treh možnosti implementacije iz četrtega poglavja.

Vendar pa po opravljeni analizi informacijskih potreb, po pregledanih poslovnih procesih in obstoječih rešitvah ne predlagam klasične ERP-rešitve. Vseeno pa bi tako rešitev poimenoval celovita ali pa enovita programska rešitev, ki na eni strani pokriva izvajanje vseh ključnih poslovnih procesov v predlaganih modulih, na drugi strani pa sloni na centralni bazi podatkov.

Na prvo mesto je treba postaviti modularni pristop, ki je osnova take rešitve, kar pomeni, da s ključnimi moduli (v našem primeru štirimi) podpremo poslovne procese, hkrati pa se dodatni moduli po potrebi dodajo kasneje. Taka rešitev bi bila primerna tudi za druge podobne agencije, čeprav bi bila optimalno prilagojena tipu agencije, ki je opisan v tej magistrski nalogi.

To pomeni, da je v ospredju prilagodljivo, dinamično in učinkovito načrtovanje (s funkcijo peskovnika za ugotavljanje optimalne izrabe vseh virov podjetja) in izvedba prevajalskih projektov, ki mora reševati probleme omejitev človeških virov na eni in neznano količino dela na drugi strani. Še posebej pomemben vidik je učinkovito prerazporejanje dela, kar pomeni, da se vse spremembe načrta izvedbe naredijo enkrat na enem mestu in se nato samodejno uveljavijo na vseh ustreznih mestih. Vse to bi dosegli z zmožnostmi novega sistema, od katerih je treba izpostaviti poročanje v realnem času, uporabo enotne baze podatkov in tesno integracijo s prevajalskimi računalniškimi programi.

Stopnja integracije programske rešitve s prevajalskimi računalniškimi programi je najvišja možna, saj se vse dodeljevanje in razporejanje izvaja v programski rešitvi, ki nato vsa določila

sporoči računalniškemu prevajalskemu programu. To stopnjo trenutno poskušajo doseči nekateri izmed že omenjenih ponudnikov obstoječih programskih rešitev.

Konkretno od take rešitve pričakujem predvsem manjšo porabo časa pri projektne managementu, točnost pri načrtovanju izvedbe prevajalskih projektov in pri prerazporejanju dela. Dodatna prednost bi se seveda pokazala v obliki bolj kakovostnih podatkov. Poleg tega bi rešitev omogočila ohranitev trenutne ravnalne strukture z enim projektne managerjem tudi pri znatno povečanem obsegu dela in posledično visoko učinkovit enoten prevajalski tim. S tem bi še povečali konkurenčnost na trgu.

Predvsem pa bi taka programska rešitev sledila pravilu, da se mora programska rešitev prilagajati in podpirati poslovne procese, namesto da se poslovni procesi prilagajajo programski rešitvi. Nenazadnje pa bi tako rešitev lahko tudi tržili, v kolikor bi se odločili za lasten razvoj.

Naslednji korak po opravljeni analizi informacijskih potreb je podrobnejša preučitev in izbira ene od treh možnosti implementacije in, odvisno od izbire, priprava načrta implementacije.

## LITERATURA IN VIRI

1. Arlow, J., & Neustadt, I. (2002). *UML and the unified process : practical object-oriented analysis and design*. Boston [etc.]: Addison-Wesley.
2. Blaha, M., & Premerlani, W. (1998). *Object-oriented Modeling and Design for Database Applications*. Upper Saddle River, New Jersey: Prentice-Hall, Inc.
3. Bosilj Vukšić, V., Hernaus, T., & Kovačič, A. (2008). *Upravljanje poslovnim procesima*. Zagreb: školjska knjiga.
4. Casadesus-Masanell, R., & Enric Ricart, J. (2009). From Strategy to Business Models and to Tactics. *Harvard Business School Working Paper*. No. 10-036. Najdeno 27. junija na spletnem naslovu <http://www.hbs.edu/faculty/Publication%20Files/10-036.pdf>.
5. CeFormT d.o.o. (2012a). *Baze prevodov v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
6. CeFormT d.o.o. (2012b). *Informacijski sistemi v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
7. CeFormT d.o.o. (2012c). *Izobraževanje in izpopolnjevanje v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
8. CeFormT d.o.o. (2012č). *Obdelava dokumentov v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
9. CeFormT d.o.o. (2012d). *Področja prevajanja v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
10. CeFormT d.o.o. (2012e). *Področno svetovanje v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
11. CeFormT d.o.o. (2012f). *Prevajalske merske enote v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
12. CeFormT d.o.o. (2012g). *Projektni menedžment v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
13. CeFormT d.o.o. (2012h). *Navodila za terminološko delo v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
14. CeFormT d.o.o. (2012i). *Uporabniški vidik prevajalskih računalniških programov v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
15. CeFormT d.o.o. (2012j). *Vloge v prevajalskem procesu v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
16. CeFormT d.o.o. (2012k). *Zagotavljanje kakovosti v podjetju CeFormT d.o.o.* (interno gradivo). Trzin: CeFormT d.o.o.
17. *Choose your application*. Najdeno 27. avgusta 2013 na spletnem naslovu <http://www.openerp.com>.
18. Grad, J., & Jaklič, J. (1996). *Baze podatkov*. Ljubljana: Ekonomska fakulteta.
19. Gradišar, M., Jaklič, J., & Turk, T. (2007). *Osnove poslovne informatike*. Ljubljana: Ekonomska fakulteta.
20. Harmon, P. (2007). *Business Process Change. A guide for Business Managers and BPM and Six Sigma Professionals*. Burlington: Morgan Kaufmann Publishers.



21. Hoberman, S. (2007). *DATA MODELING MADE SIMPLE* (2<sup>nd</sup> ed.). Bradley Beach: Technics Publications, LLC.
22. *How do translation companies work?* Najdeno 26. junija 2012 na spletnem naslovu <http://www.xtrf.eu>.
23. Jaklič, J. (2002). *Upravljanje in uporaba podatkov*. Ljubljana: Ekonomska fakulteta.
24. Jewett, T. (2002–2006). Database design with UML and SQL. *Design pattern: many-to-many (order entry)*. Najdeno 10. decembra 2013 na spletnem naslovu <http://www.tomjewett.com/dbdesign/dbdesign.php?page=manymany.php>
25. Kovačič, A., & Bosilj Vukšič, V. (2005). *Management poslovnih procesov : prenova in informatizacija poslovanja s praktičnimi primeri*. Ljubljana: Založba GV.
26. Kovačič, A., Groznik, A., & Ribič, M. (2009). *Temelji elektronskega poslovanja*. Ljubljana: Ekonomska fakulteta.
27. Kovačič, A., Jaklič, J., Indihar Štemberger, M., & Groznik, A. (2004). *Prenova in informatizacija poslovanja*. Ljubljana: Ekonomska fakulteta.
28. Larman, C. (2005). *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development*. Upper Saddle River: Prentice Hall PTR.
29. Lipičnik, B. (2002). *Organizacija podjetja*. Ljubljana: Ekonomska fakulteta.
30. Lipovec, F. (1987). *Razvita teorija organizacije*. Maribor: Založba Obzorja.
31. M 74.300 - Prevajanje in tolmačenje. (b.l.) V *Bonitete.si – napredno iskanje*. Najdeno 1. septembra 2013 na spletnem naslovu <http://www.bonitete.si/CreditReportsPrimus/Kontrola/Iskalnik.aspx?Lang=sl-SI&Mode=SI&App=SI>.
32. Memsource. (2013a). *About us*. Najdeno 23. decembra 2013 na spletnem naslovu <http://www.memsource.com/about-us/>.
33. Memsource. (2013b). *XTRF and MemSource Announce API-driven Integration*. Najdeno 23. decembra 2013 na spletnem naslovu <http://blog.memsource.com/memsource-and-xtrf-announce-api-driven-integration/>.
34. Močnik, D. (2010). Poslovni model in informacijski sistemi. *Elektrotehniški vestnik*, letnik 77, številka 2/3, str. 143-148. URN:NBN:SI:DOC-AAK3ON5E. Najdeno 27. junija 2014 na spletnem naslovu <http://www.dlib.si>.
35. *Object Data Management Group*. Najdeno 21. avgusta 2013 na spletnem naslovu <http://www.odbms.org/odmg-standard/>
36. Orna, E. (2004). *Information strategy in practice*. Aldershot; Burlington (VT): Gower.
37. Osterwalder, A., & Pigneur, Y. (2010). *Business model generation : a handbook for visionaries, game changers, and challengers*. Hoboken, New Jersey: John Wiley & Sons, Inc.
38. *Poslanstvo*. Najdeno 15. avgusta 2013 na spletnem naslovu <http://www.ceformt.com/glavna.asp?f=poslanstvo&lng=slo>
39. Post, G., V., & Anderson, D., L. (2003). *Management information systems: solving business problems with information technology*. Boston: McGraw-Hill/Irwin.
40. Pučko, D. (2008). *Strateški management 1*. Ljubljana: Ekonomska fakulteta.
41. Pučko, D. (2009). *Strateški management 2*. Ljubljana: Ekonomska fakulteta.
42. Rozman, R. (2000). *Analiza in oblikovanje organizacije*. Ljubljana: Ekonomska fakulteta.

43. Rozman, R., & Stare, A. (2008). *Projektni management ali ravnateljstvo projekta*. Ljubljana: Ekonomska fakulteta.
44. Sargent, B. (2012). Translation Management Systems: build or buy? *Tcworld*, str. 18-20.
45. Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2009). *System Analysis and Design in a Changing World* (5<sup>th</sup> ed.). Boston: Course Technology.
46. Seddon, P., B., & Lewis, G.,P. (2003). *Strategy and Business Models: What's the Difference?* Najdeno 27. junija 2014 na spletnem naslovu <http://www.pacis-net.org/file/2003/papers/e-business/219.pdf>.
47. Shelly, G., B., Cashman, T., J., & Rosenblatt, H., J. (2006). *Systems analysis and design*. Boston (MA): Thomson Course Technology.
48. Sherry, K. J. (2011). *Insight into business processes : a guide to understanding and designing business processes*. Germany: Admaks.
49. Slovenski inštitut za standardizacijo. (2007). *Slovenski standard SIST EN 15038:2007, Prevajalske storitve - Zahteve za storitve*. Ljubljana: Slovenski inštitut za standardizacijo.
50. Strimbei, C. (2005). Object Oriented Principles in Information Systems Alignment with Enterprise Modelling. *Economy Informatics*. Najdeno 27. junija 2014 na spletnem naslovu [http://www.economyinformatics.ase.ro/content/EN5/Strimbei\\_Catalin\\_Full\\_Paper.pdf](http://www.economyinformatics.ase.ro/content/EN5/Strimbei_Catalin_Full_Paper.pdf).
51. Turban, E., Rainer, R. Kelly, Jr., & Potter, R. E. (2001). *Introduction to information technology*. New York: John Wiley & Sons, Inc.
52. Valacich, J., S., George, J., F., & Hoffer, J., A. (2009). *Essentials of systems analysis and design*. Upper Saddle River (N.J.): Perason Education International.
53. van der Hoeven, H. (2009). *ERP and Business Processes*. Vlijmen, Coral Springs (FL): Llumina Press.
54. *Vizija*. Najdeno 15. avgusta 2013 na spletnem naslovu <http://www.ceformt.com/glavna.asp?f=vizija&lng=slo>.
55. Wallace, T. F., & Kremzar, M. H. (2001). *ERP : making it happen : the implementers' guide to success with enterprise resource planning*. New York: John Wiley & Sons, Inc.
56. *What is ERP?* Najdeno 15. decembra 2012 na spletnem naslovu <http://www.erp.com>.
57. Wood, B. (2010, 31. maj). ERP vs. ERP II vs. ERP III Future Enterprise Applications. Najdeno 16. avgusta na spletnem naslovu <http://www.r3now.com/erp-vs-erp-ii-vs-erp-iii-future-enterprise-applications/>.