

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

**ANALIZA INFORMACIJSKEGA SISTEMA ZA JAVNA NAROČILA
PO EVIDENČNEM POSTOPKU**

Ljubljana, april 2016

PAVEL KOŠIR

IZJAVA O AVTORSTVU

Spodaj podpisani Pavel Košir, študent Ekonomske fakultete Univerze v Ljubljani, izjavljam, da sem avtor magistrskega dela z naslovom Analiza informacijskega sistema za javna naročila po evidenčnem postopku, pripravljenega v sodelovanju s svetovalcem prof. dr. Mirkom Gradišarjem.

Izrecno izjavljam, da v skladu z določili Zakona o avtorski in sorodnih pravicah (Ur. l. RS, št. 21/1995 s spremembami) **ne dovolim** objave magistrskega dela 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, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam v magistrskem delu, citirana oziroma navedena v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, in
 - pridobil 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;
- se zavedam, da je plagiatorstvo – 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 magistrskega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu relevantnim pravilnikom.

V Ljubljani, dne _____

Podpis avtorja: _____

KAZALO

UVOD	1
1 TEORETIČNE OSNOVE JAVNEGA NAROČANJA, UPORABLJENIH PROGRAMSKIH ORODIJ IN METODOLOGIJ ZA RAZVOJ PROGRAMSKE REŠITVE.....	3
1.1 Splošno o javnih naročilih	3
1.1.1 Postopek oddaje javnega naročila male vrednosti	4
1.1.2 Postopek oddaje javnega naročila pod mejo NMV	5
1.2 Metodologije razvoja programske rešitve	9
1.2.1 Zaporedni ali linearni model	10
1.2.2 Prototipni model.....	14
1.2.3 Spiralni model.....	16
1.2.4 Inkrementalni model	18
1.3 Programska orodja.....	20
1.3.1 Visual Studio 2012.....	20
1.3.2 Spletni strežnik.....	23
1.3.3 Strežnik SQL Server 2012	27
2 ANALIZA OBSTOJEČEGA STANJA	28
2.1 Predstavitev izbrane organizacije	28
2.2 Analiza temeljnih poslovnih procesov	29
2.3 Analiza informacijskega sistema	32
2.3.1 Postopek evidentiranja in kroženja dokumentov	32
2.4 Podatkovni model.....	33
2.5 Programska, komunikacijska in strojna oprema.....	36
2.6 Kritična analiza starega sistema	37
2.6.1 Uporabniki	37
2.6.2 Povezljivost.....	38
2.6.3 Tehnična zastarelost.....	38
2.6.4 Revizijska sled	38
2.6.5 Podatkovni model	38
2.6.6 Analiza podatkov	38
3 IZVEDBA INFORMACIJSKEGA SISTEMA.....	39
3.1 Sodelujoči pri razvoju informacijskega sistema.....	39
3.2 Terminski načrt razvoja informacijskega sistema	40
3.2.1 Časovni plan.....	40
3.2.2 Finančni načrt.....	41
3.3 Izdelava funkcijske specifikacije.....	45
3.3.1 Splošne funkcionalnosti	45
3.3.2 Specifične funkcionalne rešitve	52

3.4	Izdelava podatkovnega modela	66
3.4.1	Podatkovna analiza – normalizacija	67
3.4.2	Proučitev podatkovnega modela	69
3.4.3	Prepis starih podatkov v novi podatkovni model.....	71
3.5	Izdelava programske rešitve	71
3.5.1	Vzpostavitev testnega okolja	71
3.5.2	Pravice uporabnikov	72
3.5.3	Urejanje šifrantov	72
3.5.4	Urejanje registra strank	73
3.5.5	Vnosne maske	73
3.5.6	Povezava z dokumentnim sistemom.....	75
3.5.7	Analitične poizvedbe in poročila	75
3.6	Testiranje rešitve	75
3.7	Izdelava dokumentacije.....	76
3.8	Usposabljanje uporabnikov	77
3.9	Prehod v produkcijo	78
3.10	Vzdrževanje programske rešitve	78
	SKLEP.....	79
	LITERATURA IN VIRI.....	81
	PRILOGE	

KAZALO TABEL

Tabela 1:	Časovni plan za izgradnjo informacijskega sistema	41
Tabela 2:	Faze izdelave programske rešitve	41
Tabela 3:	Strošek razvoja programske rešitve v primeru notranjega razvoja	43
Tabela 4:	Referenčni cenik storitev s področja informacijskih storitev.....	44
Tabela 5:	Vnos oziroma zajem podatkov	45
Tabela 6:	Faze v postopku naročanja	47
Tabela 7:	Uporabniki menijev programske rešitve	55
Tabela 8:	Podatki povezave »moji predlogi«.....	56
Tabela 9:	Ostali podatki povezave »moji nepotrjeni predlogi«	59
Tabela 10:	Ostali podatki povezave »moje nepotrjene naročilnice«.....	59
Tabela 11:	Ostali podatki povezave »moji preklicani predlogi«.....	59
Tabela 12:	Ostali podatki povezave »moji nezaključeni postopki«.....	60
Tabela 13:	Ostali podatki povezave »moji zaključeni postopki«.....	60
Tabela 14:	Ostali podatki povezave »predlogi v pripravi za potrditev«	60
Tabela 15:	Ostali podatki povezave »naročilnice v pripravi za potrditev«.....	61
Tabela 16:	Ostali podatki povezave »pogodbe v pripravi za potrditev«.....	63
Tabela 17:	Ostali podatki povezave »dokument pogodba«	63

Tabela 18: Ostali podatki povezave »knjiženje«.....	64
Tabela 19: Ostali podatki povezave »nezaključena naročila«.....	64
Tabela 20: Podatkovna analiza – normalizacija	67
Tabela 21: Proučitev podatkovnega modela.....	69
Tabela 22: Izdelana dokumentacija	77

KAZALO SLIK

Slika 1: Zaporedni model razvoja.....	11
Slika 2: V-model razvoja.....	13
Slika 3: Prototipni model razvoja	14
Slika 4: Spiralni model razvoja	17
Slika 5: Inkrementalni model razvoja.....	19
Slika 6: Notranja organizacija izbrane organizacije	29
Slika 7: Poenostavljena shema evidenčnega postopka oddaje naročila	31
Slika 8: Materialna in informacijska pot dokumentov	33
Slika 9: Podatkovni model.....	34
Slika 10: Strežniška infrastruktura izbrane organizacije	36
Slika 11: Glavni meni programske rešitve	54
Slika 12: Povezava »priprava naročil«	55
Slika 13: Povezava »moji predlogi«.....	56
Slika 14: Dokument »predlog za izdajo naročila«.....	57
Slika 15: Vnosna maska »predmet naročil«	58
Slika 16: Vnosna maska »dobavitelj«	58
Slika 17: Vnosna maska »urejanje zneska naročila«.....	61
Slika 18: Vnosna maska »številka naročilnice«	62
Slika 19: Dokument »naročilnica«	62
Slika 20: Vnosna maska »račun«.....	64
Slika 21: Vnosna maska »urejanje računa«.....	65
Slika 22: Podatkovni model programske rešitve	70
Slika 23: Vmesnik za urejanje pravice uporabnikov	72
Slika 24: Vnosna maska »predlog naročila«	74

UVOD

Predstavitev problema. Živimo v času, ko smo priča neprestanim in hitrim spremembam tako v organizacijah kot tudi v okolju, kjer se organizacije nahajajo, in te spremembe so vse bolj nepredvidljive, po drugi strani pa dinamika sprememb okolja dviguje raven potreb po medsebojni konkurenčnosti organizacij ter tudi pospešitvi pretoka podatkov in informacij znotraj organizacije ter med organizacijo in njenim okoljem (Kovačič & Vintar, 1994, str. 20).

Konkurenčnost organizacije je v veliki meri odvisna tudi od programskih rešitev, ki jih uporablja. Organizacija se lahko odloči za razvoj lastne programske rešitve ali pa najame zunanje izvajalce. Smernice za razvoj programskih rešitev so se v zadnjih letih zelo spremenile. Organizacije, kot so javne službe, javni zavodi ipd., so imele v preteklosti veliko posluha za razvoj lastnih informacijskih rešitev v okviru notranjega razvoja. Z leti se je razvoj informacijskih rešitev začel preusmerjati na zunanje izvajalce.

Pristop k razvoju informacijskega sistema je največkrat posledica notranjih zahtev po izboljšanju poslovanja ter vpeljavi notranje urejenosti v podjetje. Zahteve, ki jih pri vsakdanjem delu izražajo izvedbeni delavci, so le redkokdaj zadosten razlog za pristop k projektu prenove oz. vzpostavitve informacijskega sistema. Po navadi poda zahtevo po boljši informatizaciji vrhni management, ki ne dobiva pravočasnih in ustreznih informacij, potrebnih za svoje odločanje (Zornada, 2002, str. 224).

Pri lastnem razvoju programskih rešitev so prednosti predvsem v tem, da sodelujoči v projektni skupini zelo dobro poznajo vsebine, saj so zaposleni v organizaciji. Komunikacija med zaposlenimi je boljša in hitrejša, kar pa lahko predstavlja tudi nižje stroške razvoja. Pri vzdrževanju programskih rešitev so odzivni časi za odpravo napak lahko bistveno krajši. Slabosti se lahko izkažejo v pristranskosti in slabšem poznavanju najnovejših tehnologij na področju razvoja informacijskih sistemov. Pri zunanjem razvoju programskih rešitev so prednosti lahko predvsem v nepristranskosti in boljšem poznavanju najnovejših tehnologij razvoja. Slabosti se največkrat kažejo predvsem v nerazumevanju vsebine organizacije, posledica pa je daljši čas razvoja, lahko pa tudi bistveno višji stroški razvoja.

Informacijska rešitev za vodenje evidenc o javnih naročilih malih vrednosti po evidenčnem postopku ne obstaja kot paket, ki bi ga bilo možno kupiti na trgu. Poraja pa se vprašanje, ali jo je smiselno razviti v okviru notranjega razvoja ali pa jo naročiti pri zunanjem naročniku. Sam sem se v zadnjih letih pri svojem nepoklicnem udejstvovanju kot predsednik krajevne skupnosti na raznih področjih srečeval z različnimi organizacijami (občine, zavodi idr.), ki so izkazovale potrebe po razvoju informacijske rešitve za vodenje evidenc o javnih naročilih male vrednosti po evidenčnem postopku, zato sem se odločil za razvoj nove programske rešitve. Razvoj opisane informacijske rešitve rešuje naslednje težave:

- postopek oddaje javnega naročila ni v celoti informatiziran,
- uporabnikom ni omogočena izdelava dokumentov, ki nastanejo v postopku naročanja,
- uporabnik nima vsebinskega pregleda nad postopkom javnega naročanja.

Namen dela. Namen magistrskega dela je analizirati in razviti informacijski sistem za vodenje evidenc o javnih naročilih male vrednosti po evidenčnem postopku. Nova rešitev bo omogočila boljše poslovanje, nižje stroške in hitrejše izvajanje poslovnih procesov.

Cilj dela. Osnovni cilj magistrskega dela je na osnovi podrobne analize obstoječega stanja in veljavne pravne podlage o javnih naročilih male vrednosti analizirati in razviti nov informacijski sistem.

Podrobnejši cilji magistrskega dela so:

- analizirati obstoječe stanje (analiza poslovnih procesov in analiza informacijskega sistema),
- pridobiti osnovna znanja s področja javnih naročil malih vrednosti in metodologij razvoja informacijskih sistemov,
- pridobiti zahteve uporabnikov in izvesti sistemsko analizo informacijskega sistema,
- glede na postavljene zahteve razviti informacijski sistem,
- ugotoviti možnosti nadaljnjega razvoja.

Programska rešitev bo poenostavila postopek javnega naročanja po evidenčnem postopku. Omogočila bo pripravo, pregled in urejanje dokumentov, ki nastopajo v postopku javnega naročanja, in vsebinski nadzor nad postopki.

Metode dela. Pri izdelavi magistrskega dela sem uporabil znanja, pridobljena na dodiplomskem in podiplomskem študiju, in večletne izkušnje na področju programiranja programskih rešitev. Z vsebino obravnavane problematike sem se seznanil s študijem domače in tuje literature.

Za seznanjanje z vsebino obstoječega stanja na področju javnih naročil in prenove načrtovanja informacijskih sistemov sem uporabil metodo analize obstoječega stanja, metodo analize zelenega stanja in metodo sinteze nove rešitve.

Magistrsko delo je sestavljeno iz teoretičnega in praktičnega dela. V okviru Zakona o javnem naročanju (Ur. l. RS, št. 128/2006, 16/08, 19/10, 18/11, 43/12, 90/12, 12/13, 19/14, 90/14; v nadaljevanju ZJN-2) je predstavljen postopek javnega naročanja malih vrednosti po evidenčnem postopku. Opisana so osnovna orodja, ki so pomembna za razvoj in delovanje prenovljenega informacijskega sistema. Nadaljujem s predstavitev štirih metodologij, ki se uporabljajo pri razvoju informacijskih sistemov. Pri izgradnji in načrtovanju informacijskih sistemov lahko uporabimo več metodologij razvoja. Nekateri izmed modelov so: linearni,

prototipni, spiralni in inkrementalni. Zelo znan je tudi objektivi pristop k razvoju. Izbrana je bila metoda, ki je najbolj ustrezala konkretnemu primeru razvoja programske rešitve.

Pri analizi se najprej posvetim analizi osnovnega procesa javnega naročanja malih vrednosti po evidenčnem postopku. Na ta način lahko proučim vse vidike obravnavanega sistema. V tej fazi je pomembno proučiti organizacijsko strukturo za ugotavljanje vseh informacijskih potreb in povezav sistema z njegovo okolico: natančno opredelim tok dokumentov, informacijske tokove, podatkovni model in vse postopke, s katerimi se podatki obdelujejo. Področje proučim tudi z zaposlenimi in s strokovnjaki, ki delajo na proučevanem področju. Na osnovi vseh ugotovitev opredelim glavne pomanjkljivosti starega informacijskega sistema.

V praktičnem delu predstavim postopek razvoja in izdelave lastne programske rešitve za vodenje evidenc o javnih naročilih malih vrednosti po evidenčnem postopku. Ugotovitve, ki izhajajo iz teoretičnega in praktičnega dela, so zapisane v zaključku magistrskega dela.

1 TEORETIČNE OSNOVE JAVNEGA NAROČANJA, UPORABLJENIH PROGRAMSKIH ORODIJ IN METODOLOGIJ ZA RAZVOJ PROGRAMSKE REŠITVE

1.1 Splošno o javnih naročilih

Javno naročilo je skupek vseh dejanj, ki jih mora naročnik opraviti s ciljem nabave blaga, naročila storitve ali gradenj po Zakonu o javnem naročanju ob upoštevanju temeljnih načel javnega naročanja (Primec, 2008).

Naročnik je dolžan nabavljati blago, naročati storitve in izvedbe gradnje v skladu s pravili, ki so določena v ZJN-1 (Ur. l. RS, št. 39/2000, 102/2000, 2/04; v nadaljevanju ZJN-1) in na njegovi osnovi izdanih predpisih. Javno naročilo je dosti širši pojem od javnega razpisa, ki je sicer najpomembnejši del javnega naročila. Javno naročilo se prične, ko naročnik ugotovi, da potrebuje neko blago, izvedbo storitve ali gradnje, konča pa se šele, ko potečejo garancije po pogodbah, ki so sklenjene prek sistema javnega naročanja. Posebna, nekoliko milejša pravila javnega naročanja veljajo za javna naročila male vrednosti, pri katerih javnega razpisa po navadi sploh ni (Javornik, Železnik, & Čerič, 2006, str. 4).

Nusdorfer (2004, str. 4) navaja, da je postopek javnega naročanja sklop dejanj vseh udeležencev, naročnika in ponudnika od trenutka, ko naročnik na osnovi sklepa pristojnega organa ugotovi potrebo po neki dobrini (blagu, storitvi, gradbenem delu), pa vse do poteka garancijskega roka za nabavljeno blago ali opravljeno storitev.

Po 3. členu ZJN-2 so naročniki:

- organi Republike Slovenije in samoupravnih lokalnih skupnosti,
- javni skladi, javne agencije, javni zavodi,
- javni gospodarski zavodi in
- druge osebe javnega prava.

Po 24. členu ZJN-2 naročnik izvede javno naročanje po enem izmed naslednjih postopkov:

- odprti postopek,
- postopek s predhodnim ugotavljanjem sposobnosti,
- konkurenčni dialog,
- postopek s pogajanjem brez predhodne objave,
- postopek s pogajanjem po predhodni objavi,
- postopek zbiranja ponudb po predhodni objavi in
- postopek oddaje naročila male vrednosti.

1.1.1 Postopek oddaje javnega naročila male vrednosti

Naročilo male vrednosti je javno naročilo, ki po Zakonu o spremembah in dopolnitvah Zakona o javnem naročanju (Ur. l. RS, št. 19/2014) ne presega določene vrednosti:

- pri naročanju blaga ali storitev je ta vrednost enaka ali višja od 20.000 EUR in nižja od 134.000 EUR brez davka na dodano vrednost (v nadaljevanju DDV),
- pri naročanju gradenj je ta vrednost enaka ali višja od 40.000 EUR in nižja od 274.000 EUR brez DDV.

Novi zakon ZJN-2 je stopil v veljavo 16. 4. 2014 in ima glede na predhodni zakon ZJN-2 spremembe predvsem v mejnih vrednosti javnih naročil. To pomeni, da pretežna večina javnih naročil poteka po poenostavljenem postopku javnega naročanja. Bistvene značilnosti postopka oddaje javnega naročila male vrednosti po ZJN-2 so (Kosmač, 2010, str.17–19):

- naročnik mora na portalu javnih naročil objaviti obvestilo o naročilu in oddaji naročila male vrednosti;
- priprava razpisne dokumentacije ni obvezna;
- rok za oddajo ponudb določi naročnik in mora biti takšen, da ponudnikom omogoči pripravo popolnih ponudb;
- obvezno je javno odpiranje ponudb;
- popolnost ponudbe se preverja zgolj pri najugodnejši ponudbi;
- v postopek se lahko vključi pogajanja;
- naročnik lahko omeji sodelovanje ponudnikov v postopku oddaje naročila male vrednosti zgolj na osnovi vnaprej opredeljenih pogojev;

- ponudnik dokazuje izpolnjevanje pogojev s pisno izjavo, ki je naročniku v skladu z 77. členom¹ tega zakona ni treba preverjati, razen kadar dvomi v resničnost;
- v primeru ustavitve postopka oddaje naročila pred potekom roka za oddajo ponudb mora naročnik to objaviti na portalu javnih naročil in vrniti neodprte ponudbe ponudnikom;
- v primeru zavrnitve vseh ponudb je treba navesti razloge in odločitev o novem postopku ter o tem obvestiti ponudnike in objaviti na portalu javnih naročil. Nadzornega organa se o tem ne obvešča;
- za izbiro ponudbe se upoštevajo merila iz 48. člena² ZJN-2;
- v primeru neobičajno nizke ponudbe se upoštevajo merila iz 49. člena³ ZJN-2;
- naročniki morajo enkrat letno (do 28. februarja) na portal JN posredovati podatke o vrsti predmeta, skupni vrednosti in številu oddanih javnih naročil v preteklem letu (ločeno za naročila male vrednosti (v nadaljevanju NMV) in naročila pod mejo NMV);
- dokumentacija o NMV, ki jo je dolžan voditi naročnik, obsega:
 - sklep o začetku postopka, če je to primerno,
 - povabilo k oddaji ponudbe,
 - razpisno dokumentacijo, če je to primerno,
 - predložene ponudbe,
 - odločitev o oddaji naročila,
 - druge dokumente glede na način izvedbe javnega naročila.

1.1.2 Postopek oddaje javnega naročila pod mejo NMV

Postopek oddaje naročil pod mejnimi vrednostmi je prepuščen naročnikom. Vsak naročnik si mora postopek oddaje javnega naročila male vrednosti pod mejnimi vrednostmi določiti v notranjem aktu, v katerem je opredeljen postopek oddaje takih naročil.

V nadaljevanju predstavljam splošni pravilnik o oddaji naročila male vrednosti po evidenčnem postopku, ki sem ga določil glede na splošne značilnosti ZJN-2 in opisuje

¹ 77 člen ZJN-2 določa: (1) Naročnik mora najkasneje pred sklenitvijo pogodbe o oddaji naročila, lahko pa že prej, preveriti obstoj in vsebino podatkov iz izbrane ponudbe oziroma drugih navedb iz ponudbe. (2) Če naročnik v postopku javnega naročanja ugotovi, da je posamezni ponudnik v ponudbi predložil neresnična in zavajajoča dokazila, je dolžan ponudnika izločiti iz nadaljnjega postopka javnega naročanja. Naročnik je dolžan o tem obvestiti ministrstvo, pristojno za finance, ki vodi evidenco o ponudnikih z negativnimi referencami. Takega ponudnika oziroma podizvajalca se izloči iz postopkov javnega naročanja za obdobje treh let, če je predmet naročila blago ali storitev, oziroma za obdobje petih let, če je predmet naročila gradnja. Ministrstvo, pristojno za finance, objavi seznam ponudnikov z negativnimi referencami na svoji spletni strani.

² Nekatera pomembna določila 48 člena ZJN-2 so: (1) Naročnik lahko odda naročilo na osnovi ekonomsko najugodnejše ponudbe ob uporabi različnih meril v povezavi s predmetom naročila ali na osnovi najnižje cene. (2) Merila v obvestilu o javnem naročilu in razpisni dokumentaciji morajo biti enaka. (3) Merila za oddajo naročila morajo biti ovrednotena, ne smejo biti diskriminatorna in morajo biti smiselno povezana z vsebino javnega naročila. (4) V primeru dveh ali več ekonomsko najugodnejših ponudb se izbere najugodnejšo ponudbo na osnovi vnaprej določenih socialnih elementov, ki morajo biti določeni v razpisni dokumentaciji.

³ Nekatera pomembna določila 49 člena ZJN-2 so: (1) Naročnik mora preveriti, ali je ponudba neobičajno nizka in ali obstaja dvom o izpolnitvi naročila. (2) Preden naročnik izloči neobičajno nizko ponudbo, mora od ponudnika zahtevati podrobne podatke in utemeljitev o elementih ponudbe (ekonomičnost storitve, tehnične rešitve, izvirnost, skladnost s predpisi). (3) Šele na osnovi preveritev vseh podatkov in elementov ponudbe ob posvetovanju s ponudnikom lahko naročnik ponudbo izloči, če ugotovi, da je neobičajno nizka.

uporabo postopka za evidenčno naročilo, katerega glavne značilnosti so skupne za vse naročnike, ki morajo spoštovati določbe ZJN-2.

Uporaba postopka za evidenčno naročilo. Javna naročila po evidenčnem postopku, ki so predmet urejanja, so naročila, katerih vrednost brez DDV je:

- v primeru naročanja blaga in storitev nižja od 20.000 EUR,
- v primeru naročanja gradenj nižja od 40.000 EUR.

Za oddajo evidenčnih naročil po teh navodilih se ne uporabljajo določbe ZJN-2. Postopek se vodi v skladu s temi navodili in temeljnimi načeli javnega naročanja. Pri oddaji naročil je naročnik dolžan zagotoviti gospodarno in učinkovito porabo javnih sredstev in uspešno izvedbo ciljev, ki so povezani s porabo javnih sredstev, skladno s predpisi, ki urejajo porabo proračunskih in drugih javnih sredstev. Pri oddaji naročil mora naročnik ravnati tako, da je poraba sredstev najbolj gospodarna glede na namen, predmet in vrednost naročila. V primeru, da se kadar koli v postopku ugotovi, da vrednost naročila presega mejno vrednost iz tega navodila, mora naročnik naročanje prekiniti in začeti nov postopek skladno z določbami ZJN-2. Odgovorna oseba naročnika sme odrediti oddajo naročila, če je naročilo predvideno v letnem programu dela oziroma je v okviru finančnega načrta organizacije. Vsako evidenčno naročilo (na osnovi naročilnice in/ali pogodbe) mora pred izdajo naročilnice in pred podpisom sklepa o izbiri dobavitelja/izvajalca potrditi vodja organizacijske enote in zakoniti zastopnik organizacije.

Način oddaje naročila po evidenčnem postopku. Glede na ocenjeno vrednost uporablja naročnik enega izmed naslednjih načinov oddaje naročila:

- **naročila, ki se oddajo z naročilnico:** do vrednosti 5.000 EUR brez vključenega DDV;
- **naročila, ki se oddajo s pogodbo:** kot naročila za blago in storitve, katerih vrednost je višja od 5.000 EUR brez vključenega DDV in nižja od 20.000 EUR brez vključenega DDV, ter naročila za gradnje, katerih vrednost je višja od 5.000 EUR brez vključenega DDV in nižja od 40.000 EUR brez vključenega DDV.

Istovrstna naročila blaga, storitev ali gradenj enemu ali več izvajalcem oziroma dobaviteljem se ne smejo oddati v skladu s temi navodili, če v koledarskem letu skupni obseg teh naročil preseže mejne vrednosti javnega naročila male vrednosti.

Merilo za izbor. V postopku oddaje naročila se cena oziroma kakovost predmeta naročila preverjata s poizvedovanjem oziroma zbiranjem ponudb, lahko tudi z raziskavo tržišča ali na drug ustrezen način.

Merilo za izbor ponudbe je najnižja cena ali ekonomsko najugodnejša ponudba. Merilo najnižja cena pomeni, da je ob izpolnjevanju vseh zahtevanih pogojev edino merilo za oddajo naročila.

Ekonomsko najugodnejša ponudba je ponudba, ki ustreza različnim merilom, odvisno od predmeta naročila. Merila so lahko npr. datum dobave ali zaključka del, tekoči stroški, stroškovna učinkovitost, kakovost, estetske in funkcionalne lastnosti, tehnične prednosti, poprodajne storitve in tehnična pomoč, garancijska doba, obveznosti v zvezi z rezervnimi deli, pogarancijsko vzdrževanje, ugodnejši plačilni pogoji, reference ponudnika, cena in podobno.

Pri izboru ponudbe z uporabo merila ekonomsko najugodnejše ponudbe je treba navesti, opisati in ovrednotiti merila po vrstnem redu od najpomembnejšega do najmanj pomembnega.

Odgovornost za pravilno izvedbo postopka. Za določitev predmeta naročila in njegove ocenjene vrednosti, za izbiro in izvedbo naročila ter za pravilnost izvedbe je odgovoren vodja organizacijske enote, ki je naročnik blaga/storitve. Za pridobivanje ponudb, preverjanje cene in kakovosti ter za oddajo naročila skrbi zaposleni, v čigar delovno področje spada naročilo, razen če je s pisno odredbo za izvedbo naročila z naročilnico ali s potrjenim sklepom o oddaji naročila določen drug zaposleni (v nadaljevanju: vodja naročila). Vodja naročila je tudi skrbnik pogodbe oziroma skrbi za izvedbo oddanega naročila. Vsak vodja naročila je odgovoren, da se sredstva na naročilnici/pogodbi ne presežejo – odgovoren je za skladnost prejetega računa z vsebino (količina, kakovost idr.) prevzetega blaga/storitve in ponudbeno vrednostjo. Odgovorna oseba za izvajanje javnega naročanja v organizacijski enoti za finančno področje je odgovorna, da so vsa evidenčna naročila pravilno evidentirana in da se o njih poroča.

Pridobivanje ponudb. Vodja naročila mora pridobiti najmanj tri ponudbe. Pri oddaji naročila mora vodja naročila zagotoviti, da pogodbeni cena ni višja od primerljive cene na tržišču, in skrbno preveriti kakovost predmeta, ki je predmet naročila. Čim več ponudb je treba pridobiti zlasti pri naročilih v ocenjeni vrednosti nad 5.000 EUR brez DDV.

Pridobivanje več ponudb ni potrebno, kadar glede na naravo naročila in njegovo vrednost to ne bi bilo smotno oziroma kadar to zaradi časovne stiske ne bi bilo možno in bi lahko nastala nepopravljiva škoda, če se naročilo ne izvede takoj.

Vodja naročila lahko izvaja poizvedbe in pridobiva ponudbe po telefonu, z dopisi po pošti, faksu ali elektronski pošti, prek elektronskih medijev (internet) oziroma na način, ki ustreza naravi in vrednosti naročila. V primeru preverjanja po telefonu mora vodja naročila pridobljene informacije zapisati v obliki zaznamka. Gradiva z informacijami o naročilu in ponudbah mora vodja naročila shraniti v spisu.

Vodja naročila zbira in hrani dokumentacijo⁴, dokler trajajo pogodbeni roki in garancijska doba. Dokumentacija, ki ima vpliv na davčno področje, se hrani 15 let, ostala dokumentacija (zavrnjene ponudbe, ipd.) pa najmanj 5 let po oddaji naročila. Pri naročilih, ki se oddajo na osnovi pogodbe, se po podpisu en izvod pogodbe predloži finančni službi skupaj s kopijo sklepa o oddaji naročila.

Naročila do vrednosti 5.000 EUR se praviloma oddajo z naročilnico, lahko pa tudi s pogodbo, če je tako primerneje glede na naravo, zahtevnost in specifičnost naročila. Naročilo poteka po naslednjem postopku:

- vodja naročila zbira informacije in pridobiva ponudbe,
- izvedejo se morebitna pogajanja,
- sestavi se predlog za izdajo naročilnice s podpisi predlagatelja (ki je vodja naročila), vodje organizacijske enote in odgovorne osebe naročnika,
- naročilnica se izda.

Predlog za izdajo naročilnice mora vsebovati predvsem:

- predmet, vrsto in količino naročila ali opis storitve,
- ime organizacijske enote, ki naroča,
- vrsto blaga ali storitve,
- ocenjeno vrednost naročila brez DDV,
- opredelitev postavke v finančnem načrtu, na katerem ima naročnik zagotovljena sredstva za izvedbo naročila,
- naziv in naslov izbranega ponudnika,
- vrednost izbrane ponudbe brez DDV in z DDV,
- podatke o pridobljenih ponudbah (ponudnik in cena), uporabljena merila za izbiro najugodnejšega ponudnika, uporabljeni način pridobitve najugodnejše ponudbe,
- ime in podpis vodje naročila (predlagatelj),
- ime in podpis vodje organizacijske enote,
- ime in podpis odgovorne osebe naročnika in datum.

Naročilnica se izda na osnovi podpisanega predloga za izdajo naročilnice v dveh izvodih. Na naročilnici mora biti zaporedna številka izdane naročilnice, datum izdaje naročilnice, vsebina naročila, naziv in naslov dobavitelja. Naročilnico podpišeta vodja naročila in odgovorna oseba naročnika. Žigosan izvod podpisane naročilnice prejme izbrani dobavitelj oz. izvajalec, drugi izvod pa ostane v evidenci naročilnic v tajništvu organizacije.

⁴ Za izvedbo postopkov po teh navodilih, se uporabljajo naslednji dokumenti:

- predlog za izdajo naročilnice in naročilnica,
- sklep o oddaji naročila za naročila, ki so sklenjena s pogodbo in je priloga tega navodila.

Oddaja naročila s pogodbo. Naročilo s sklenitvijo pogodbe poteka po naslednjem postopku:

- določitev vodje naročila (določi ga uradna oseba naročnika),
- pridobitev zaporedne evidenčne številke naročila,
- zbiranje informacij in pridobivanje ponudb,
- morebitna pogajanja,
- izdaja in potrditev sklepa o oddaji naročila,
- priprava in podpis pogodbe,
- pregled pogodbe s strani pravnega sektorja,
- posredovanje izvoda podpisane pogodbe v finančno službo.

Pogodba se oštevilči z zaporedno evidenčno številko naročila. O naročilih, oddanih s pogodbo, se v finančni službi vodi posebna evidenca, ki obsega navedbo predmeta naročila, izbranega ponudnika in vrednost posameznega naročila.

1.2 Metodologije razvoja programske rešitve

Metodologija je skupek postopkov, tehnik in metod, ki se uporabljajo pri reševanju problemov. Pod pojmom »metodologija gradnje informacijskih sistemov« si v praksi največkrat predstavljamo organizacijsko-tehnično znanje, ki ga uporabljamo pri zasnovi in izdelavi računalniških storitev (Kovačič & Vintar, 1994, str. 24).

Metodologija za razvoj informacijskih sistemov je zbirka postopkov, tehnik, orodij in dokumentacije, ki jih uporabljajo programerji pri implementaciji novega informacijskega sistema. Metodologija je sestavljena iz faz in podfaz sistema, ki vodijo programerje pri izbiri tehnik in jim pomagajo načrtovati, upravljati in ocenjevati projekt izgradnje informacijskega sistema (Avison & Fitzgerald, 1995, str. 13).

Metodologijo razvoja pri izgradnji informacijskega sistema uporabljamo z namenom uporabe različnih ciljev, ki nam jih omogoča beleženje informacijskih zahtev (Avison & Fitzgerald, 1995, str. 11):

- metodologija razvoja pomaga pri analizi in opredeljevanju uporabniških zahtev;
- sistematična metoda razvoja mora biti podana tako, da lahko napredek razvoja učinkovito nadzorujemo. Pri obsežnih projektih je nujna uporaba kontrolnih točk in dobro definiranje faz, ki omogočajo uporabo najrazličnejših tehnik razvoja programskih rešitev;
- metodologija naj omogoča razvoj informacijskega sistema v okviru sprejemljivih stroškov in v primernem časovnem okviru;

- programska rešitev naj bo enostavna za vzdrževanje. Treba je načrtovati tudi manjše spremembe programske rešitve na osnovi sprememb v organizaciji in okolju;
- v primeru sprememb pri izgradnji sistema naj se te predvidevajo čim bolj v začetni fazi, saj se v nasprotnem primeru stroški sprememb povečujejo;
- informacijski sistem naj bo prijazen do uporabnikov in enostaven za uporabo.

Modeli za razvoj informacijski rešitev, ki sem jih opisal, imajo tako prednosti kot slabosti in so uporabni v konkretnem primeru v določeni organizaciji. Za razvoj programske rešitve javna naročila po evidenčnem postopku je značilno:

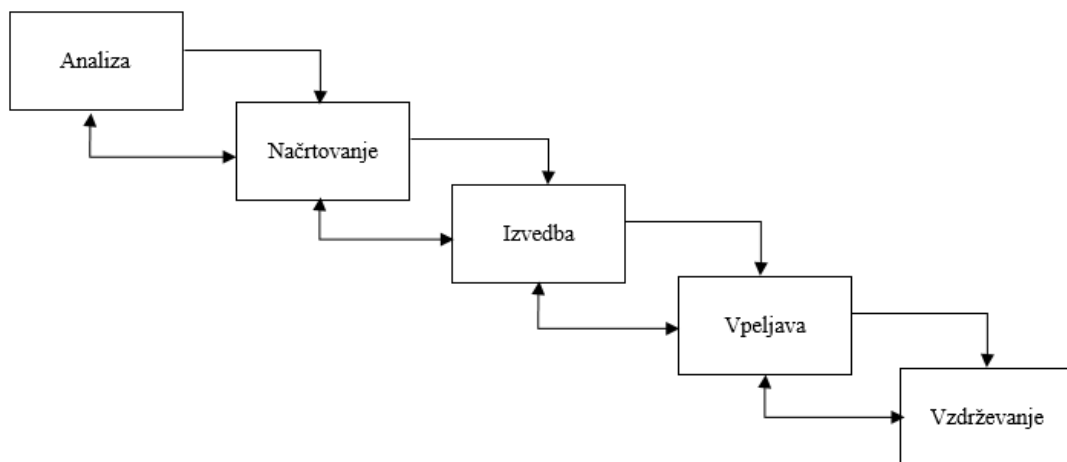
- poslovni proces javnih naročil po evidenčnem postopku se izvaja zaporedno. Posamezne faze procesa je mogoče zelo natančno definirati. Prevladujejo strukturirani procesi, ki jih je možno definirati vnaprej;
- aktivnosti pri razvoju informacijskega sistema se izvajajo sočasno s fazami razvoja. Razporeditev aktivnosti med posamezne faze razvoja je enostavna;
- zahteve oziroma potrebe naročnika so lahko jasno definirane tako z vidika zakonodaje kot notranjega okolja organizacije;
- strošek izgradnje informacijskega sistema je mogoče dokaj natančno opredeliti in v določenem časovnem okvirju tudi strošek vzdrževanja;
- vsaka faza razvoja je lahko natančno opredeljena in zato tudi natančno dokumentirana.

1.2.1 Zaporedni ali linearni model

Zaporedni model velja za enega od najstarejših modelov razvoja informacijskega sistema. Faze razvoja si sledijo popolnoma zaporedno in se ne prekrivajo. V literaturi je zaporedni model največkrat poimenovan model slapa (angl. *Waterfall Model*), saj razvoj poteka podobno, kot se voda pretaka z višjega nivoja na nižji nivo. Model predstavlja osnovno izhodišče za nastanek mnogih kasneje razvitih modelov. Zaporedni model je organiziran v seriji linearnih faz, to pa pomeni natančno določen in dokumentiran postopek razvoja programskih rešitev (Guimarães & Vilela, 2005, str. 339–347). V literaturi je zaporedni model (Slika 1) predstavljen z različnim številom faz razvoja, najpogosteje je opisan kot model s petimi fazami razvoja.

Zaporedni model se je pojavil v sedemdesetih letih dvajsetega stoletja in je natančno definiral aktivnosti posameznih faz razvoja. Namenjen je bil predvsem za razvoj enostavnejših aplikacij. Zaporedni model uvaja sistematično izvajanje aktivnosti po posameznih fazah, ki so pregledne, testirane in dobro dokumentirane. Njegova glavna pomanjkljivost je, da v okviru procesa skoraj nikoli ne zaključimo ene faze pred drugo, ampak po potrebi med fazami prehajamo. Pri tem pristopu pa vrnitev v predhodne faze ni več mogoča, kar je velika pomanjkljivost, saj se je treba velikokrat vračati v fazo analize. Lahko se tudi zgodi, da se iz faze testiranja nenehno vračamo v fazo načrtovanja, kar predstavlja težavo pri vpeljavi določene aplikacije (Winston, 1970, str. 328–338).

Slika 1: Zaporedni model razvoja



Vir: B. K. Jayaswal & P. C. Patton, *Software Development Methodology Today*, 2006, str. 12.

V nadaljevanju na kratko opišem glavne faze zaporednega modela razvoja, povzeto po navedeni literaturi. V prvi fazi analize nas zanima, katere postopke mora podpirati programska rešitev. Na osnovi vsebine projekta lahko določimo procese, poslovna pravila in podatke, ki so pomembni za razvoj programske rešitve. V fazi analize torej natančno opišemo vsebino obravnavanega področja in procese, ki se v njem izvajajo.

V fazi načrtovanja se izdelava načrt programske rešitve na osnovi analize. Običajno se v obliki dokumenta pripravi arhitektura programske rešitve. V dokumentu se opredeli še podroben načrt uvedbe nove programske rešitve, tj. finančni in terminski načrt rešitve. Proučijo se tudi morebitna tveganja ob uvedbi nove programske rešitve. Načrtovanje programske rešitve običajno poteka v projektni skupini. V fazi izvedbe se na osnovi načrta izdelava programska rešitev. V tej fazi se najprej izdelava podatkovna baza in programski vmesniki ter pripravi ustrezna tehnična dokumentacija. Na osnovi testnih podatkov in delovnih različic programske rešitve se izvedejo testi programske rešitve in celotnega sistema. V fazi izvedbe se pripravijo tudi vse aktivnosti za začetek uvajanja uporabnikov za uporabo programske rešitve. V fazi vpeljave se programska rešitev vpelje v nov sistem. Izvede se migracija podatkov oziroma vpis začetnih podatkov, potrebnih za delovanje aplikacije. V tej fazi se dokončajo tudi navodila za uporabnike in vsi ostali dokumenti. Zelo pomembno je, kako bomo izvedli prehod v nov sistem. V primeru, da se pri prehodu kaj zalomi, je treba predvideti alternativne scenarije. Faza vzdrževanja se po fazi vpeljave programske rešitve izvaja neprestano. V tej fazi spremljamo delovanje programske rešitve in odpravljamo večje ali manjše napake v okviru najrazličnejših nadgradenj.

V različni literaturi najdemo kar nekaj razprav o tem, ali je zaporedni model razvoja primeren model za razvoj programskih rešitev. Dvom v uporabo zaporednega modela je

izrazil že eden izmed ustanoviteljev modela W. R. Winston (1970, str. 328–338). Tudi razvijalci programskih rešitev imajo o primernosti zaporednega modela razvoja različna mnenja. Michael Stal (2008) na primer v članku z naslovom »Zakaj linearni modeli ne delujejo« poda mnenje, zakaj so agilne metode razvoja primernejše od zaporednega modela razvoja. V članku razpravlja o tem, da v teoriji zaporedni model razvoja sicer lahko deluje, v resničnem življenju projekta pa se moramo spopasti z napakami in posledično z neuspehom. Pri agilnih metodah se uporablja testno voden razvoj v obliki varnostnih mrež, kar razvijalcem omogoča boljše komunikacijo z uporabniki in hitrejšo odkrivanje morebitnih težav.

David Parnas (1986, str. 251) navaja razloge, zakaj se razvoj programske rešitve nikoli ne bo izvajal racionalno:

- naročnik nikoli točno ne ve, kakšno programsko rešitev potrebuje,
- določene zahteve spoznamo šele v času implementacije,
- izkušnje kažejo, da ljudje ne morejo v celoti razumeti obilice podatkov, ki jih je treba upoštevati, da bi oblikovali in zgradili pravi sistem,
- čeprav lahko definiramo vse zahteve za izdelavo programske rešitve, pa ne moremo predvideti zunanjih dejavnikov, ki vplivajo na sistem,
- velikokrat dajemo poudarek podrobnim idejam ali teoretičnim znanjem, namesto da bi upoštevali glavne ideje za izgradnjo sistema.

Kljub slabostim zaporednega modela pa je treba poudariti, da lahko lastnosti zaporednega modela danes najdemo v različnih modelih. Zaporedni model uporabljamo predvsem za izgradnjo informacijskih sistemov, kjer so zahteve zelo dobro znane in jih lahko natančno definiramo. Povzeto po spletni strani Software Testing Help (2016) so prednosti zaporednega modela razvoja naslednje:

- je preprost in enostaven za razumevanje in uporabo,
- uporaben je predvsem za manjše projekte,
- faze so natančno definirane,
- je enostaven za vzdrževanje,
- rešitev je podrobno dokumentirana,
- vhod in izhod sta zelo dobro definirana, kar zagotavlja preprost in sistematičen pristop pri razvoju ter potrebno kakovost razvoja.

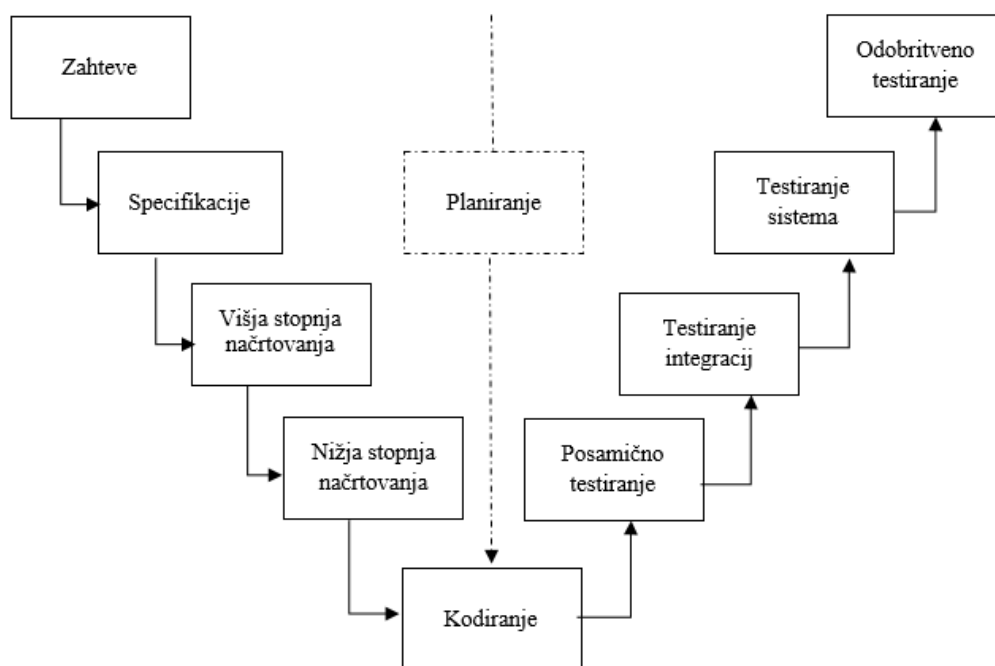
Slabosti zaporednega modela razvoja so naslednje (ibid.):

- zelo težko se je vračati v prejšnje faze razvoja,
- zelo težko je upoštevati zahteve za spremembe,

- šele na koncu razvoja vidimo končni rezultat, saj nam model ne omogoča vmesnega vpogleda, podobno kot pri prototipnem razvoju,
- ni primeren za večje projekte, saj so dejavniki tveganja previsoki,
- ni primeren za projekte, kjer se zahteve pogosto spreminjajo,
- faza testiranja se izvaja šele ob koncu projekta, zato je zelo težko prepoznati dejavnike tveganja v zgodnji fazi razvoja.

Razširjeni zaporedni model razvoja, tako imenovani V-model, je v letu 1980 predstavil Paul Rook. Model ima večji poudarek na testiranju v vsaki fazi razvoja. Takšna metoda razvoja temelji na potrjevanju podrobne dokumentacije v vsaki fazi razvoja. Šele ko je dokumentacija v vsakem koraku razvoja potrjena, se lahko preide v novo fazo razvoja. Z nenehnim testiranjem se poveča učinkovitost postopka razvoja ter zgodnjega odkrivanja in odpravljanja napak. V-model razvoja je prikazan na Sliki 2 in ima zelo podobne faze kot zaporedni model, le da se vzporedno z vsako fazo izvaja še faza testiranja (Isaias & Issa, 2015, str. 28–29). V-model razvoja se uporablja, kjer so majhne možnosti za spremembo sistema in je potreba po varnosti in stabilnosti sistema velika. Uporablja se za manjše projekte, kjer je izvedljivo celotno testiranje. Prednost V-modela v primerjavi z zaporednim modelom razvoja je v večjem uspehu sistema, saj ga sproti testiramo. Na ta način tudi sproti odpravljamo napake in zmanjšamo možnost za pojav večjih napak v sistemu. Slabost V-modela razvoja se kaže predvsem v dolgotrajnih testih in njihovih nadgrajevanjih v primerih manjših sprememb razvoja (TutorialsPoint, 2016).

Slika 2: V-model razvoja

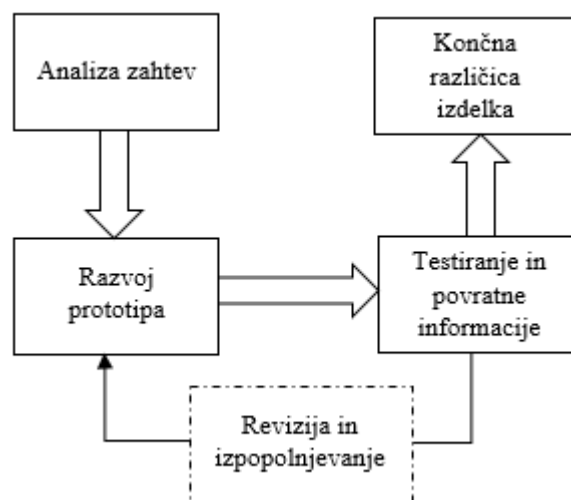


Vir: P. Isaias & T. Issa, *High Level Models and Methodologies for Information Systems*, 2015, str. 28.

1.2.2 Prototipni model

Prototipni model razvoja se je pojavil v zgodnjih osemdesetih letih kot agilni pristop k razvoju informacijskega sistema. Prototipni model razvoja je bolj dinamičen in bolj odziven na potrebe uporabnikov, zato je tudi učinkovitejši. Model razvoja temelji na ideji ustvarjanja poskusne različice sistema, zato ga imenujemo prototipni model razvoja. Cilj modela je razviti sistem v različnih izvedbah in ga dosledno izboljšati do končnega izdelka. Poudarek je na oblikovanju programske opreme in manj na dokumentiranju. Za uporabnika predstavlja usmerjen pristop, saj v času razvoja pridobi veliko povratnih informacij. Prototipni model razvoja vključuje štiri stopnje razvoja sistema in je prikazan na Sliki 3. Prva stopnja je identifikacija in analiziranje uporabnikovih zahtev. V naslednji fazi skupina programerjev izdelava informacijski sistem, ki ga uporabniki testirajo in podajo morebitne pripombe. Testiranje tako poteka v realnem času. V primeru zahtevkov za spremembo informacijskega sistema se izvedejo popravki prototipa in znova dajo v testiranje. Krog popravkov in testiranja se izvaja toliko časa, dokler uporabniki niso zadovoljni z izdelkom oziroma informacijski sistem ne zahteva bistvenih sprememb oziroma posodobitev (Isaias & Issa, 2015, str. 33–34).

Slika 3: Prototipni model razvoja



Vir: P. Isaias & T. Issa, *High Level Models and Methodologies for Information Systems*, 2015, str. 33.

Prototipni model razvoja je nastal kot posledica slabosti zaporednega modela razvoja. Lahko ga opišemo kot orodje za modeliranje, kjer se na delujočem sistemu preizkušajo ideje, odločitve ter logične in tehnične rešitve. Ker se spremembe pri prototipnem razvoju izvajajo hitro in enostavno, tak model predstavlja zelo dobro osnovo za razvoj informacijskega sistema (Kovačič & Vintar, 1994, str. 45–48).

Kovačič in Vintar (1994, str. 238) navajata, da mora prototip vsebovati določene lastnosti oziroma biti dovolj nujen za izvedbo, dovolj poceni in dolgoročen, vpliven, prepričljiv in majhen za izvedbo.

Avison in Fitzgerald (1995, str. 77) pravita, da prototip predstavlja predstavitev funkcionalnosti programske rešitve in se lahko uporablja, kjer:

- je komunikacija med uporabniki in analitiki slaba,
- je področje rešitve slabo definirano,
- bi bil strošek razvoja neustrezne programske rešitve visok,
- tehnologija za izdelavo rešitve v organizaciji ni dovolj dobro poznana.

Spletna stran podjetja Tutorialspoint (2016) opisuje štiri glavne tipe prototipnega razvoja programskih rešitev:

- hiter prototipni razvoj. Ta vrsta prototipov se uporablja za področja, kjer so potrebe po razvoju informacijske rešitve manjše in zahteve slabo poznane. Ko so zahteve za razvoj informacijske rešitve bolj jasno opredeljene, se tak prototip dejansko zavrže;
- evolucijski razvoj prototipov. Razvoj informacijske rešitve se imenuje tudi maketa prototipov in poteka na osnovi na začetku minimalno opisanih funkcionalnosti. Razviti so samo najpomembnejši gradniki, ki predstavljajo osnovo za razvoj celotnega sistema v nadaljevanju. Ostale komponente so razvite šele, ko razumemo zahteve za razvoj informacijske rešitve;
- inkrementalni prototipni razvoj. Nanaša se na izgradnjo več funkcionalnih podsistemov, ki se integrirajo v celoten sistem;
- ekstremni prototipni razvoj. Uporablja se predvsem pri razvoju spletnih aplikacij. Razdeli se v tri razvojne faze. V prvi fazi izdelamo stran Hypertext Markup Language (v nadaljevanju HTML), v drugi fazi opremimo zaslonske maske in v tretji fazi funkcionalnosti dejansko implementiramo v prototip.

Spletna stran I answer 4 U (2016) navaja prednosti prototipnega modela razvoja:

- ko je prototip prikazan uporabniku, ta dobi jasen pogled na funkcionalnost programske rešitve, zato lahko predlaga spremembe in modifikacije;
- prototipni model se običajno uporablja za naročnike, ki se ne spoznajo najbolje na razvoj informacijskih rešitev, zato tudi ne znajo definirati zahtev in pričakovanj za določeno programsko opremo;
- v primeru, ko naročnik ni prepričan v sposobnosti razvijalca programske opreme, ga prosi za razvoj manjšega prototipa, na osnovi katerega presodi sposobnosti razvijalca;
- uporablja se za demonstracijo potencialnim investitorjem z namenom pridobitve določenih sredstev pri razvoju programske rešitve;

- zmanjšuje tveganje za neuspeh, saj je morebitna tveganja mogoče spoznati dovolj zgodaj, da se lahko sprejmejo določeni ukrepi;
- sodelovanje med projektno skupino in naročniki ustvarja zelo ustvarjalno okolje za razvoj programske rešitve;
- čas za razvoj programske rešitve se zmanjšuje, saj ima razvijalec boljši pregled nad tem, kako bi moral pristopiti k razvoju.

Pomanjkljivosti prototipnega modela razvoja so (ibid.):

- zagonski stroški za vzpostavitev projektne ekipe za razvoj programske rešitve so lahko visoki;
- lahko se zgodi, da bo programska rešitev povsem neustrezna zaradi povsem napačno razumljenih zahtev naročnika;
- prototipni model razvoja je počasen proces;
- prevelika vključenost naročnika pri razvoju programske rešitve ni vedno po volji razvijalca;
- preveč sprememb lahko povzroči težave oziroma neuskklajenost razvojne ekipe.

1.2.3 Spiralni model

Prvi spiralni model razvoja programske rešitve je opisal Barry Boehm konec osemdesetih let. V model je vpeljal nekaj, česar drugi modeli do takrat niso vključevali, in sicer analizo tveganja. V bistvu poskuša spiralni model združiti predvsem ustrezne lastnosti ostalih razvojnih modelov (zaporedni in inkrementalni model), saj so projekti običajno preveč prilagodljivi za ostale razvojne modele. Model sicer spominja na posamezne faze inkrementalnega modela razvoja, vendar poudarek na oceni tveganja predstavlja bistveno razliko. Zaradi velikega poudarka na analizi tveganj je spiralni model razvoja primeren za večje projekte (Isaias & Issa, 2015, str. 26–28).

Spiralni model razvoja programske rešitve je proces, ki je sestavljen iz elementov načrtovanja in prototipnega razvoja po korakih. Model je torej kombinacija zaporednega in prototipnega modela in se uporablja pri razvoju večjih in kompleksnih sistemov. Prvi modeli so temeljili na osnovi tradicionalnega razvoja programske rešitve, zato so bili poimenovani »modeli slapa«. Pri spiralnem modelu gre za iterativni model razvoja programske rešitve in podrobno razlaga, zakaj so povezave pomembne (Boehm, 1988, str. 63–67). Spiralni model razvoja (Slika 4) je sestavljen iz štirih faz (Boehm, 1988, str. 64):

- določitev ciljev projekta,
- načrtovanja sistema,
- razvoja v korakih,
- ovrednotenja in testiranja.

Model določa za vsak cikel enake korake. Začne se na osi x in konča, ko spirala zaobjame kot 360 stopinj. Pri spiralnem modelu se vsak cikel spirale začne z ugotavljanjem ciljev oziroma možnosti za doseg ciljev in omejitev. Pri ugotavljanju možnosti za doseg ciljev je bistveno ugotavljanje tveganj pri projektu. Tveganje je treba identificirati in pripraviti ustrezne strategije, da ga odpravimo. Tveganje lahko odpravimo z izdelavo prototipov, vprašalnikov, raznih simulacij in z drugimi tehnikami. Tehnike za odpravljanje tveganja ponavljamo toliko časa, da ga v čim večji meri odpravimo in da lahko preidemo v fazo načrta razvoja in potrjenega načina testiranja. Sledi faza, ki se imenuje operacijski prototip, v kateri se izvede programiranje, testiranje in uvedba v produkcijo (Boehm, 1988, str. 63–67). Spiralni model se uporablja pri razvoju, kjer so zahteve za izdelavo programske rešitve kompleksne in so zahteve za dopolnitev programske rešitve relativno pogoste.

Slika 4: Spiralni model razvoja



Vir: B. Boehm, A Spiral Model of Software Development and Enhancement, 1988, str. 64.

Spletna stran Bagya Tech (2016) navaja naslednje prednosti prototipnega modela razvoja:

- že v začetku se izognemo velikemu tveganju pri razvoju zaradi nenehnega ugotavljanja tveganj pri razvoju projekta,
- primeren je za velike in kritične projekte,
- dodatne funkcionalnosti se lahko vključujejo v kateri koli fazi projekta,
- zahteve se lahko definirajo veliko bolj natančno,

- uporabniki oziroma naročniki zelo hitro vidijo celotno rešitev,
- omogočen je zelo dober nadzor nad dokumentacijo,
- omogoča pravočasen razvoj programske rešitve.

Pomanjkljivosti prototipnega modela razvoja so (ibid.):

- lahko predstavlja zelo drag model razvoja programske rešitve,
- analiza tveganja zahteva visoko usposobljene strokovnjake,
- uspeh projekta je v veliki meri odvisen od uspešnosti analize tveganja,
- neprimeren je za manjše projekte,
- zaradi spiralne oblike razvoja se lahko razvoj ponavlja nedoločeno dolgo,
- postopek razvoja je zapleten,
- zaradi velikega števila vmesnih faz je treba izdelati veliko dokumentacije.

1.2.4 Inkrementalni model

Inkrementalni model spada med tradicionalne modele razvoja informacijskih sistemov in je nadgradnja razvoja zaporednega modela. Faze pri inkrementalnem modelu se izvajajo hitreje in so bolj prilagodljive. Značilnost modela je, da se razvoj programske rešitve izvaja v obliki različic glede na zahteve naročnika oziroma uporabnika, vse dokler z izboljšavami nismo v celoti zadovoljni. Prav tako je vsaka faza razvoja načrtovana tako, da lahko predstavlja podsistem, ki ga kasneje lahko združimo v celoten sistem. Ta model v bistvu opisuje postopen proces razvoja programske rešitve z dodajanjem novih funkcij vse do zaključka sistema. Ker se sistem razvija v vsaki fazi posebej, omogoča še boljši razvoj naslednjih faz in lažje testiranje na koncu (Isaias & Issa, 2015, str. 24–25).

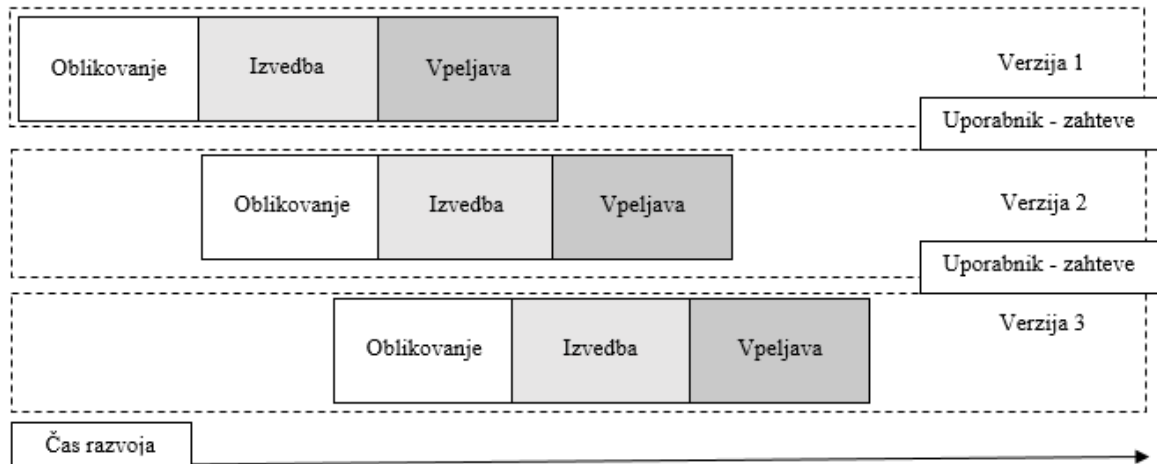
Na spletni strani Software Development Life Cycle (2016) je inkrementalni model opisan kot model, sestavljen iz več modulov. Moduli so oblikovani glede na vlogo in stopnjo prioritete v okviru razvoja programske rešitve. Vsak modul je samostojna funkcija, ki jo je treba razviti v skladu z izvajanjem in napredovanjem projekta. Čeprav so moduli med seboj povezani, lahko delujejo samostojno, ne glede na zahteve in potrebe drugih modulov. Inkrementalni model se običajno uporablja za razvoj velikih programskih rešitev, ki predvidevajo dodatne funkcionalnosti samostojnih modelov na dolgi rok.

Inkrementalni model, prikazan na Sliki 5, je v vsaki stopnji razvoja sestavljen iz oblikovanja, izvedbe in vpeljave. Na prvi stopnji modela se pripravijo funkcionalnosti, v izvedbeni fazi se načrt implementira in testira. Na koncu se izvede analiza delno razvite programske rešitve. Razvojni proces se izvaja toliko časa, dokler niso implementirane vse funkcije programske rešitve.

Takšni modeli razvoja se uporabljajo pri razvoju programskih rešitev, kjer so zahteve zelo jasno opredeljene in so lahko implementirane v več fazah. Večinoma se taki modeli

uporabljajo pri razvoju spletnih aplikacij in v podjetjih, ki proizvajajo različne izdelke ali ponujajo različne storitve.

Slika 5: Inkrementalni model razvoja



Vir: P. Isaias & T. Issa, *High Level Models and Methodologies for Information Systems*, 2015, str. 25.

Spletna stran Incremental Model (2016) navaja naslednje prednosti inkrementalnega modela razvoja:

- v zgodnjem razvoju ustvarja hitro delujočo programsko rešitev,
- programska rešitev je bolj prilagodljiva in cenejša za razvoj dodatnih funkcionalnosti,
- omogoča lažje testiranje in razhroščevanje v primeru manjših nadgradenj,
- omogoča lažje obvladovanje tveganj, saj so moduli med seboj neodvisni,
- vsaka nadgradnja je enostavno izvršljiva.

Pomanjkljivosti inkrementalnega modela razvoja so (ibid.):

- težave pri podpiranju različnih različic programskih rešitev,
- težave pri sistemski arhitekturi, saj zahteve za razvoj niso poznane vnaprej oziroma za celotni življenjski cikel programske rešitve,
- potrebna je dobra analiza,
- stroški razvoja lahko presežejo načrtovane stroške organizacije.

1.3 Programska orodja

1.3.1 Visual Studio 2012

Prva različica programskega orodja, ki ga je razvil Microsoft, je bila različica BASIC iz leta 1975. Takrat so bili programski jeziki večinoma interpretativni. V zadnjih desetletjih je programiranje zelo napredovalo. Visual Studio je postal eno izmed največjih razvojnih orodij za razvoj aplikacij v različnih programskih jezikih na različnih platformah. V obdobju razvoja 32-bitnih računalnikov je razvojno orodje Microsoft Visual Studio postalo integrirano razvojno orodje, imenovano Integrated Development Environment (v nadaljevanju IDE), ki ne vsebuje samo prevajalnika, temveč tudi številne napredne funkcije. Poleg naprednih funkcij se uporabljajo tudi različni uporabniški vmesniki v povezavi z razvojnimi okni. Razvijalcem je omogočeno spreminjanje lastnosti in komponent oken in nadzor nad uporabniki (Randolph & Gardner, 2008, str. xxxviii).

Visual Studio podpira različne programske jezike. Urejevalnik kode in razhroščevalnik omogočata uporabo skoraj katerega koli programskega jezika, npr. C, C++, VB.NET, C# in F#. Visual Studio ima tudi podporno okolje za programske jezike M, Python in Ruby. Prav tako podpira Extensible Markup Language (v nadaljevanju XML)/Extensible Stylesheet Language Transformations (XSLT), HTML/Extensible Hypertext Markup Language (XHTML), JavaScript in programske jezike Cascading Style Sheets (v nadaljevanju CSS) (Microsoft, 2012, str. 20).

Microsoft je prvo razvojno okolje Visual Studio razvil v letu 2002 in ga poimenoval Visual Studio .NET 2002. Vse od takrat dalje je Microsoft razvil več različic razvojnega okolja Visual Studio. Vsaka nova različica podpira nove tehnologije, nove programske jezike in večje število knjižnic. Poznamo več vrst različic razvojnega okolja Visual Studio:

- Visual Studio Express Edition je osnovna različica in je namenjena začetnikom, ki lahko pripravijo povsem funkcionalne aplikacije. Z večanjem potreb po dodatnem razvoju se lahko taka aplikacija prenese na nove različice razvojnega okolja. Različica se nahaja na spletni strani Microsoft in je brezplačna (Moore, 2010, str. 62–63).
- Različica Visual Studio Professional je namenjena razvijalcem .NET. Razvojno okolje je namenjeno naprednim oblikovalcem, katerim programiranje zelo veliko pomeni in ga programerji največ uporabljajo. Okolje podpira vse programske jezike Microsoft .NET, vizualna orodja za podatkovne baze, oblikovalce za shemo in sloge XML, napredno iskanje napak, napredna razvojna orodja, razvoj procedur, funkcij in prožilcev za Structured Query Language (v nadaljevanju SQL) Server 2005 ali SQL Server 2008 in Crystal Reports (Moore, 2010, str. 64).
- Različica Visual Studio Professional Edition je namenjena programiranju komercialnih izdelkov,

- Različica Visual Studio Premium Edition je namenjena skupini programerjev pri agilnem programiranju.
- Visual Studio Ultimate Edition je najbolj izpopolnjena različica in je namenjena različnim skupinam razvijalcev kot so načrtovalci in preizkuševalci aplikacij ter razvijalci podatkovnih baz (Moore, 2010, str. 832–835).

Pri razvoju informacijskega sistema za vodenje evidenc o javnih naročilih male vrednosti po evidenčnem postopku sem uporabil razvojno okolje Visual Studio Professional.

Visual Studio Professional. Orodje je namenjeno razvoju najrazličnejših vrst namiznih in spletnih aplikacij v različnih programskih jezikih. Osnovne komponente orodja so urejevalnik kode, razhroščevalnik in orodjarna.

Urejevalnik kode je podprt za vrsto programskih jezikov. Uporabnikom je na voljo funkcija Intellisense, ki uporabniku prikazuje možne sintakse za pisanje spremenljivk, funkcij, metod in programskih zank z možnostjo obarvanja kode. Urejevalnik kode omogoča tudi vstavljanje zaznamkov v kodi z namenom hitre navigacije. Kodo se lahko združi tudi v posamezne bloke in ponavljajočo se kodo v predloge (Microsoft, 2012, str. 84–87). Urejevalnik kode podpira tudi orodje, ki ponuja možnosti za poimenovanje in prerazporeditev spremenljivk, funkcij in metod z namenom, da se ne ponavljajo (angl. *Code Refactoring*). Visual Studio ima tudi opcijo prevajanja programske kode v ozadju, kar pomeni, da uporabniku ni treba prevesti celotne programske kode. Uporabniku se napake označijo s podčrtano rdečo valovito črto.

Z razhroščevalnikom preverjamo napake v programski kodi. Razhroščevalnik lahko deluje tudi na nivoju računalnika, kjer se programska koda izvaja. To pomeni, da preverjamo tudi ostale procese v računalniku, ki se izvajajo v okviru izvajalnega programa. Razhroščevalnik omogoča vstavljanje prelomnih točk v programsko kodo in na ta način ustavljanje programa na določeni prelomni točki. Tako lahko preverjamo posamezne vrednosti spremenljivk parametrov ali funkcij in jih v času preverjanja lahko tudi popravljamo. Omogočen je namig za popravek programske kode v primeru, ko kazalec miške zadržimo na določeni spremenljivki.

Microsoft (2012, str. 69) v navodilih o izdelku Visual Studio 2012 navaja, da Visual Studio vključuje zmogljivo iskanje in odpravljanje napak s številnimi novimi prednostmi razhroščevanja aplikacij Windows, kot so:

- robustno testiranje aplikacij v okolju Microsoft Test,
- simulator Windows 8 na primer za testiranje posameznih scenarijev za monitor različnih resolucij, usmeritev in postavitev,
- izboljšano oddaljeno odpravljanje napak s povezavo naprave prek žičnega ali brezžičnega omrežja,

- hitrejšje odpravljanje napak z možnostjo urejanja in nadaljevanja izvajanja programske kode,
- vzporedno okno za ocenjevanje izrazov v nitih programa,
- vzporedno prodorno preverjanje (angl. *Pervasive »Flagging« Support*) in označevanje programske kode.

Orodjarna je sestavljena iz različnih kontrolnikov, ki ji uporabljamo na obrazcih (angl. *Form*). Nekateri pomembnejši kontrolniki so (Michelbrink, 2004, str. 4–6):

- oznaka za besedilo se uporablja za opis raznih elementov obrazca in tekstovnih polj in je med delovanjem programa ne moremo spremeniti;
- potrditveno polje uporabimo takrat, kadar uporabniki izbirajo med eno ali več možnostmi, ki se med seboj izključujejo;
- ukazni gumb, s katerim potrdimo neko izvajanje funkcij v programu;
- spustni seznam, iz katerega uporabnik izbere določen element, ki je običajno zapisan v šifrantu aplikacije;
- okence za poljubno pisanje besedila, v katerega uporabnik vpiše poljubno besedilo. Program lahko besedilo v nadaljevanju znova prikaže;
- podatkovna mreža je namenjena prikazovanju podatkov v obliki tabel. Običajno so podatki, ki jih želimo prikazati v tabeli, zapisani v ustrezni podatkovni bazi. Podatkovna mreža ima vgrajeno vrsto funkcionalnosti, kot so na primer sortiranje, grupiranje in razvrščanje podatkov po različnih atributih;
- oznaka za dodajanje različnih grafičnih podob (aktivnih ali neaktivnih);
- oznaka za koledar omogoča vstavljanje koledarja z namenom izbire določene datuma;
- drugo.

Visual Basic. Tylee (1998, str. 2) navaja, da je Visual Basic orodje, ki omogoča razvoj aplikacij Windows Graphical User Interface (v nadaljevanju GUI). Napisano je dogodkovno, kar pomeni, da se programska koda izvede šele ob proženju nekega dogodka (npr. izbira gumba, izbira menija).

Skozi čas se je razvilo kar nekaj različic programskega jezika Visual Basic. Najbolj znana različica 6.0 je bila izdana leta 1998 in se ni bistveno spremenila vse do različice Visual Basic .net leta 2002, ko je postala del ogrodja Microsoft .NET. Različice programskega jezika Visual Basic v okviru ogrodja .NET so si sledile kot različice 7.0 in vse do zadnje različice 14.0 v letu 2015.

O tem, ali je Visual Basic.net povsem druga različica kot Visual Basic 6.0, bi lahko veliko razpravljali. V ogrodju .NET obstajajo nove funkcije, kot je na primer strukturiranje izjem. Pomembno je omeniti tudi spremembo dveh podatkovnih tipov s prehodom na ogrodje .NET. Podatkovni tip integer se je povečal na dolžino 32 bitov, podatkovni tip long pa na

dolžino 64 bitov. Veliko novosti se je pojavilo tudi z objektnim programiranjem, katerega lastnosti so bile vključene na primer v knjižnice razredov Framework Class Library (v nadaljevanju FCL) in so opisane v razdelku Ogrodje .Net.

Nekatere funkcije in knjižnice v ogrodju .NET ne obstajajo več, nekatere so še na voljo, vendar niso več tako učinkovite. Tudi če jih poizkušamo prevesti, je potrebno določeno reprogramiranje, da lahko aplikacija izkoristi vse značilnosti novega jezika Visual Basic. Členi, ki sestavljajo programsko kodo Visual Basic, so (Mesojedec, 1998, str. 79–124):

- elementi kode. V jeziku Visual Basic je vsaka vrstica lahko programski stavek, ki je sestavljen iz različnih elementov. Konstante, spremenljivke, podprograme poimenujemo z določenimi imeni;
- izrazi in operatorji. Najpogosteje uporabljeni operatorji so računski, primerjalni in logični. Uporabljamo še operatorje za združevanje nizov in operatorje za določevanje izračunavanja vrstnega reda izrazov;
- podatki in vrste. Podatki so predstavljeni v obliki spremenljivk. Spremenljivka je lahko število, logična vrednost ali besedilo. Vsaki spremenljivki določimo tudi vrsto spremenljivke. Spremenljivka lahko vsebuje samo vrednosti glede na vrsto, ki jo določimo. V primeru, da se določene vrednosti v programu ponavljajo, uporabimo spremenljivko za konstanto;
- sestavljeni tipi. Uporabljajo se polja, strukture in predmeti. Polja so namenjena za shranjevanje niza podatkov istega tipa v eno samo spremenljivko in so lahko statična ali dinamična. V strukturo lahko shranjujemo podatke različnih tipov;
- podprogrami. Poznamo procedure in funkcije, ki se uporabljajo za večkratno izvajanje programske kode. Parametri podprograma so spremenljivke, ki vplivajo na vrstni red izvajanja stavkov programske kode. Klic podprograma izvedemo tako, da zapišemo njegovo ime in vrednosti parametrov, kot so zapisane v njegovi glavi;
- sklicevanja. Uporabimo jih takrat, ko želimo dostopati do določenega elementa v programu. Poznamo posredno in neposredno sklicevanje;
- nadzor izvajanja. Uporabljajo se odločitveni stavki if, case in zanke za ponavljanje programskih stavkov. Poznamo zanke, ki se izvajajo, dokler velja pogoj, in zanke z vnaprej določenim številom ponovitev.

1.3.2 Spletni strežnik

Na spletni strani PCMAG (2016) je navedeno, da je spletni strežnik (angl. *Web Server*) računalnik, na katerem deluje spletna stran. Z uporabo protokola Hypertext Transfer Protocol (v nadaljevanju HTTP) spletni strežnik prenaša spletne strani (angl. *Web Pages*) do odjemalcev in druge podatkovne datoteke. Spletni strežnik vključuje strojno opremo, operacijski sistem, programsko opremo spletnega strežnika, protokol Transmission Control Protocol/Internet Protocol (TCP/IP) in vsebino spletnega mesta.

Na spletni strani podjetja TechTarget (b. l.) je navedeno, da je spletni strežnik program, ki uporablja protokol HTTP za prenos datotek spletnih strani do uporabnikov na osnovi zahtev. Zahteve spletnemu strežniku posredujejo HTTP-računalniki uporabnikov. Podobno velja za namenske računalnike in ostale naprave.

Spletna stran podjetja Tutorialspoint (2016) navaja, da je vsak računalnik, na katerem deluje spletna stran, spletni strežnik. Spletni strežnik, ki je povezan v internet, ima vedno naslov, sestavljen iz niza štirih števil med 0 in 255, ločen s pikami, na primer 10.0.190.61. Ko registriramo spletni naslov oziroma ime domene, je treba navesti naslov Internet Protocol (v nadaljevanju IP), ki bo gostil spletno stran. Obstaja kar nekaj spletnih strežnikov: Apache, Internet Information Services (v nadaljevanju IIS), lighttpd, Sun Java Microsystem, Jigsaw, Iplanet, WebSphere, nginx, Google Web Server (v nadaljevanju GWS) itd.

Spletni strežnik Apache je najbolj priljubljen spletni strežnik na svetu in je odprtokodna programska oprema. Mogoče ga je namestiti skoraj na vsak operacijski sistem: Windows, Linux, Unix, Mac idr. Danes v svetu 30 % spletnih strežnikov uporablja spletni strežnik Apache.

Spletni strežnik IIS je visoko zmogljiv Microsoftov strežnik in je integriran v operacijski sistem Windows, zato ga je zelo enostavno uporabljati. Različne različice spletnega strežnika IIS delujejo na različnih operacijskih sistemih Windows.

Lighttpd je odprtokodni spletni strežnik, ki je namenjen predvsem za operacijske sisteme Unix. Strežnik je hiter, varen in porabi malo procesorske moči računalnika. Lahko deluje tudi na operacijskih sistemih Windows, Mac, Linux in Solaris.

Spletni strežnik Sun Java Microsystem je primeren za srednje in velike spletne strani. Čeprav je strežnik brezplačen in odprtokodni, deluje na operacijskih sistemih Windows, Linux in Unix.

Za delovanje informacijskega sistema za vodenje o evidenc o javnih naročilih male vrednosti sem uporabil spletni strežnik IIS različice 7.0. in je nameščen na strežnik Windows Server 2008.

Ogrodje .NET. Pred razvojem ogrodja .NET so morali programerji napisati ogromno ponavljajoče se kode, npr. za odpiranje datoteke, branje in prikaz vsebine datoteke. Pri tem so uporabljali določene pomožne funkcije, ki pa so porabile preveč spomina, niso bile primerne z vidika varnosti in niso bile podprte za različne platforme. Rešitev je bila v ustvaritvi virtualnega okolja na strežnikih, kjer aplikacije delujejo kot gostitelji in ponujajo najrazličnejše storitve. Vse te storitve ponuja ogrodje .NET. Ogrodje .NET je Microsoftovo okolje za razvoj spletnih in mobilnih aplikacij Windows. Razvijalcem omogoča razvoj izredno učinkovitih in hitrih aplikacij (Moore, 2010, str. 17–18).

Povzeto po spletni strani Components of .Net Framework (b.l.) predstavljamo glavne module ogrodja .NET:

- Izvajalnik kode Common Language Runtime (v nadaljevanju CLR) je namenjen delovanju vseh programov .NET. Programerjem ni treba skrbeti za upravljanje pomnilnika in upravljanje niti, saj to zagotavlja izvajanje kode CLR. Prevajalniki jezikov najprej prevedejo program v jezik Microsoft Intermediate Language (MSIL) in nato v izvorni jezik z izvajalnikom kode CLR.
- Knjižnice razredov ogrodja FCL je skupno ime za knjižnice, ki se uporabljajo v ogrodju .NET. Knjižnice vsebujejo veliko vnaprej definiranih funkcij in so namenjene za aplikacije Windows, konzolne aplikacije, spletne aplikacije, XML-servise in servise Windows. Torej programerji te knjižnice vključijo v svojo programsko kodo in uporabijo svoje vnaprej definirane metode in lastnosti za razvoj kompleksnih funkcij, kot so branje in pisanje datoteke, grafično upodabljanje, interakcija z bazo ipd.
- Skupina tipov Common Type System (v nadaljevanju CTS) opisuje podatkovni tipe, ki se uporabljajo v različnih programskih jezikih .NET. CTS zagotavlja povezljivost objektov, ki so napisani v različnih programskih jezikih .NET. Podatkovne tipe CTS je treba prevesti na osnovnem nivoju ogrodja .NET. Poznamo dve skupini podatkovnih tipov CTS: spremenljivke, ki vsebujejo vrednosti, ki so locirane v skupini ali v strukturi, ter referenčne spremenljivke, ki se sklicujejo na pomnilniški naslov (kazalec na spremenljivko).
- Specifikacija jezika Common Language Specification (v nadaljevanju CLS) je podnabor skupine tipov CTS in določa sklop pravil, ki jih je treba upoštevati pri vseh programskih jezikih, ki podpirajo CLR. Pravila zagotavljajo, da programi pravilno delujejo v okolju .NET. Nekatera izmed njih so: zastopstvo besedilnih nizov, notranja predstavitev vrednotenj, opredelitev statičnih članov v podskupini CTS itd.

Active Server Pages (v nadaljevanju ASP) ASP.net. Okolje ASP.net predstavlja del ogrodja .NET in se uporablja za izdelavo dinamičnih spletnih strani in servisov. Programe lahko pišemo v katerem koli programskem jeziku, ki podpira izvajalnik kode CLR v ogrodju .NET. Najpogosteje uporabljena programska jezika za aplikacije ASP.net sta Visual Basic in C# (Microsoft, 2016). Poznamo tri tipe okolja ASP.net:

- ASP.NET Web Forms,
- Asp.net Model View Controller (v nadaljevanju MVC),
- in Asp.net Web Pages.

Okolje Asp.net Web Forms je namenjeno razvijalcem, ki nimajo veliko strokovnega znanja o programskih jezikih HTML in JavaScript in lahko prav tako izdelajo spletno aplikacijo. Bolj izkušeni razvijalci imajo raje standardne kontrolnike, ki so značilni za aplikacije Windows, ker so bolj seznanjeni s tradicionalnim razvojem odjemalca v programskem jeziku Visual Basic ali C#. Prednosti uporabe modela Web Forms so:

- omogoča programiranje dogodkov enako, kot bi programirali aplikacijo odjemalca oziroma aplikacije Windows;
- strežnik nadzora, ki nadomesti HTML, prilagodimo z nastavitvijo lastnosti in slogov;
- nudi bogat izbor nadzora za dostop in prikaz podatkov;
- možna je enostavna uvrstitev aplikacije in podatkov med zahteve HTTP za programerje, katerim so bližje aplikacije Windows, in na ta način se naučijo tudi, kako ustvariti spletne aplikacije.

Web Forms uporabljajo predvsem manjše skupine spletnih programerjev, ki želijo izkoristiti njihove prednosti in velik nabor komponent za hiter razvoj aplikacij. Na splošno velja, da tako programiranje zahteva manj napora kot programiranje v okolju MVC. Komponente in različni kontrolniki so integrirani, zato aplikacije vsebujejo manj kode kot aplikacije MVC. Ne nazadnje aplikacije Web Forms niso samo enostavne aplikacije, ampak so velikokrat zelo zahtevne poslovne aplikacije.

Aplikacije in kontrole Web Forms se samodejno generirajo in pošljejo brskalniku, zato nimamo podrobnega nadzora nad jezikom HTML, kar drugi modeli ASP.NET omogočajo. Prav tako take aplikacije skrijejo lastnosti obnašanja aplikacij HTML in HTTP.

Programerji MVC prisegajo na testno voden razvoj aplikacij. To okolje omogoča boljše ločevanje poslovne logike spletne aplikacije od predstavitvenega nivoja aplikacije. Aplikacije so razdeljene v tri sklope: model, vpogledi in kontrolniki. Na ta način se lažje upravlja kompleksnost v večjih aplikacijah. Za razvoj aplikacij MVC so lahko programerji razdeljeni v več skupin. Razvijalci delujejo na primer na poslovni logiki, oblikovalci pa na področju oblikovanja in označevanja.

Z uporabo okolja MVC programerji delajo neposredno s HTML in HTTP. Programerjem omogoča popoln nadzor nad kontrolniki v aplikaciji in popoln nadzor nad delovanjem v spletnem okolju. MVC je bil zasnovan tako, da je razširljiv in programerjem zagotavlja popolno okolje za potrebe aplikacij, ki jih želijo razviti. Izvorna koda MVC je na voljo pod licenco Open Systems Interconnection Model (v nadaljevanju OSI).

Okolje ASP.net Web Pages je namenjeno razvijalcem, ki želijo izdelati preproste spletne aplikacije, podobno kot Personal Home Page Tools (v nadaljevanju PHP). Spletne strani se ustvarijo kot strani HTML in se jih namesti na strežnik, kar omogoča dinamičen nadzor na njihimi. Programerji običajno nimajo veliko izkušenj s programiranjem, dobro pa poznajo jezik HTML. To so običajno študenti in jim je ta način programiranja spletnih strani najljubši, zato najlažje začnejo z uporabo ogrodja .NET. Podobno kot pri okolju Web Forms velja, da so razvijalci usmerjeni k hitremu razvoju aplikacij. Okolje Web Pages vsebuje določene komponente, ki so nekakšne predloge, programerji pa na ta način vpisujejo samo nekaj vrstic kode. Najpogostejše predloge se uporabljajo za prikaz podatkov, povezovanje različnih spletnih strani (npr. Facebook), dodajanje zemljevidov, kart ipd.

Okolje Web Pages omogoča enostavnejši pristop kot okolje Web Forms. Uporabljajo se HTML server-side datoteke in visual basic HTML datoteke, ki so po strukturi zelo podobne datotekam HTML, le da vsebujejo določene komponente okolja ASP.net. Zato jim je zelo enostavno dodajati funkcionalnosti s programskim jezikom JavaScript in jQuery (Microsoft, 2016).

Informacijski sistem za vodenje evidenc o javnih naročilih male vrednosti po evidenčnem postopku je razvit v okolju ASP.net Web Forms s programskih jezikom Visual Basic.

1.3.3 Strežnik SQL Server 2012

Microsoft SQL Server 2012 je sistem za upravljanje baz podatkov (RDMS) in je sestavljen iz več komponent. Najpomembnejša komponenta strežnika SQL Server 2012 se imenuje Database Engine in jo sestavlja pogon za shranjevanje podatkov. Na strežnik lahko podatke zapisujemo in jih beremo. Strežnik SQL deluje kot inštanca servisa na strežniku. Na strežniku lahko deluje več inštanc strežnika SQL. Ko se na strežnik povežemo, transakcija SQL pošlje zahtevo inštanci strežnika SQL, ta pa vrne podatke uporabniku (LeBlanc, 2013, str. 3–5).

Poznamo več različic strežnika SQL: Enterprise, Standard, Web, Developer in Express. Podatkovna baza za javna naročila male vrednosti po evidenčnem postopku je nameščena na strežniku MS SQL Server 2012 Standard.

Ostale komponente, ki delujejo v okviru strežnika, so še Analysis Services, Integration Services in Master Data Services. Pomembno orodje, ki je namenjeno načrtovanju, upravljanju in vzdrževanju podatkovnih baz strežnika SQL, se imenuje SQL Management Studio.

Z orodjem SQL Management Studio lahko konfiguriramo strežnik SQL, nastavimo replikacijo, kreiramo skupine SQL in uporabnike, kreiramo podatkovne baze, ustvarimo tabele, poglede, procedure itd. Orodje SQL Management Studio je zelo podobno večini Microsoftovih izdelkov in je sestavljeno iz glavnega menija, standardne orodne vrstice in večnamenskega okna (angl. *Object Explorer*). V večnamenskem oknu lahko pišemo različne skripte SQL s pomočjo funkcije Inetllisense, ki uporabniku ponuja predloge za pisanje skript in s podčrtavanjem opozarja na napake v sintaksi skripte. Podatkovna baza za javna naročila male vrednosti po evidenčnem postopku je bila ustvarjena z orodjem SQL Management Studio.

2 ANALIZA OBSTOJEČEGA STANJA

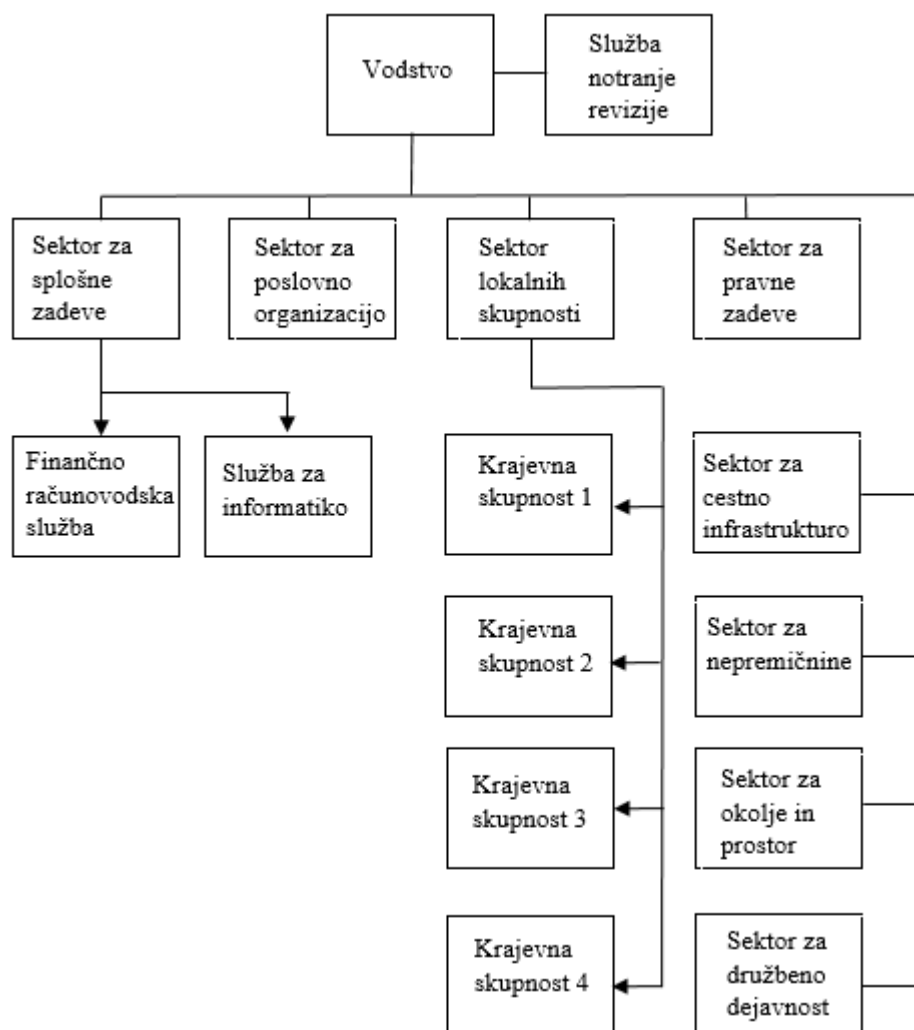
2.1 Predstavitev izbrane organizacije

Organizacija, ki jo obravnavam v magistrski nalogi, je lahko kateri koli javni organ, zavod, javna agencija ali lokalna skupnost. Za lažjo predstavo in predstavitev konkretnega problema sem izbral organizacijo, za katero je značilna organiziranost, kot velja v večini slovenskih občin (v nadaljevanju izbrana organizacija). Izbrana organizacija je samoupravna lokalna skupnost, ustanovljena z zakonom, in je pravna oseba javnega prava, ki ima pravico posredovati, pridobivati in razpolagati z vsemi vrstami premoženja. Izbrana organizacija opravlja naloge:

- upravlja lokalne zadeve javnega pomena,
- gospodari z nepremičninami,
- sprejema prostorske akte,
- pospešuje razvoj gospodarskih dejavnosti, obrti, turizma in kmetijstva,
- sodeluje z gospodarskimi subjekti ter v okviru interesov in nalog občine pomaga gospodarskim subjektom pri razreševanju gospodarskih problemov,
- ustvarja pogoje za gradnjo stanovanj in povečanje najemnega socialnega sklada stanovanj,
- ureja, upravlja in skrbi za lokalne javne službe,
- zagotavlja in pospešuje vzgojno-izobraževalno in zdravstveno dejavnost,
- pospešuje službe socialnega skrbstva za socialno ogrožene, invalide in ostarele, predšolskega varstva, osnovnega varstva otrok in družine,
- pospešuje raziskovalno, kulturno in društveno dejavnost ter razvoj športa in rekreacije,
- zagotavlja varstvo okolja in učinkovito rabo energije in drugo.

Organizacijo izbrane organizacije določa statut, ki ga sprejme svet kot organ upravljanja. Izbrana organizacija izvaja naloge na sedežu in območnih izpostavah. Naloge se v izbrani organizaciji izvajajo v okviru osmih sektorjev, treh služb in štirih lokalnih izpostav, kar prikazuje Slika 6. Ena služba je samostojna, dve pa sodita v okvir sektorjev. Naloge so torej opredeljene v okviru notranjih organizacijskih enot: vodstvo organizacije, sektor za splošne zadeve, sektor lokalnih skupnosti, sektor za poslovno organizacijo, sektor za pravne zadeve, sektor za cestno infrastrukturo, sektor za nepremičnine, sektor za okolje in prostor, sektor za družbene dejavnosti, finančno-računovodska služba, služba notranje revizije in služba za informacijsko tehnologijo. Za delovanje izbrane organizacije je odgovoren župan. Izbrano organizacijo upravljata svet in nadzorni organ.

Slika 6: Notranja organizacija izbrane organizacije



Vir: Izbrana organizacija, Poročilo o delu in zaključni račun za leto 2014, 2015a, str. 6.

2.2 Analiza temeljnih poslovnih procesov

Pri analizi poslovnega procesa evidenčnega postopka sem proučil:

- organizacijsko strukturo izbrane organizacije,
- glavne poslovne procese izbrane organizacije,
- vso dokumentacijo, ki jo je treba upoštevati pri izvajanju procesa,
- zaporedje izvajanja nalog v procesu,
- število zaposlenih in vloge zaposlenih v procesu,
- informacije zaposlenih, pridobljene na osnovi razgovorov.

Evidenčni postopek (Slika 7) oddaje naročila je sestavljen iz procesov:

- začetek postopka,
- priprava predloga,
- pregled predloga,
- priprava naročilnice/pogodbe,
- potrditev naročilnice/pogodbe,
- izdaja naročilnice/pogodbe,
- oddaja naročila,
- knjiženje računa,
- zaključitev postopka.

Vodenje postopka javnega naročanja je organizirano hierarhično in poteka linearno. Pooblastila v postopku naročanja si izmenjujejo:

- vodja naročila (ustvari nov postopek),
- vodja naročila (pripravi naročilo),
- vodja organizacijske enote/tajništvo (potrdi naročilo),
- tajništvo (pripravi naročilnico),
- pravni sektor (pripravi pogodbo),
- tajništvo (potrdi naročilnico/pogodbo),
- tajništvo/vodja naročila (odda naročilo),
- finančna služba (knjiži račun),
- vodja naročila (zaključi postopek).

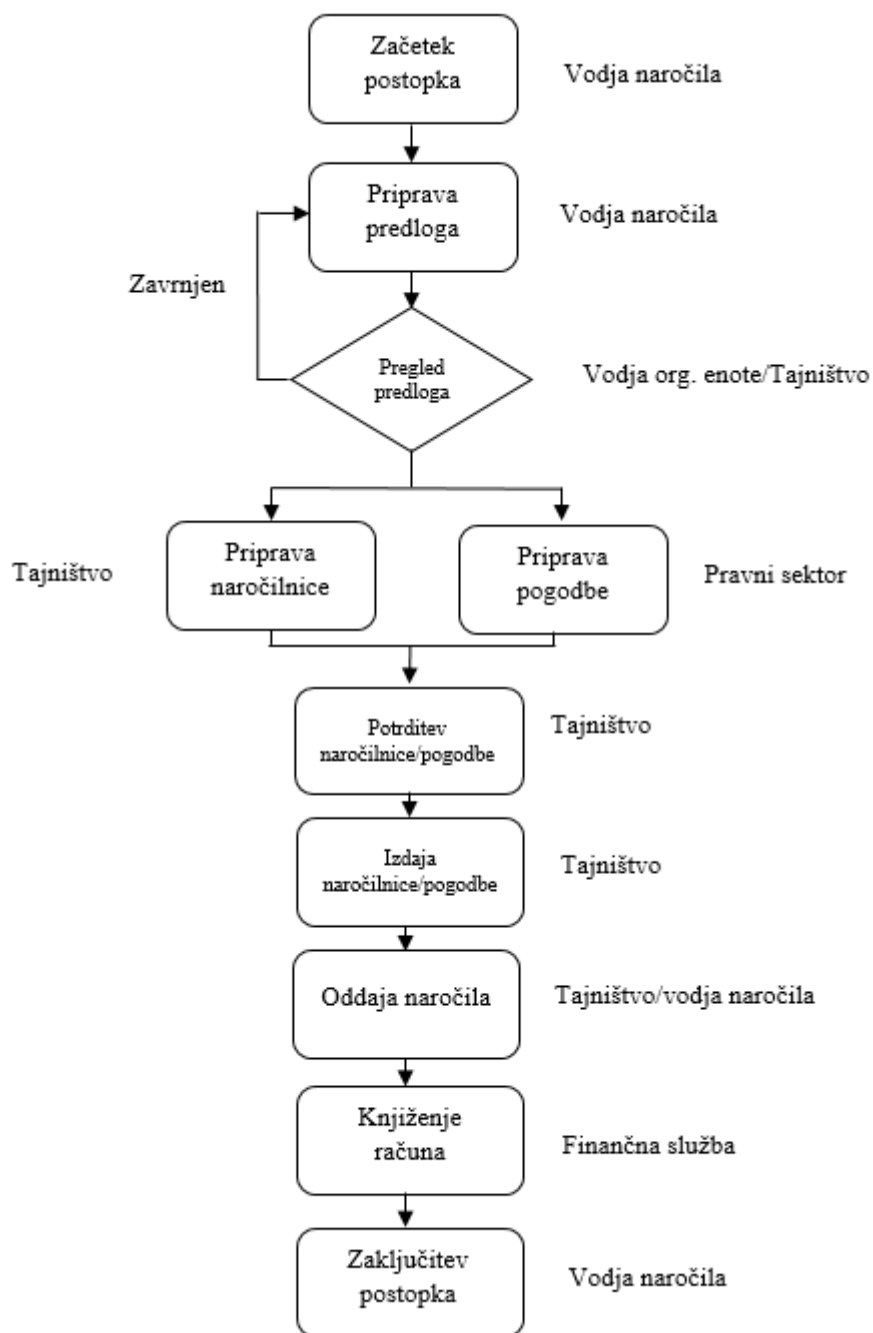
Začetek postopka. Vodja naročila pripravi vso potrebno dokumentacijo za začetek postopka oddaje naročila po evidenčnem postopku.

Priprava predloga. Vodja naročila pripravi predlog za izdajo naročilnice/pogodbe, ki jo da v podpis vodji organizacijske enote. Pri oblikovanju predloga za izdajo naročilnice/pogodbe je treba utemeljiti najugodnejšega ponudnika. Najugodnejša ponudba se izbere izmed treh ponudb ponudnikov. S tem, ko vodja naročila potrdi predlog, ta preide v naslednjo fazo potrjevanja (tajništvo).

Pregled predloga. Predlog se pošlje v pregled tajništvu. V primeru, da je predlog nepopoln, se ga vrne v vnovičen popravek vodji naročila. Predlog s podpisom potrdi predstavnik organizacije. Predlog se lahko tudi prekliče in zaključi.

Priprava naročilnice/pogodbe. Naročilnico pripravi tajništvo. Pri tem je treba na naročilnico vpisati vse potrebne podatke, kot so opredeljeni že v okviru predloga naročila. Naročilnico se posreduje v podpis predstavniku organizacije. Pogodbo pripravi pravni sektor in jo pošlje v podpis predstavniku organizacije.

Slika 7: Poenostavljena shema evidenčnega postopka oddaje naročila



Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 3.

Izdaja naročilnice/pogodbe. Zakoniti zastopnik podpiše naročilnico ali pogodbo.

Oddaja naročila. Naročilo se odda v tajništvu ali pa jo odda vodja naročila.

Knjiženje računa. V finančno-računovodski službi se knjiži prejeti račun. Za eno naročilnico/pogodbo je možno izdati več računov, vse dokler vodja naročila ne zaključi posameznega naročila.

Zaključitev postopka. Vodja naročila zaključi postopek naročila ob zaključenem knjiženju računa. V primeru, da naročilo ni izpeljano, ga prekliče. Vso dokumentacijo naročila se ob zaključku preda v arhiv.

2.3 Analiza informacijskega sistema

V informacijskem sistemu izbrane organizacije je že razvita vrsta programov in rešitev, ki podpirajo poslovanje. V izbrani organizaciji so v rabi naslednje kategorije informacijskih rešitev:

- infrastrukturne rešitve: informacijska školjka sistema (npr. internet/intranet, dokumentni sistem, administracija uporabniških računov/user management/);
- pregledovalniki: vpogled v mozaik podatkov, ki so v povezavi s poslovanjem (npr. pregledovalnik podatkov o zemljiščih, grafični pregledovalniki zemljišč);
- podpora posameznim postopkom in poslovnim procesom – programi, ki so namenjeni vodenju posameznih postopkov in poslovnih procesov;
- analitični programi in aplikacije – programi za izdelavo zahtevnih poročil in analiz, ki nastopajo predvsem kot interakcija programov za podporo posameznih postopkov;
- integrirane programske rešitve – programi, ki se v celoti ali s posameznimi funkcionalnostmi vključujejo v enega ali več drugih programov.

Za informacijski sistem izbrane organizacije je značilna visoka stopnja prepletanja in interakcije, zato ni ostrih ločnic med navedenimi informacijskimi rešitvami. Obstoječi informacijski sistem za oddajo naročila po evidenčnem postopku ni ustrezen in ni integriran v informacijski sistem izbrane organizacije.

2.3.1 Postopek evidentiranja in kroženja dokumentov

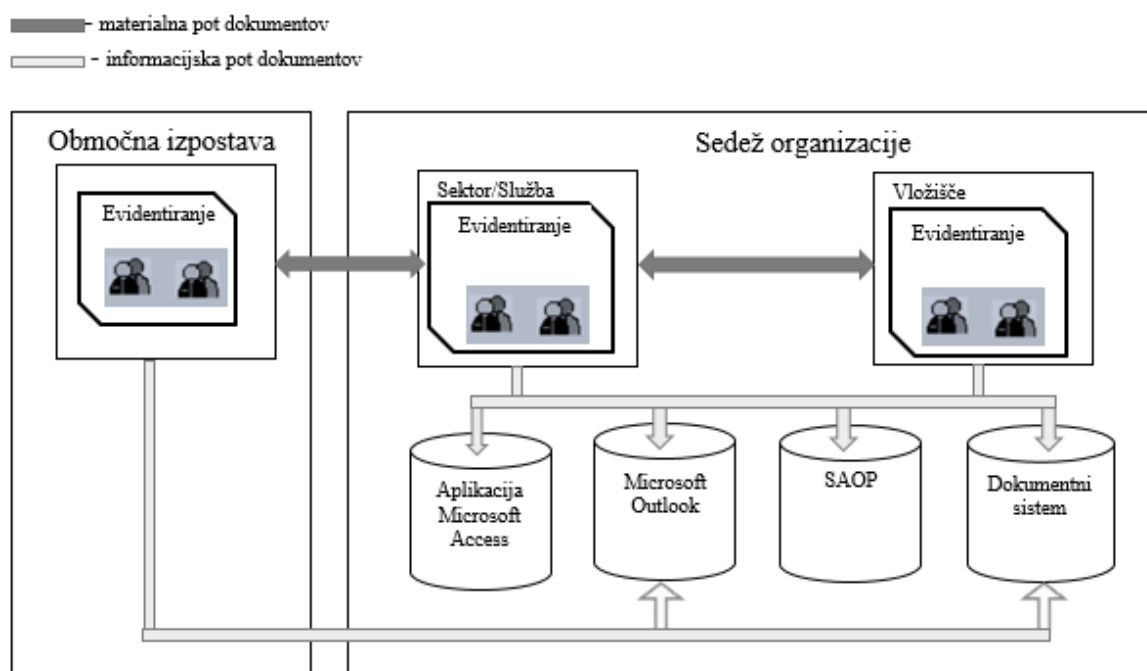
V evidenčnem postopku oddaje naročila se uporabljajo naslednji pomembnejši dokumenti:

- predlog za izdajo naročila,
- naročilnica/pogodba,
- račun.

Vodja naročila ustvari predlog za izdajo naročila v aplikaciji Microsoft Access. V primeru, da je zaposleni zaposlen v območni izpostavi, se v tajništvo z elektronsko pošto pošljejo vsi podatki za pripravo predloga naročila. Vodja naročila v tem primeru ni zaposleni v območni

izpostavi, ampak zaposleni v tajništvu. Iz aplikacije Microsoft Access se natisne dokument »predlog za izdajo naročilnice/pogodbe«. Predlog za izdajo naročilnice/pogodbe se posreduje v podpis vodji organizacijske enote. V primeru, da se predlog ne odobri, se znova posreduje v popravek vodji naročila. Če je predlog odobren, se posreduje v tajništvo, kjer se znova vnesejo podatki o odobritvi predloga v aplikacijo Microsoft Access. V tajništvu se pripravi naročilnica ali pogodba, ki se skupaj s predlogom pošlje v podpis zastopniku organizacije. Naročilnico se napiše ročno, pogodbo se pripravi z orodjem Microsoft Word in natisne. Zastopnik organizacije lahko predlog s podpisom odobri ali zavrne. V tajništvu znova vnesejo nekatere podatke o naročilu v aplikacijo Microsoft Access. Dokumenti (predlog za izdajo naročilnice, naročilnica/pogodba) se posredujejo v vložišče, kjer se vpišejo v dokumentni sistem in pošljejo dobavitelju naročila. Ob prejemu blaga ali zaključku storitve dobavitelj pošlje račun, ki se ga posreduje v finančno-računovodsko službo. Račun se knjiži v aplikacijo Microsoft Access in računovodsko poslovno aplikacijo SAOP (v nadaljevanju SAOP). Postopek evidentiranja in kroženja dokumentov prikazuje Slika 8.

Slika 8: Materialna in informacijska pot dokumentov



Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 4.

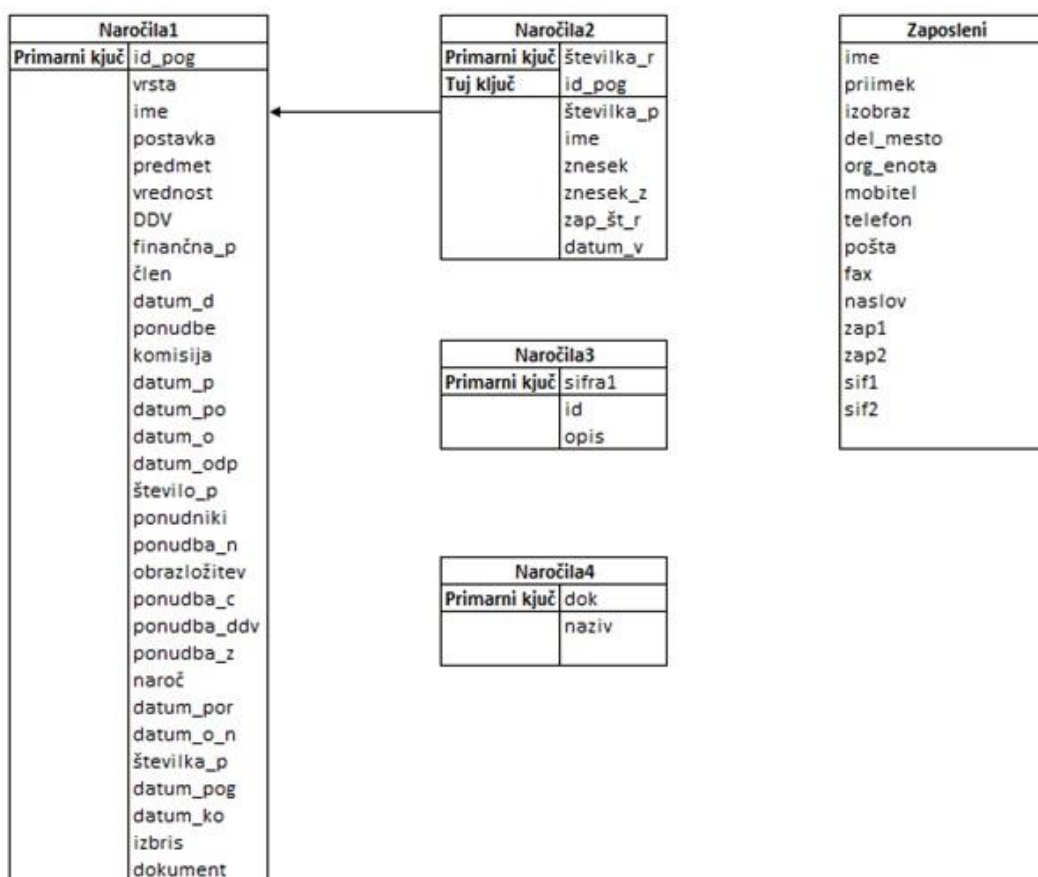
2.4 Podatkovni model

Podatkovna baza je bila izdelana z orodjem Microsoft Access in je vzpostavljena na Microsoft SQL 2000. Pri nekaterih tabelah so bili definirani primarni ključi. Opredeljene so relacije med tabelami. Podatkovni model (Slika 9) je sestavljen iz tabel:

- Naročila1 (vsebuje podatke o naročniku in ponudnikih),
- Naročila2 (vsebuje podatke o računih),
- Naročila3 (vsebuje podatke o vrstah izvedb naročila),
- Naročila4 (vsebuje podatke o vrstah pogodbe).

Podatkovna baza je povezana s podatkovno bazo zaposlenih prek vpogleda naročila_zaposleni.

Slika 9: Podatkovni model



Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 5.

V nadaljevanju so opisani posamezni atributi podatkovnih tabel. Podatkovna tabela Naročila1 vsebuje naslednje attribute:

- Id_pog: zaporedna številka predloga naročila. Številka predloga naročila Id je v tabeli Naročila1 primarni ključ. Povezan je na tuj ključ id_pog v tabeli račun Naročila2.
- Vrsta: vrsta predloga naročila: 1 – z naročilnico, 2 – z letno pogodbo, 3 – s pogodbo.
- Ime: uporabniško ime zaposlenega, ki je ustvaril predlog.

- Postavka: vrsta postavke iz tabele jnmv_sif01. 0 – ni opredeljena, 10 – blago, 20 – storitve in 30 – gradnje.
- Predmet: predmet naročila.
- Vrednost: ocenjena vrednost naročila brez DDV.
- DDV: ocenjena vrednost naročila z DDV.
- Finančna_p: finančna postavka naročila.
- Člen: številka člena iz pogodbe.
- Datum_d: datum predvidene dobave blaga.
- Ponudbe: obrazložitev izbora naročila.
- Komisija: ugotovitve komisije.
- Datum_p: datum kreiranja predloga za izdajanje naročilnice.
- Datum_po: datum potrditve predloga za izdajo naročilnice.
- Datum_o: datum objave ponudbe.
- Datum_odp: datum odpiranja ponudbe.
- Število_p: število vseh ponudb.
- Ponudniki: nazivi in naslovi ponudnikov.
- Ponudba_n: naziv najugodnejšega ponudnika.
- Obrazložitev: obrazložitev najboljšega ponudnika.
- Ponudba_c: najugodnejša ponudba v EUR.
- Ponudba_ddv: znesek DDV najugodnejše ponudbe.
- Ponudba_z: obrazložitev zavrnitve ponudbe.
- Naroč: naročilo je izvedeno/ni izvedeno.
- Datum_por: datum poročila o naročilu.
- Datum_o_n: datum odobritve naročila.
- Stevilka_p: številka predloga za izdajo naročila.
- Datum_pog: datum pogodbe.
- Datum_ko: datum zaključitve postopka naročila.
- Izbris: stanje dokumenta.
- Dokument: prosti atribut.

Podatkovna tabela Naročila2 vsebuje naslednje attribute:

- Id_pog: zaporedna številka predloga naročila.
- Ime: uporabniško ime zaposlenega, ki knjiži račun.
- Številka_r: številka računa.
- Znesek: znesek računa z DDV.
- Znesek_z: znesek računa brez DDV.
- Zap_št_r: zaporedna številka računa.
- Datum_v: datum knjiženja računa.

Podatkovna tabela Naročila3 vsebuje naslednje attribute:

- Id_pog: zaporedna številka predloga naročila
- Šifra1: šifra vrste naročila.
- Opis: opis vrste naročila.

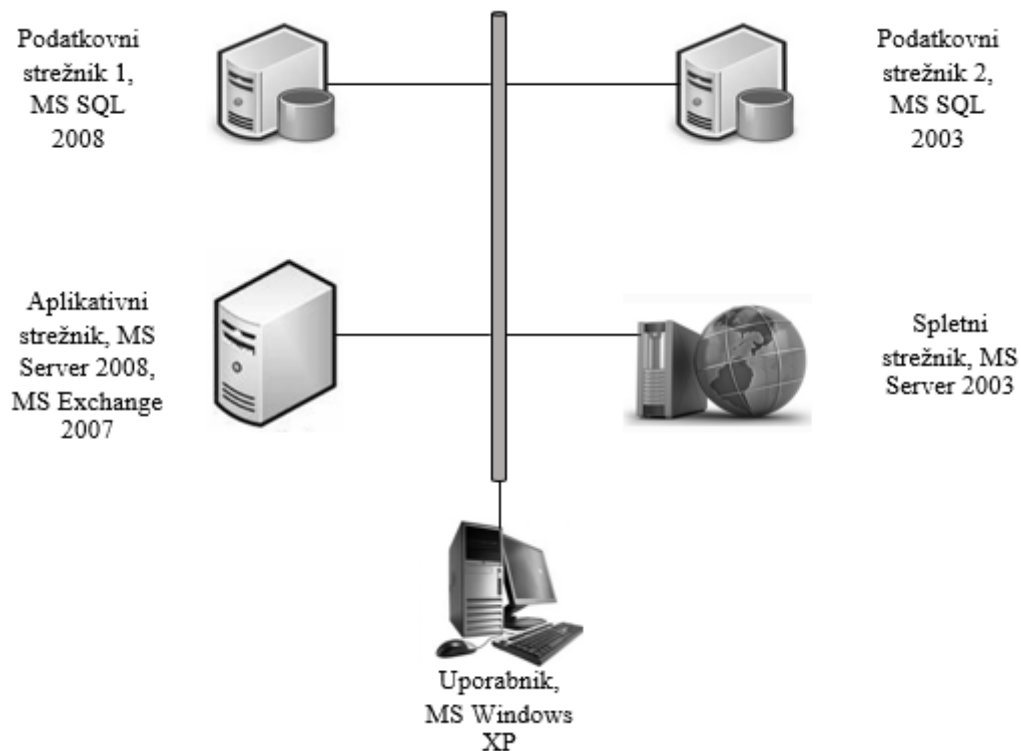
Podatkovna tabela Naročila4 vsebuje naslednje attribute:

- Dok: tip dokumenta.
- Naziv: naziv dokumenta je lahko naročilnica ali pogodba.

2.5 Programska, komunikacijska in strojna oprema

Strežniška infrastruktura izbrane organizacije deluje na Microsoftovi platformi Windows Server 2003 in Windows Server 2008 in je prikazana na Sliki 10.

Slika 10: Strežniška infrastruktura izbrane organizacije



Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 6.

Virtualno okolje je vzpostavljeno na strežnikih VMWare različic 5.1 in 4.0. Vanj so v gručo (angl. *Cluster*) povezani štirje strežniki. V prvi del gruče sta vključena strežnika SQL v

produkciji, v drugi del gruče pa sta vključena testna strežnika SQL. V virtualnem okolju je na voljo 12,5 TB prostora, od katerega je uporabljenega ca. 8,5 TB. Uporabljata se dve diskovni polji: International Business Machines Corporation (IBM) v velikosti 14 TB in Hewlett-Packard (v nadaljevanju HP) v velikosti 10 TB.

Na strežniški platformi se uporabljata operacijska sistema Server 2003 in Sever 2008. Za elektronsko pošto se uporablja Microsoft Exchange 2007, ki je postavljen na dveh strežnikih v gruči. Operacijski sistem na delovnih postajah je Windows XP.

Lokalno omrežje je sestavljeno iz dveh mrežnih stikal Cisco in štirih mrežnih stikal HP, prek katerih so povezani strežniki, računalniki in mrežni tiskalniki. Omrežje je prek usmerjevalnika povezano v državno komunikacijsko omrežje HKOM.

2.6 Kritična analiza starega sistema

Prvotna aplikacija je bila zasnovana z orodjem Microsoft Access in je bila izdelana zgolj kot evidenčna aplikacija za javna naročila malih vrednosti, ki jo je uporabljal le en zaposleni v tajništvu. Aplikacija do leta 2014 ni bila deležna nikakršnih popravkov. V izbrani organizaciji se je v tem času spremenila organizacijska struktura in s tem posledično tudi poslovni procesi. S spremembo ZJN- 2 o javnem naročanju malih vrednosti v letu 2014 se je pojavila potreba po izdelavi nove aplikacije, saj je bila aplikacija tehnološko in vsebinsko povsem neustrezna in neprimerna za uporabo. ZJN-2 namreč določa naročniku voditi evidenco za javna naročila, ki ne presegajo mejne vrednosti za objavo v skladu z določili zakona. Za ta namen sem razvil nov informacijski sistem v skladu z veljavno zakonodajo ZJN-2 in v skladu s sprejetim internim aktom. Pomanjkljivosti obstoječe aplikacije bi lahko strnil v naslednja področja:

- uporabniki,
- povezljivost,
- tehnična zastarelost,
- revizijska sled,
- podatkovni model,
- analiza podatkov.

2.6.1 Uporabniki

Aplikacijo uporablja le en zaposleni v tajništvu. Ostali zaposleni niso uporabniki aplikacije. Pravice uporabnika niso ustrezno urejene na nivoju domene in na nivoju uporabniških vlog. Podatki uporabnikov aplikacije se ne usklajujejo iz aktivnega imenika, ampak ločeno v podatkovni tabeli. Aplikacija ne preverja identitete uporabnika prek aktivnega imenika.

2.6.2 Povezljivost

Postopek oddaje naročila ni informatiziran v celoten proces oddaje naročila po evidenčnem postopku. Aplikacija ni povezljiva z ostalimi aplikacijami v organizaciji. Uporabnik ustvarja dokumente v ločenih aplikacijah namesto v enotnem dokumentnem sistemu. Podatki, ki se vpisujejo v aplikacijo, se ne prepisujejo avtomatsko v SAOP. Finančne stranke se ne vpisujejo iz SAOP, ampak se vpisujejo v aplikacijo, kar predstavlja težave zaradi podvojenosti podatkov. Aplikacija ni povezana z zunanjimi podatkovnimi bazami, npr. s katastrom Geodetske uprave RS, kar bi lahko omogočilo tudi vpisovanje parcel v aplikacijo. Aplikacija ni ustrezno vključena v intranetni sistem izbrane organizacije.

2.6.3 Tehnična zastarelost

Aplikacija je zastarela tako z vidika razvoja kot tudi z vidika vključenosti v obstoječo arhitekturo informacijskega sistema.

2.6.4 Revizijska sled

Aplikacija ne vpisuje nikakršnih sprememb podatkov v primeru, ko uporabnik izvede določeno spremembo. Prav tako se ne zagotavlja ustrezno beleženje dogodkov v primeru, ko uporabnik izvede vpis ali izbris podatkov. Aplikacija ne omogoča beleženja pomembnih poizvedb o tabelah v podatkovni bazi.

2.6.5 Podatkovni model

Pri načrtovanju podatkovnega modela normalizacija podatkov ni bila ustrezno izvedena. Atributi v podatkovnem modelu nimajo logičnih imen. Tabele niso indeksirane, kar predstavlja težavo pri hitrosti izpisovanja podatkov v aplikaciji. Poslovna logika aplikacije ni vključena na nivoju podatkovne baze, procedure in vpogledi se ne uporabljajo. Podatkovna baza deluje na strežniku Microsoft Corporation (v nadaljevanju MS) SQL 2003 in ni združljiva s podatkovnim strežnikom MS SQL 2008.

2.6.6 Analiza podatkov

Aplikacija ne omogoča priprav najrazličnejših poizvedb, poročil ali izpisov podatkov. Poizvedbe je mogoče izvajati le na nivoju podatkovne baze. Uporabnik nima vsebinskega pregleda nad postopkom oddaje naročila.

3 IZVEDBA INFORMACIJSKEGA SISTEMA

Izdelan je bil informacijski sistem za javna naročila po evidenčnem postopku za izbrano organizacijo. Informacijski sistem podpira področja oddaje naročil, potrjevanja naročil, knjiženja računov ter analize podatkov in priprave poročil.

Za izdelavo informacijskega sistema sem uporabil tehnologijo .NET. Uporabil sem razvojno orodje MS Visual Studio 2012 Professional in programski jezik Visual Basic. Informacijski sistem ima večnivojsko arhitekturo, za katero so značilni elementi odjemalec, spletni strežnik, aplikacijski strežnik in podatkovni strežnik. Izdelan informacijski sistem deluje kot spletna aplikacija, ki deluje na odjemalcih MS Internet Explorer in Google Chrome. Aplikacijski strežnik je vzpostavljen na MS Windows Server 2008, na katerem sta nameščena IIS 7.0 in Microsoft Framework 3.5. Za temeljni strežnik se uporablja MS SQL Server 2008.

3.1 Sodelujoči pri razvoju informacijskega sistema

V izbrani organizaciji je bila za namen izdelave informacijskega sistema ustanovljena projektna skupina, ki je bila sestavljena iz nadzornikov projekta, vodje projekta in vsebinskih izvajalcev projekta. Poleg razvijalca informacijskega sistema sem zastopal tudi vlogo vodje projekta. Vloge nadzornikov projekta so:

- sodelujejo pri ključnih fazah projekta,
- zagotavljajo resurse za izvajanje projekta,
- potrjujejo faze projekta,
- odločajo o morebitnih spremembah projekta,
- potrjujejo dokumentacijo projekta in
- izvedejo prevzem projekta.

Vodja projekta ima odgovornost razpolagati z dodeljenimi viri v dogovorjenem času glede na stroške projekta. Odgovoren je za komunikacijo med vsemi udeleženci projektne skupine. Pomembnejše naloge vodje projekta so:

- izdelava terminskega plana projekta,
- pripravljanje rednih poročil za nadzornike projekta,
- podpisovanje raznih zapisnikov,
- organiziranje sestankov s sodelujočimi v projektu,
- vodenje dokumentacije projekta,
- spremljanje aktivnosti v projektu,
- priprava prevzemnega zapisnika.

Vsebinski izvajalci projekta so aktivno sodelovali pri predstavljanju obravnavanega področja v okviru sestankov, intervjujev ipd. Pomembnejše naloge vsebinskih izvajalcev projekta so:

- vsebinsko potrjevanje dokumentacije,
- vsebinsko sodelovanje pri aktivnostih v projektu,
- opredelitev potrebnih postopkov in kontrol,
- opredelitev morebitnih napak,
- opredelitev povezav z ostalimi informacijskimi sistemi,
- vsebinsko testiranje informacijskega sistema.

Nadzorniki projekta so se na zahtevo vodje projekta sestali vsaj enkrat na dva meseca oziroma po zaključitvi posameznih faz projekta. Redno so spremljali napredovanje projekta. V primeru, da projekt ni dosegal zastavljenih ciljev, je vodja projekta lahko predlagal popravek plana projekta. Vsebinski izvajalci projekta so se na zahtevo vodje projekta sestali dvakrat tedensko oziroma po potrebi in nenehno spremljali stanje projekta in v zvezi s tem podajali vsebinske zahteve za uspešno izvedbo projekta.

3.2 Terminski načrt razvoja informacijskega sistema

Terminski načrt razvoja je zajemal okvirni časovni plan in ovrednotenje stroškov za izdelavo informacijskega sistema.

3.2.1 Časovni plan

Časovni plan (Tabela 1) je bil izdelan glede na zmožnosti sodelovanja zaposlenih v izbrani organizaciji in glede na dejstvo, da sem v celoti sam izdelal informacijski sistem za javna naročila po evidenčnem postopku. Iz Tabele 1 je razvidno, da sem največ časa porabil za izvedbo projekta, in sicer skupaj 570 ur. Veliko časa sem porabil tudi za načrtovanje programske rešitve, saj je bilo potrebnih veliko usklajevalnih sestankov z zaposlenimi v izbrani organizaciji. Z načrtovanjem programske rešitve sem se ukvarjal več kot dva meseca. Za izdelavo programske rešitve sem porabil 440 ur. Sestavljena je bila iz faz, predstavljenih v Tabeli 2.

Tabela 1: Časovni plan za izgradnjo informacijskega sistema

Projekt: Izgradnja informacijskega sistema za javna naročila po evidenčnem postopku					
Zap. št.	Aktivnost	Faza razvoja	Začetek	Konec	Št. ur
1.	Analiza temeljnih poslovnih procesov	Analiza	2.3.2015	15.3.2015	40
2.	Analiza obstoječega informacijskega sistema		15.3.2015	27.3.2015	30
3.	Analiza podatkovnega modela		29.3.2015	4.4.2015	10
4.	Izdelava terminskega načrta razvoja	Načrtovanje	4.4.2015	12.4.2015	40
5.	Izdelava funkcijske specifikacije		12.4.2015	1.6.2015	160
6.	Uskladitev funkcijske specifikacije		1.6.2015	6.6.2015	20
7.	Izdelava baze in prepis podatkov	Izvedba	7.6.2015	15.6.2015	30
8.	Izdelava programske rešitve		15.6.2015	15.9.2015	440
9.	Testiranje rešitve		15.9.2015	20.9.2015	25
10.	Izdelava dokumentacije		20.9.2015	3.10.2015	40
11.	Usposabljanje uporabnikov		3.10.2015	15.10.2015	35
12.	Prehod v produkcijo	Vpeljava	15.10.2015	17.10.2015	10
13.	Vzdrževanje programske rešitve	Vzdrževanje	19.10.2015	19.10.2017	96
14.	Odpravljanje napak programske rešitve		19.10.2015	19.10.2017	48
				Št. ur skupaj:	1024

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (specifikacija dizajna), 2015e, str. 7

Tabela 2: Faze izdelave programske rešitve

Aktivnost: Izdelava programske rešitve		
Zap. št.	Aktivnost	Št. ur
1.	Vzpostavitev testnega okolja	16
2.	Pravice uporabnikov	16
3.	Urejanje šifrantov	40
4.	Urejanje registra strank	40
9.	Vnosne maske	136
10.	Vpis spisov in dokumentov	40
11.	Povezava z dokumentnim sistemom	56
12.	Analitične poizvede in poročila	96
Št. ur skupaj:		440

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (specifikacija dizajna), 2015e, str. 7.

Za vzdrževanje programske rešitve se predvidevajo štiri ure mesečno in za odpravljanje napak programske rešitve dve uri mesečno.

3.2.2 Finančni načrt

Finančni načrt je bil izdelan na osnovi dveh ovrednotenih določitev cene urne postavke za izdelavo programske rešitve. V prvem primeru predpostavljam, da je programska rešitev izdelana v okviru notranjega razvoja izbrane organizacije. Ceno urne postavke sem izračunal na osnovi vseh odhodkov, ki so bili namenjeni za zaposlene v izbrani organizaciji v letu 2014:

- stroški, namenjeni za plače – 1.854.648 EUR (stroški plač),
- ostali stroški zaposlenih – 141.034 EUR (ostali stroški),
- stroški izobraževanja zaposlenih – 3.979 EUR (stroški izobraževanj).

V stroške plač se šteje celoten strošek delodajalca, ki ga mora plačati zaposlenemu. Sem spadajo vsi prispevki, obračunani v plači, vključno s stroškom prehrane in prevoza na delo. Med ostale stroške se štejejo vsi stroški, ki nastanejo z nudenjem vseh osnovnih sredstev, ki so potrebni za delovanje zaposlenega. Sem spadajo stroški nabave računalniške opreme, pisarniškega materiala, najemnin, izobraževanja ipd.

Pri izračunu cene urne postavke je smotrno, da se upoštevajo tudi ostali stroški, ki z razvojem programske rešitve sicer niso povezani neposredno, so pa pomembni z vidika zagotavljanja vseh potrebnih sredstev za razvoj programske rešitve.

Za izračun cene urne postavke sem upošteval še:

- število zaposlenih v organizaciji – 48 (število zaposlenih),
- število učinkovitih delovnih dni v letu – 220 (število dni),
- število delovnih ur v dnevu – 8 (število ur).

Cena urne postavke v okviru notranjega razvoja se izračuna po enačbi (1).

$$\text{cena urne postavke (notranji razvoj)} = (\text{stroški plač} + \text{ostali stroški} + \text{stroški izobraževanj}) / (\text{število zaposlenih}) / \text{število dni} / \text{število ur} \quad (1)$$

Iz enačbe (1) izhaja, da znaša cena urne postavke za izbrano organizacijo v primeru notranjega razvoja približno 24 EUR.

Strošek izdelave programske rešitve za izbrano organizacijo v primeru notranjega razvoja, izračunamo na osnovi enačbe (2):

- število ur, ki ga potrebujemo za izdelavo programske rešitve iz Tabele 1 – 1024 ur,
- cene urne postavke v okviru notranjega razvoja pridobljene iz enačbe (1) – 24 EUR.

$$\text{Strošek izdelave programske rešitve} = \text{cena urne postavke (notranji razvoj)} * \text{število ur za izdelavo programske rešitve} \quad (2)$$

Strošek izdelave programske rešitve v primeru notranjega razvoja znaša 24.576 EUR. V Tabeli 3 so podrobneje prikazani stroški razvoja programske rešitve po posameznih aktivnostih razvoja programske rešitve.

Tabela 3: Strošek razvoja programske rešitve v primeru notranjega razvoja

Stroški razvoja programske rešitve - notranji razvoj				
Zap. št.	Aktivnost	Faza razvoja	Št. ur	Znesek (Eur)
1.	Analiza temeljnih poslovnih procesov	Analiza	40	960
2.	Analiza obstoječega informacijskega sistema		30	720
3.	Analiza podatkovnega modela		10	240
4.	Izdelava terminskega načrta razvoja	Načrtovanje	40	960
5.	Izdelava funkcijske specifikacije		160	3.840
6.	Uskladitev funkcijske specifikacije		20	480
7.	Izdelava baze in prepis podatkov	Izvedba	30	720
8.	Izdelava programske rešitve		440	10.560
9.	Testiranje rešitve		25	600
10.	Izdelava dokumentacije		40	960
11.	Usposabljanje uporabnikov		35	840
12.	Prehod v produkcijo	Vpeljava	10	240
13.	Vzdrževanje programske rešitve	Vzdrževanje	96	2.304
14.	Odpravljanje napak programske rešitve		48	1.152
			Znesek	24.576

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (specifikacija dizajna), 2015e, str. 10.

Iz tabele 3 je razvidno, da strošek razvoja za izdelavo programske rešitve znaša 24.576 EUR, vključno z dveletnim vzdrževanjem.

V drugem primeru predpostavljam, da je programska rešitev izdelana v okviru zunanjskega razvoja. Ceno urne postavke sem povzel po referenčnem ceniku storitev s področja informacijskih storitev. Cenik je objavljen na spletnem naslovu <https://www.gzs.si> in velja že od 23. 4. 2008.

V tabeli 4 so navedene cene urnih postavk glede na značilnosti dela oziroma zahtevnost del po referenčnem ceniku s področja informacijskih storitev. V cenah ni upoštevan DDV. Upoštevane so samo učinkovite ure, brez stroškov malice, vožnje ipd.

Ocenjujem, da znanja, ki so potrebna za razvoj programske rešitve, lahko uvrstim pod stopnjo 5 po referenčnem ceniku informacijskih storitev. Za izdelavo programske rešitve je namreč potrebnih veliko znanj s področja programiranja spletnih rešitev, načrtovanja podatkovnih baz, administracije različnih strežnikov, vodenja projektov ipd. Cena delovne ure v tem primeru znaša 95 EUR brez DDV.

Za bolj verodostojno določitev cene urne postavke v primeru zunanjskega razvoja, bi bilo smotrnejše izvesti analizo javnih naročil v primerih naročanja programskih rešitev s podobno težavnostjo. Analiza javnih naročil na področju informacijskih storitev ni predmet te naloge in bi predstavljala kar veliko dodatnega dela. Oceno za razvoj programske rešitve v primeru zunanjskega razvoja sem podal na podlagi izkušenj, ki sem si jih pridobil v zadnjih letih s sodelovanjem pri več javnih razpisih na področju programskih rešitev.

Strošek izdelave programske rešitve za izbrano organizacijo v primeru zunanjega razvoja izračunamo na osnovi enačbe (3):

- števila ur, ki ga potrebujemo za izdelavo programske rešitve iz Tabele 1 – 1024 ur,
- cene urne postavke v okviru zunanjega razvoja, pridobljene na osnovi referenčnega cenika – 95 EUR.

*Strošek izdelave programske rešitve = cena urne postavke (zunanj razvoj) * število ur za izdelavo programske rešitve (3)*

Strošek izdelave programske rešitve v primeru zunanjega razvoja znaša 97.280 EUR brez DDV.

Tabela 4: Referenčni cenik storitev s področja informacijskih storitev

Stopnja	Zahtevnost del	Značilna dela	Strokovnjaki (primeri)	Cena Delovne ure EUR
1.	Rutinska dela brez potrebnih posebnih znanj in predhodnega uvajanja.	Pisarniško administrativna ter ostala pomožna dela. Manj zahtevna servisno strojno tehnična dela.	Administrativni delavec	32
2.	Potrebno predhodno izobraževanje in izkušnje, ter krajše uvajanje – do 1 meseca.	Popravila, osnovna tehnična in vzdrževalna dela. Izvajanje anket, zbiranje gradiva. Samostojno reševanje preprostih problemov po vnaprej znanih postopkih.	Serviser, Tehnik	42
3.	Potrebno poglobljeno uvajanje do nekaj mesecev.	Enostavno programiranje. Srednje zahtevna dela na nivoju delovnih postaj in omrežij. Samostojno reševanje zahtevnejših problemov v znanih situacijah.	Administrator, Programer I, Sistemski tehnik	58
4.	Potrebno daljše specialistično izobraževanje ter najmanj 1 letne izkušnje.	Srednje zahtevno programiranje. Enostavno izobraževanje in uvajanje uporabnikov. Dokumentiranje programskih rešitev. Priprava in oblikovanje uporabniške dokumentacije. Spletno oblikovanje. Enostavno administriranje strežnikov. Samostojno reševanje zahtevnih problemov.	Programer II., Analitik, Spletni oblikovalec, Sistemski inženir I	79

nadaljevanje

5.	Reševanje zelo zahtevnih problemov, pri katerih so potrebne večletne delovne izkušnje.	Vodenje manj zahtevnih projektov, Sistemska analiza in načrtovanje, Zahtevno administriranje strežnikov.	Inženir, Administrator podatkovnih baz, Sistemski inženir II	95
6.	Reševanje izjemno zahtevnih problemov, pri katerih so potrebne večletne delovne izkušnje in specialistično izobraževanje.	Vodenje in koordinacija zahtevnih projektov. Zahtevno programiranje in izobraževanje. Konceptualno oblikovanje. Izdelava zahtevnih strokovnih gradiv.	Informacijski arhitekt, Svetovalec, Programer III., Upravljavec varnosti IS, Inženir IS	114
7.	Samostojno reševanje najzahtevnejših problemov za katere so potrebne večletne izkušnje in poglobljeno znanje na posameznem strokovnem področju.	Strateško svetovanje. Svetovanje za uvajanje. Implementacija in integracija aplikacij in kompleksnih sistemov. Zahtevna dela, analize in načrtovanja na specializiranih ali kompleksnih sistemih.	Izvedenec Ekspert	143

Vir: *Gospodarska zbornica Slovenije, Referenčni cenik storitev s področja informacijskih storitev, 2016.*

3.3 Izdelava funkcijske specifikacije

Za izdelavo funkcijske specifikacije sem porabil 180 ur dela, kar mi je vzelo približno dva meseca časa. Funkcionalnosti programske rešitve sem razdelil na dve področji:

- splošne funkcionalnosti in
- specifične funkcionalne rešitve.

3.3.1 Splošne funkcionalnosti

Vnos oziroma zajem podatkov. Vnos oziroma zajem podatkov poteka s postopkom oddaje naročila blaga v višini do 20.000 EUR oziroma storitve v višini do 40.000 EUR. Zajem podatkov vsebuje dokumente predlog za izdajo naročila, naročilnico in pogodbo (Tabela 5).

Tabela 5: Vnos oziroma zajem podatkov

Dokument »predlog za izdajo naročila« vsebuje naslednje podatke:
• zaporedna številka naročila
• naziv organizacijske enote (služba/sector)
• vrsta naročila (blago ali storitev)

se nadaljuje

nadaljevanje

• predmet naročila (količina, enota mere, znesek brez DDV, DDV, znesek z DDV)
• naziv dobavitelja (naziv, naslov, pošta, matična številka, davčna številka)
• vrednost naročila brez DDV
• vrednost naročila z DDV
• opredelitev ekonomsko najugodnejše ponudbe, način preverjanja in ocenjevanja
• opredelitev finančne postavke iz SAOP
• nazivi ponudnikov, pri katerih so se preverjala uporabljena merila
• obrazložitev izbora
• opredelitev, ali gre za naročilnico, letno naročilnico ali pogodbo
• datum vloge predloga za izdajo naročila
• vodja naročila
• vodja organizacijske enote
• odgovorna oseba
Naročilnica vsebuje naslednje podatke:
• številka naročilnice
• naziv naročnika (naslov, kraj, pošta)
• davčna številka naročnika
• matična številka naročnika
• davčni zavezanec (da/ne)
• številka predloga za izdajo naročila
• datum izdaje naročilnice
• kraj izdaje naročilnice
• naziv dobavitelja (naslov, kraj, pošta)
• vrednost naročila brez DDV in z DDV
• podpis in žig dobavitelja
• datum naročila
• naslov za odpremo
• vodja naročila
Pogodba vsebuje naslednje podatke:
• številka pogodbe
• številka predloga za izdajo naročila
• naziv organizacijske enote
• vodja naročila
• skrbnik pogodbe
• naziv dobavitelja (naziv, naslov, pošta, matična številka, davčna številka)
• zastopnik dobavitelja (naziv, naslov, pošta, matična številka, davčna številka)
• številka računa dobavitelja
• vrsta naročila (blago ali storitev)
• predmet naročila (količina, enota mere, znesek brez DDV, DDV, znesek z DDV)

se nadaljuje

nadaljevanje

• pogodbena vrednost brez DDV
• pogodbena vrednost z DDV in brez DDV
• vrednost pogodbene menice
• vrednost menice za odpravo napak
• veljavnost ponudbe
• izvedbeni rok
• garancijski rok
• datum pogajanj
• obrazložitev pogajanj
• datum potrditve sklepa o oddaji naročila
• datum preverjanja pogodbe
• zaposleni, ki je opravil pregled pogodbe
• datum podpisa pogodbe

Kontrola podatkov. Osnovna kontrola podatkov se zagotavlja v okviru podatkovnega modela, ki je predstavljen kot relacijski model podatkov. Med podatkovnimi tabelami so vzpostavljene relacije, ki zagotavljajo referenčno integriteto med podatkovnimi tabelami. Kontrola podatkov se izvaja tudi pri vnašanju vseh podatkov na vnosnih maskah, in sicer glede na to, ali vnašamo znakovni ali številčni tip podatka. Uporabniku se izpiše obvestilo v primeru vnesenega neustreznega tipa podatka.

Faze v postopku naročanja. Programska rešitev vsebuje štiri faze v postopku naročanja (Tabela 6). Te faze so: priprava naročil, potrditev naročil, knjiženje in analiza podatkov.

Tabela 6: Faze v postopku naročanja

Prva faza postopka: priprava naročil	
Področje	Opis
Moji predlogi	Vodja naročila pripravi dokument »predlog za izdajo naročila«. Predlog se lahko ureja toliko časa, dokler ga ne potrdi vodja organizacijske enote. Uporabnik lahko predlog prekliče, ga natisne in potrdi. Potrditev predloga pomeni pošiljanje predloga v potrditev vodji organizacijske enote. Predlog za izdajo naročila se lahko uvozi tudi v dokumentni sistem organizacije.
Moji nepotrjeni predlogi	Seznam vsebuje vse predloge, ki so posredovani v potrditev vodji organizacijske enote oziroma v potrditev tajništvu organizacije.

se nadaljuje

nadaljevanje

Moje nepotrjene naročilnice	Seznam vsebuje vse nepotrjene naročilnice, ki so na čakanju in še niso bile potrjene v tajništvu.
Moje nepotrjene pogodbe	Seznam vsebuje vse pogodbe, ki so bile posredovane v pregled pravnemu sektorju in v potrditev zastopniku organizacije.
Moji preklicani predlogi	Seznam vsebuje vse preklicane predloge naročil, ki jih prekliče vodja naročila, vodja organizacijske enote ali tajništvo. Predlog se lahko znova pošlje v pripravo (Moji predlogi) ali se ga izbriše iz vseh seznamov, vendar ne iz podatkovne baze.
Moji nezaključeni postopki	Seznam vsebuje vsa naročila, ki so v postopku knjiženja računov. Uporabnik izbere določeno naročilo in zanj potrdi zaključitev postopka. Na naročilo ni več možno dodajati računov.
Moji zaključeni postopki	Seznam vsebuje vse zaključene predloge za izdajo naročila. Zaključenih predlogov ni več mogoče preklicati ali izbrisati.
Druga faza postopka: potrditev naročil	
Področje	Opis
Predlogi v pripravi za potrditev	Seznam vsebuje pregled vseh predlogov, ki jih potrdi vodja naročila, in je namenjen uporabnikom v tajništvu in vodjem organizacijskih enot. Omogočeno je potrjevanje, preklic predloga ali vrnitev predloga vodji naročila. Uporabnik lahko ureja tudi znesek naročila. Dokončna potrditev predloga pomeni, da je predlog potrdil vodja organizacijske enote in zastopnik organizacije.
Naročilnice v pripravi za potrditev	Seznam vsebuje pregled vseh naročilnic v pripravi in je namenjen zaposlenemu v tajništvu. V primeru, da se ugotovi neustreznost naročila, se to lahko vrne v predhodno fazo in znova vodji naročila. Za izdano naročilnico se vpiše številka naročilnice, ostali podatki se prenesejo iz predloga naročila. Izpolnjeno naročilnico se natisne, možno pa je natisniti tudi neizpolnjeno naročilnico, kadar zaradi časovne stiske ni mogoče izpeljati celotnega postopka naročila.

se nadaljuje

nadaljevanje

Pogodbe v pripravi za potrditev	Seznam vsebuje pregled vseh predlogov pogodb, ki jih potrdi vodja naročila in je namenjen uporabnikom v pravni službi in tajništvu. Omogočeno je potrjevanje, preklic predloga pogodbe ali vrnitev predloga pogodbe vodji naročila. Dokončna potrditev predloga pogodbe pomeni, da je predlog pogodbe potrdil zaposleni v pravni službi in zastopnik organizacije. V primeru, da se ugotovi neustreznost predloga pogodbe, se ta lahko vrne v prehodno fazo in znova vodji naročila. Za pripravo predloga pogodbe se določeni podatki prepisejo iz predloga naročila, ostali podatki pa se urejajo programu MS Word.
Tretja faza postopka: knjiženje	
Področje	Opis
Naročila v postopku	Seznam vsebuje vsa naročila, ki so bila potrjena v postopku naročanja. in je namenjen uporabnikom v finančni službi. Ko uporabnik poknjiži prvi račun, se podatki avtomatsko prepisejo v SAOP in naročilo preide med nezaključena naročila.
Nezaključena naročila	Seznam vsebuje vsa nezaključena naročila, ki imajo knjižen en račun ali več računov. Zaposleni v finančni službi lahko knjiži račune toliko časa, dokler jih vodja naročila ne zaključi. V primeru napačnega knjiženja je omogočeno tudi spreminjanje podatkov računa.
Četrta faza postopka: analiza podatkov	
Odprti postopki v tajništvu	
Pregledovalniki so namenjeni pregledu podatkov v postopku potrjevanja naročil v tajništvu, in sicer za vsa naročila, ki imajo potrjene predloge za izdajo naročila, nimajo pa še potrjenih naročilnic in pogodb. Pregledovalniki prikazujejo podatke glede na različne ravni organizacije.	
Področje	Opis
Glede na organizacijo	Prikazani so vsi postopki skupaj po vrsti naročila za poljubno obdobje. Omogočen je tudi podroben pregled posameznega naročila.
Glede na organizacijsko enoto	Prikazani so vsi postopki skupaj po vrsti naročila za poljubno obdobje po organizacijskih enotah organizacije. Omogočen je tudi podrobni pregled posameznega naročila.

se nadaljuje

nadaljevanje

Glede na vodjo naročila	Prikazana so vsa nepotrjena naročila. Za posameznega vodjo naročila je omogočen vpogled v vse postopke skupaj za poljubno obdobje. Omogočen je tudi podroben pregled posameznega naročila.
Odprti postopki v finančah	
Pregledovalniki so namenjeni pregledu podatkov v postopku potrjevanja naročil v finančah, in sicer za vsa naročila, ki so potrjena, nimajo pa še knjiženih računov. Pregledovalniki prikazujejo podatke glede na različne ravni organizacije.	
Področje	Opis
Glede na organizacijo	Prikazani so vsi postopki skupaj po vrsti naročila za poljubno obdobje. Omogočen je tudi podroben pregled nepotrjenega postopka v finančni službi.
Glede na organizacijsko enoto	Prikazani so vsi postopki skupaj po vrsti naročila za poljubno obdobje po organizacijskih enotah organizacije. Omogočen je tudi podroben pregled nepotrjenega postopka v finančni službi.
Glede na vodjo naročila	Prikazana so vsa nepotrjena naročila. Za posameznega vodjo naročila je omogočen vpogled v vse postopke skupaj za poljubno obdobje. Omogočen je tudi podroben pregled nepotrjenega postopka v finančni službi.
Fakturirani postopki	
Pregledovalniki so namenjeni pregledu podatkov v postopku fakturiranja naročil v finančah, in sicer za vsa naročila, ki imajo knjižen en račun ali več računov. Pregledovalniki prikazujejo podatke glede na različne ravni organizacije.	
Področje	Opis
Glede na organizacijo	Prikazani so vsi postopki skupaj po vrsti naročila za poljubno obdobje. Omogočen je tudi podroben pregled posameznega knjiženega naročila.
Glede na organizacijsko enoto	Prikazani so vsi postopki skupaj po vrsti naročila za poljubno obdobje po organizacijskih enotah organizacije. Omogočen je tudi podroben pregled posameznega knjiženega naročila.
Glede na vodjo naročila	Prikazana so vsa nepotrjena naročila. Za posameznega vodjo naročila je omogočen vpogled v vse postopke skupaj za poljubno obdobje. Omogočen je tudi podroben pregled posameznega knjiženega naročila.
Zaključeni postopki	

se nadaljuje

nadaljevanje

Pregledovalniki so namenjeni pregledu vseh zaključenih postopkov. To so vsa naročila, ki jih je zaključil vodje naročila, in na njih ni več mogoče izdajati računov. Pregledovalniki prikazujejo podatke glede na različne ravni organizacije.	
Področje	Opis
Glede na organizacijo	Prikazani so vsi postopki skupaj po vrsti naročila za poljubno obdobje. Omogočen je tudi podroben pregled zaključenih postopkov.
Glede na organizacijsko enoto	Prikazani so vsi postopki skupaj po vrsti naročila za poljubno obdobje po organizacijskih enotah organizacije. Omogočen je tudi podroben pregled zaključenih postopkov.
Glede na vodjo naročila	Prikazana so vsa nepotrjena naročila. Za posameznega vodjo naročila je omogočen vpogled v vse postopke skupaj za poljubno obdobje. Omogočen je tudi podroben pregled zaključenih postopkov.

Druge funkcionalnosti. Ostale pomembnejše funkcionalnosti so:

- stranka na dokumentih,
- predmet naročil,
- finančna postavka,
- znesek naročila,
- knjiženje računov,
- povezava z dokumentnim sistemom.

Stranka na dokumentih. Uporabnik izbere dobavitelja v postopkih priprave predloga in pogodbe iz šifranta strank. Šifrant strank je povezan s SAOP. Podatki se avtomatsko replicirajo v aplikacijo. V primeru, da finančna stranka ni vpisana v podatkovni bazi, zaposleni v finančnem sektorju zagotovi vpis stranke v SAOP na osnovi zahteve uporabnika aplikacije.

Predmet naročil. Uporabnik ima možnost dodati več predmetov naročil v okviru postopka izdaje predloga naročila.

Finančna postavka. Uporabnik izbere finančno postavko, iz katere bodo zagotovljena sredstva za nabavo blaga ali storitve. Finančna postavka je šifrant v SAOP. Podatki se avtomatsko replicirajo v aplikacijo. V primeru, da finančna postavka ni natančno opredeljena v podatkovni bazi, uporabnik posreduje predlog nadrejenemu za izbiro prave finančne postavke.

Znesek naročila. Omogoča se urejanje zneska naročila v primeru, da je vodja naročila vpisal napačen znesek na predlog za izdajo naročila. Znesek naročila lahko popravi zaposleni v tajništvu organizacije. V tem primeru vodja naročila in vodja organizacijske enote znova podpišeta popravljeni predlog za izdajo naročila.

Knjiženje računov. Zaposleni v finančnem sektorju lahko na en predlog naročila knjiži enega ali več računov, vse dokler vodja naročila ne zaključi posameznega naročila. Omogoči se tudi popravljanje zneskov naročila, če pride do napačnih vnosov. Podatki knjiženja se replicirajo v podatkovno bazo SAOP, tako da knjiženje računov v SAOP ni potrebno.

Povezava z dokumentnim sistemom. V postopkih je omogočeno tudi dodajanje dokumentov v dokumentni sistem. V dokumentni sistem se lahko dodajo dokumenti, ki nastanejo v aplikaciji za oddajo naročila ali kateri koli drugi dokumenti v postopku naročanja.

Ostale zahteve programske rešitve. Ostale zahteve programske rešitve se nanašajo na arhitekturo celotnega informacijskega sistema v izbrani organizaciji:

- programska rešitev je izdelana modularno, kar pomeni, da je predstavitveni nivo ločen od poslovne logike. Poslovna logika je izdelana v podatkovni bazi;
- za dostop do baze se uporabljajo samo bazne procedure, ne pa neposredne povezave na tabele;
- programska rešitev deluje tudi na oddaljenih lokacijah, na katerih hitrost prenosa znaša vsaj 0,5 MB na uporabnika. Omogočeno je, da na oddaljenih lokacijah dela ca. 20 uporabnikov;
- programska rešitev deluje z vsaj maksimalno 200 uporabniki;
- odzivni časi aplikacije niso krajši od 5 sekund/zahtevo;
- vsi strežniki so postavljeni centralizirano;
- programska rešitev je napisana in dokumentirana v skladu s standardi razvoja programskih rešitev. Programska koda in elementi v podatkovni bazi so logično jasno poimenovani glede na namen uporabe;
- programska rešitev ob primopredaji postane last naročnika.

3.3.2 Specifične funkcionalne rešitve

Splošne značilnosti vmesnikov. Splošne značilnosti grafičnih vmesnikov, razen kjer to ni drugače določeno, so:

- na vrhu grafičnega vmesnika se vedno nahaja opis identifikacije uporabnika (uporabniško ime, ime in priimek ter opis pravic);

- pod identifikacijo se nahaja navigacija z dinamično strukturo (meniji z dinamično strukturo);
- v sredini se nahaja polje za prikaz vsebin, ki ga je možno poljubno spreminjati po širini in višini;
- prikazuje splošne značilnosti vseh tabelarnih prikazov v programski rešitvi: izbrano število zadetkov, število strani je označeno s trenutno pozicijo, indeksiranje po katerem koli stolpcu tabele; nudi možnost spreminjanja velikosti fontov;
- možna je uporaba standardiziranih slik pri gumbih in hiperpovezavah;
- vse spremembe podatkov se beležijo, tako da je razvidno, kdo je dostopal, kdaj, s katerega naslova IP, kaj je bilo spremenjeno, v kateri tabeli se je zgodila sprememba, zabeleži se tudi prejšnja vrednost (vsebina) podatka;
- za vsa tekstovna polja, katerih vpis ne temelji na šifrantu, se pri vnosu preveri dolžina vpisa (možna napaka zaradi predolgega znakovnega niza) in prikaže ustrezno opozorilo;
- programska rešitev mora v kar največji možni meri omejiti klice na strežnik, da se poveča učinkovitost pri vnosu podatkov;
- za namen oblikovanja se uporabijo datoteke CSS.

Vnos podatkov. Programska rešitev je zasnovana tako, da uporabniku v čim večji meri olajša izpeljati postopek oddaje javnega naročila po evidenčnem postopku za naročila v višini do 5.000 EUR, ko gre za izdajo naročilnice, in v višini do 20.000 EUR oziroma 40.000 EUR v primerih oddaje naročila s pogodbo. Programska rešitev je prilagojena izdelavi in evidenci oddaje javnega naročila po evidenčnem postopku.

Priprava naročil obsega faze:

- moji predlogi,
- moji nepotrjeni predlogi,
- moje nepotrjene naročilnice,
- moje nepotrjene pogodbe,
- moji preklicani predlogi,
- moji nezaključeni postopki,
- moji zaključeni postopki.

Potrditev naročil obsega faze:

- predlogi v pripravi za potrditev,
- naročilnice v pripravi za potrditev,
- pogodbe v pripravi za potrditev.

Knjiženje obsega fazi:

- naročila v postopku,
- nezaključena naročila.

Analiza podatkov obsega faze:

- odprti postopki v tajništvu,
- odprti postopki v financah,
- fakturirani postopki,
- zaključeni postopki.

Notranji tok dokumentov. Vodja naročila v seznamu pripravi predlog naročila in ga pošlje v potrditev vodji organizacijske enote in v tajništvo. Vodja organizacijske enote in tajništvo lahko predlog zavrneta. Po potrditvi predlog preide v fazo »naročilnice v pripravi za potrditev« oziroma v fazo »pogodbe v pripravi za potrditev«. Naročilnica oziroma pogodba se lahko potrdi ali zavrne. Pogodbo potrdi zaposleni v pravnem sektorju v fazi »pogodbe v pripravi za potrditev«. Ko je naročilo (naročilnica ali pogodba) potrjeno, se izvede. Postopek naročanja preide v fazo »naročila v postopku«. Za naročilo se izda enega ali več računov, vse dokler vodja naročila ne zaključi postopka v fazi »moji nezaključeni postopki«.

Glavne povezave. Glavne povezave, ki so namenjene uporabnikom v postopku oddaje javnega naročila po evidenčnem postopku so prikazane na Sliki 11.

Slika 11: Glavni meni programske rešitve

Javna naročila po evidenčnem postopku		Uporabnik: org001 Pavel Košir Administrator			
Priprava naročil ▶	Potrditev naročil ▶	Knjiženje ▶	Analiza podatkov ▶	Administracija ▶	Pomoč in navodila ▶

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 7.

Uporabnikom aplikacije se dodelijo različne pravice (Tabela 7) glede na vrsto uporabnika, ki je lahko:

- Uporabnik1 – vodja naročila,
- Uporabnik2 – tajništvo,
- Uporabnik3 – finance,
- Uporabnik4 – pravna služba,
- Uporabnik5 – administrator.

Tabela 7: Uporabniki menijev programske rešitve

Meni programske rešitve	Uporabnik
Priprava naročil	Uporabnik1
Potrditev naročil	Uporabnik2/Uporabnik3/Uporabnik4
Knjiženje	Uporabnik3
Analiza podatkov	Uporabnik1
Administracija	Uporabnik5
Pomoč in navodila	Uporabnik1

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 9.

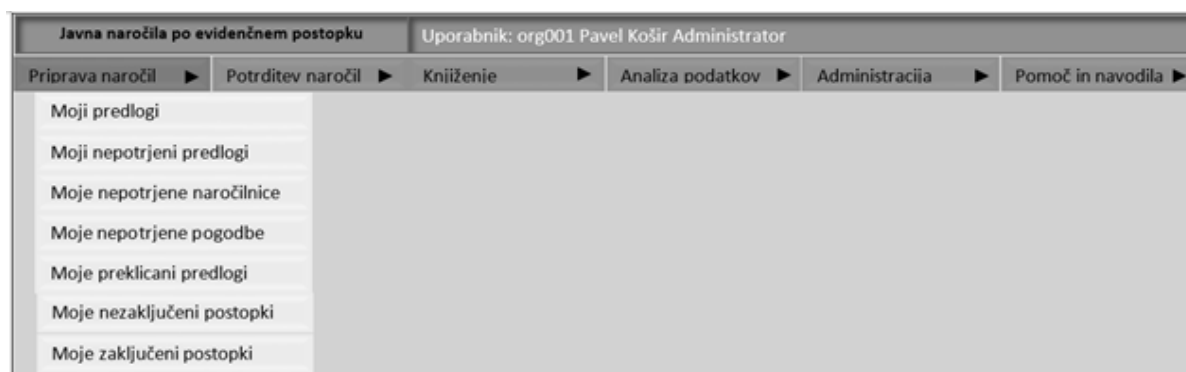
Uporabniški vmesniki so opisani v nadaljevanju.

1. Povezava »priprava naročil«

Povezavo uporablja vodja naročila za oddajo naročila po evidenčnem postopku za naročila v višini do 5.000 EUR, ko gre za izdajo naročilnice, in v višini do 20.000 EUR oziroma 40.000 EUR v primerih oddaje naročila s pogodbo. Vodja naročila v postopku naročanja uporablja povezave (Slika 12):

- moji predlogi,
- moji nepotrjeni predlogi,
- moje nepotrjene naročilnice,
- moje nepotrjene pogodbe,
- moji preklicani predlogi,
- moji nezaključeni postopki,
- moji zaključeni postopki.

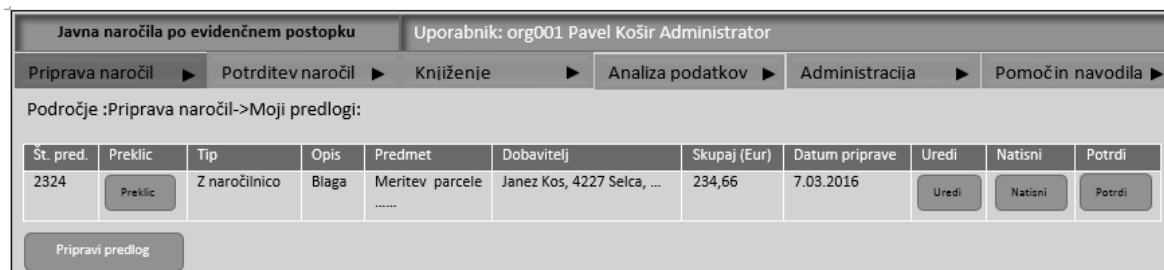
Slika 12: Povezava »priprava naročil«



Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 8.

Moji predlogi. Povezava »moji predlogi« prikaže seznam pripravljenih predlogov za oddajo naročila (Slika 13).

Slika 13: Povezava »moji predlogi«



Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 8.

Podatki seznama »moji predlogi« so prikazani v Tabeli 8.

Tabela 8: Podatki povezave »moji predlogi«

Podatek	Opis
Številka predloga	zaporedna številka predloga naročila
Preklic predloga	gumb za preklic predloga naročila
Tip naročila	tip naročila (naročilnica ali pogodba)
Opis naročila	vrsta naročila (blago, storitev ali gradnja)
Predmet naročila	opis predmeta naročila
Dobavitelj	naziv in naslov dobavitelja
Znesek skupaj	skupen znesek naročila
Datum priprave predloga	datum priprave predloga naročila
Uredi predlog	gumb za urejanje podatkov predloga
Natisni predlog	gumb za izbiro tiskanja predloga
Potrdi predlog	gumb za potrditev predloga

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 8.

Gumb »pripravi predlog«. Uporabnik izbere gumb »pripravi predlog«, ki kreira nov predlog naročila. Odpre se vnosna maska za kreiranje oziroma spreminjanje podatkov naročila. Vnosna maska »predlog naročila« je prikazana v podpoglavju 4.5.5.

Dokument »predlog za izdajo naročila«. Dokument je v obliki za tiskanje in je prikazan na Sliki 14.

Slika 14: Dokument »predlog za izdajo naročila«

Izbrana organizacija Slovenska cesta 1, 1000 Ljubljana Tel.: 01/1000-100; Faks.: 01/1001-101				
Številka:				
Datum:				
PREDLOG ZA IZDAJO NAROČILNICE ŠT.: 4822				
Organizacijska enota (sektor, oddelek);				
Sektor za splošne zadeve, Služba za in informatiko				
Prosimo za izdajo naročilnice na naslov (naziv in naslov dobavitelja, oz. izvajalca):				
Pavel Košir, Bukovščica 1, 4227 Selca				
Za dobavo oziroma izvedbo: Blaga				
Predmet naročila:				
Predmet naročila	Količina	Enota mere	Znesek (Eur)	Znesek DDV (Eur)
Meritev parcele ...	1	1	1,00	0,00
Vrednost brez DDV znaša 1,00 Eur.				
Vrednost DDV-ja znaša 0,00 Eur.				
Vrednost z DDV znaša 1,00 Eur.				
Sredstva za izvedbo javnega naročila so zajeta v finančni postavki: Geodetske meritve.				
Obrazložitev izbora (način preverjanja cene, oziroma izvedbe ocenjevanja in usposobljenosti ponudnikov, imena in naslovi ponudnikov, pri katerih so se preverjala uporabljena merila), oz. obrazložitev razlogov, zaradi katerih sta cena in usposobljenost preverjala pri manj kot treh ponudnikih:				
Najbolj ugoden ponudnik				
V skladu s pravilnikom se naročilo odda: Z naročilnico.				
Datum vloge za naročilnico: 4.3.2016				
Vodja naročila: Pavel Košir				
Podpis vodje naročila:		Podpis vodje organizacijske enote:		
_____		_____		

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 10.

Vnosna maska »predmet naročnik«. Uporablja se za vnos in urejanje podatkov o posameznemu predmetu naročila in je prikazana na Sliki 15. Na en predlog naročila se lahko vpiše več predmetov naročil.

Slika 15: Vnosna maska »predmet naročil«

The screenshot shows a web form titled 'Predmet naročila'. It contains the following fields and controls:

- A large text input field for 'Vrsta blaga oziroma storitev:'.
- Input fields for 'Količina:', 'Enota mere:', 'Znesek (Eur):', and 'Znesek DDV (Eur):'.
- A dropdown menu for tax rate, currently set to '22%'.
- 'Potrdi' (Confirm) and 'Prekliči' (Cancel) buttons at the bottom.

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 11.

Vnosna maska »dobavitelj«. Uporablja se za vnos dobavitelja za določeno naročilo (Slika 16). Podatki se preberejo iz finančne baze. Omogočeno je iskanje po kombinaciji atributov o dobaviteljih in omogočen je vnos podatkov za neizbrane ponudnike.

Slika 16: Vnosna maska »dobavitelj«

The screenshot shows a web form titled 'Dobavitelj'. It contains the following fields and controls:

- Input fields for 'Naziv:', 'Naslov:', 'Matična številka:', and 'Davčna številka:'.
- Input fields for 'Pošta:' and 'Številka:'.
- 'Išči' (Search) and 'Potrdi' (Confirm) buttons.
- Input fields for 'Dobavitelj:', 'Ponudnik (1):', and 'Ponudnik (2):'.
- 'Potrdi' (Confirm) and 'Prekliči' (Cancel) buttons at the bottom.

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 11.

Moji nepotrjeni predlogi. Povezava »moji nepotrjeni predlogi« prikaže vse predloge naročil, ki so v postopku potrditve pri vodji organizacijske enote, v tajništvu ali pravni službi. Nekateri podatki so enaki kot pri seznamu »moji predlogi«: številka predloga, tip naročila, opis naročila, predmet naročila, dobavitelj in znesek skupaj. Ostali podatki seznama »moji nepotrjeni predlogi« so prikazani v Tabeli 9.

Tabela 9: Ostali podatki povezave »moji nepotrjeni predlogi«

Podatek	Opis
Datum vloge predloga	datum vloge predloga naročila

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 12.

Moje nepotrjene naročilnice. Povezava »moje nepotrjene naročilnice« prikaže vse naročilnice, ki so v postopku potrditve v tajništvu. Nekateri podatki so enaki kot pri seznamu »moji predlogi«: številka predloga, tip naročila, opis naročila, predmet naročila, dobavitelj in znesek skupaj. Ostali podatki so prikazani v Tabeli 10.

Tabela 10: Ostali podatki povezave »moje nepotrjene naročilnice«

Podatek	Opis
Datum potrditve predloga	datum potrditve predloga naročila

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 13.

Moje nepotrjene pogodbe. Povezava »moje nepotrjene pogodbe« prikaže vse pogodbe, ki so v postopku potrditve v pravni službi. Vsebuje enake podatke kot seznam »moje nepotrjene naročilnice«.

Moji preklicani predlogi. Povezava »moji preklicani predlogi« prikaže vse predloge, ki so jih preklicali bodisi vodja naročila, tajništvo, vodja organizacijske enote ali pravna služba. Nekateri podatki so enaki kot pri seznamu »moji predlogi«: številka predloga, tip naročila, opis naročila, predmet naročila, dobavitelj in znesek skupaj. Ostali podatki so prikazani v Tabeli 11.

Tabela 11: Ostali podatki povezave »moji preklicani predlogi«

Podatek	Opis
Datum priprave predloga	datum priprave predloga naročila
Pošlji v pripravo	gumb za pošiljanje predloga v vnovično pripravo

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 14.

Moji nezaključeni postopki. Povezava »moji nezaključeni postopki« prikaže vse postopke v knjiženju v finančni službi. Ko vodja naročila zaključi postopek, nanj ni več mogoče knjižiti računov. Nekateri podatki so enaki kot pri seznamu »moji predlogi«: številka predloga, tip naročila, opis naročila, predmet naročila, dobavitelj in znesek skupaj. Ostali podatki so prikazani v Tabeli 12.

Tabela 12: Ostali podatki povezave »moji nezaključeni postopki«

Podatek	Opis
Datum potrditve predloga	datum potrditve predloga naročila
Datum fakture predloga	datum fakture predloga naročila
Številka naročilnice/pogodbe	številka naročilnice oziroma pogodbe naročila
Zaključitev postopka	gumb za zaključitev postopka naročila

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 14.

Moji zaključeni postopki. Povezava »moji zaključeni postopki« prikaže seznam vseh zaključenih postopkov vodje naročila. Podatki so enaki kot pri seznamu »moji nezaključeni postopki«. Ostali podatki so navedeni v Tabeli 13.

Tabela 13: Ostali podatki povezave »moji zaključeni postopki«

Atribut	Opis
Datum zaključka	datum zaključka naročila

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 15.

2. Povezava »potrditev naročil«

Povezavo uporabljajo zaposleni v tajništvu organizacije, vodje organizacijskih enot in zaposleni v pravni službi in obsega možnosti:

- predlogi v pripravi za potrditev,
- naročilnice v pripravi za potrditev,
- pogodbe v pripravi za potrditev.

Predlogi v pripravi za potrditev. Povezava »predlogi v pripravi za potrditev« prikaže seznam naročil, ki jih potrdijo vodje naročila, niso pa jih potrdili tajništvo, vodja organizacijske enote in pravna služba. Omogočen je preklic predloga, urejanje zneska naročila, vrnitev predloga vodji naročila in potrditev predloga. Nekateri podatki so enaki kot pri seznamu »moje nepotrjene naročilnice«. Ostali podatki so prikazani v Tabeli 14.

Tabela 14: Ostali podatki povezave »predlogi v pripravi za potrditev«

Podatek	Opis
Preklic predloga naročila	gumb za preklic naročila

se nadaljuje

nadaljevanje

Datum priprave predloga	datum priprave predloga naročila
Uredi znesek	odpre vnosno masko za urejanje zneska naročila
Vrni predlagatelju	gumb za vrnitev naročila v ponovno pripravo
Natisni predlog	gumb za izbiro tiskanja predloga
Potrditev	gumb za potrditev naročila (tajništvo, vodja organizacijske enote in zaposleni v pravni službi)
Datum potrditve vodja	datum potrditve predloga vodje organizacijske enote

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 15.

Vnosna maska »urejanje zneska naročila«. Omogočeno je urejanje zneska naročila za posamezni predmet naročila (Slika 17).

Slika 17: Vnosna maska »urejanje zneska naročila«

Urejanje zneska naročila

Naročilo z identifikacijo: 2059

Pregled predmetov naročil

Številka predmeta naročila	Znesek (Eur)	DDV (Eur)	Uredi znesek predmeta
27733	425,40	85,08	Uredi znesek predmeta

Številka predmeta	DDV (Eur)	Znesek (Eur)	Potrdi urejanje
<input type="text" value="27733"/>	<input type="text" value="85,08"/> <input type="text" value="0%"/>	<input type="text" value="425,4"/>	<input type="button" value="Potrdi"/>

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 16.

Naročilnice v pripravi za potrditev. Povezava »naročilnice v pripravi za potrditev« prikaže seznam vseh potrjenih predlogov, ki še nimajo izdane naročilnice. Nekateri podatki so enaki kot pri seznamu »moje nepotrjene naročilnice«. Ostali podatki so prikazani v Tabeli 15.

Tabela 15: Ostali podatki povezave »naročilnice v pripravi za potrditev«

Podatek	Opis
Datum priprave predloga	datum priprave predloga naročila
Vrni v fazo potrditve	gumb za vrnitev naročila v fazo potrditve naročila

se nadaljuje

nadaljevanje

Številka naročilnice	gumb za odpiranje vnosne maske za vpis številke naročilnice
Natisni naročilnico	gumb za izbiro tiskanja naročilnice
Potrditev	gumb za potrditev naročilnice

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 17.

Vnosna maska »številka naročilnice«. Namenjena je vpisu in urejanju številke naročilnice (Slika 18).

Slika 18: Vnosna maska »številka naročilnice«

Številka naročilnice

Številka predloga: 1001

Številka naročilnice:

Potrdi
Prekliči

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 17.

Dokument »naročilnica«. Dokument je v obliki za tiskanje (Slika 19). Večina podatkov se prebere iz podatkovne baze.

Slika 19: Dokument »naročilnica«

Izdajatelj: Izbrana organizacija, Slovenska cesta 1, 1000 Ljubljana		Številka predloga: 2034	Kraj izdaje: Škofja Loka	Naročilnica	
Davčna št.: SI00000000		Datum izdaje: 10.3.2016		Številka: 121/3	
Dobavitelj: Pavel Košir, Bukovščica 1, 4227 Selca			Odpremite naslovu: Pavel Košir, Bukovščica 1, 4227 Selca		
Količina	ME	Vrsta blaga oz. storitve	Znesek (EUR)	Znesek DDV (EUR)	Odpremjeno
		cenitev gradbene parcele	125,15	25,03	Da
Skupaj znesek brez DDV (EUR):				125,15	
Skupaj znesek DDV (EUR):				25,03	
Skupaj znesek (EUR):				150,17	
Odpremil (Ime in Priimek): _____		Datum: _____		Naročil (Ime in Priimek): _____	
Datum: _____		Datum: _____		Datum: _____	
Podpis: _____		Žig: _____		Podpis: _____	
Datum: _____		Datum: _____		Datum: _____	
Ponatis prepovedani!					

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 17.

Pogodbe v pripravi za potrditev. Povezava »pogodbe v pripravi za potrditev« prikaže seznam vseh potrjenih predlogov, ki še nimajo izdane pogodbe. Nekateri podatki so enaki kot pri seznamu »moje nepotrjene naročilnice«. Ostali podatki so prikazani v Tabeli 16.

Tabela 16: Ostali podatki povezave »pogodbe v pripravi za potrditev«

Podatek	Opis
Datum priprave predloga	datum priprave predloga naročila
Vrni v fazo potrditve	gumb za vrnitev naročila v fazo potrditve naročila
Številka pogodbe	gumb za odpiranje vnosne maske za vpis številke pogodbe
Uredi pogodbo	gumb, ki odpre pogodbo v programu MS Word
Natisni pogodbo	gumb za izbiro tiskanja pogodbe
Potrditev	gumb za potrditev pogodbe

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 18.

Dokument »pogodba«. Dokument je v obliki za tiskanje. Nekateri podatki se preberejo iz podatkovne baze. Členi pogodbe in ostali podatki se urejajo v programu MS Word. Izdela se standardna predloga pogodbe. Podatki, ki se preberejo iz podatkovne baze, so: številka pogodbe, številka predloga, naziv organizacijske enote, vodja naročila, skrbnik pogodbe, naziv dobavitelja (naziv, naslov, pošta, matična številka, davčna številka), zastopnik dobavitelja (naziv, naslov, pošta, matična številka, davčna številka), številka računa dobavitelja, vrsta naročila (blago ali storitev), predmet naročila (količina, enota mere, znesek brez DDV, DDV, znesek z DDV), pogodbena vrednost brez DDV, pogodbena vrednost z DDV, podatki, ki se vpišejo v urejevalniku MS Word: vrednost pogodbene menice, vrednost menice za odpravo napak, veljavnost ponudbe, izvedbeni rok, garancijski rok, datum pogajanj, obrazložitev pogajanj, datum potrditve sklepa o oddaji naročila. Ostali podatki so prikazani v Tabeli 17.

Tabela 17: Ostali podatki povezave »dokument pogodba«

Podatek	Opis
Datuma podpisa pogodbe	datum podpisov pogodbenih strank
Podpisnika pogodbe	podpis pogodbenih strank

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 18.

3. Povezava »knjiženje«

Povezavo uporabljajo zaposleni v finančni službi in obsega: naročila v postopku in nezaključena naročila.

Naročila v postopku. Povezava »naročila v postopku« prikaže seznam naročil, ki so bila oddana in še niso bila poknjžena. Nekateri podatki so enaki kot pri seznamu »moje nepotrjene naročilnice«. Ostali podatki so prikazani v Tabeli 18.

Tabela 18: Ostali podatki povezave »knjiženje«

Podatek	Opis
Vodja naročila	uporabniško ime, ime in priimek vodje naročila
Datum priprave predloga	datum priprave predloga naročila
Številka naročilnice/pogodbe	številka naročilnica oziroma pogodbe
Račun	gumb za prikaz vnosne maske »račun«

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 19.

Vnosna maska »račun« je namenjena knjiženju računov (Slika 20).

Slika 20: Vnosna maska »račun«

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 19.

Nezaključena naročila. Povezava »nezaključena naročila« prikaže seznam naročil, ki imajo knjižen en račun ali več. Nekateri podatki so enaki kot pri seznamu »knjiženje«. Ostali podatki so prikazani v Tabeli 19.

Tabela 19: Ostali podatki povezave »nezaključena naročila«

Podatek	Opis
Število računov	število knjiženih računov na naročilu
Uredi račun	gumb za prikaz vnosne maske za urejanje računa

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 20.

Vnosna maska »urejanje računa« je namenjena urejanju že knjiženega računa (Slika 21).

Slika 21: Vnosna maska »urejanje računa«

Urejanje računa

Naročilo z identifikacijo: 4142
Številka naročilnice: 22/2

Pregled računov

Številka računa	Briši račun	DDV (Eur)	Znesek	Skupaj znesek (Eur)	Datum vpisa	Uredi račun
12121	Briši	12.121,00	44,70	12.165,70	21.2.2007	Uredi račun

Številka računa	DDV (Eur)	Znesek (Eur)	Datum vpisa	Potrdi urejanje
<input type="text" value="12121"/>	<input type="text" value="12121"/>	<input type="text" value="44,7"/>	<input type="text" value="1.3.2016"/>	<input type="button" value="Potrdi"/>

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 19.

4. Analiza podatkov

Uporabljajo se izpisi naročil za področja:

- odprti postopki v tajništvu,
- odprti postopki v financah,
- fakturirani postopki,
- zaključeni postopki.

Pregled podatkov je možen glede na raven organizacije, po organizacijski enoti in po vodji naročila.

Raven organizacije. Omogočeno je izpisovanje sumarnih podatkov po poljubnem obdobju glede na vrsto naročila in podroben pregled posameznih naročil. Prikazani so podatki: poljubno časovno obdobje od do, znesek DDV, znesek brez DDV, skupaj (EUR), število postopkov, vrsta naročila in tip naročila.

Raven organizacijske enote. Omogočeno je izpisovanje sumarnih podatkov po poljubnem obdobju po organizacijskih enotah in podroben pregled posameznih naročil. Prikazani so podatki: poljubno časovno obdobje od do, znesek DDV, znesek brez DDV, skupaj (EUR), število postopkov in organizacijska enota.

Po vodji naročila. Omogočeno je izpisovanje sumarnih podatkov po poljubnem obdobju po vodji naročila in podroben pregled posameznih naročil. Prikazani so podatki: poljubno časovno obdobje od do, znesek DDV, znesek brez DDV, skupaj (EUR), število postopkov, vodja naročila.

Uporabniške vloge. Programska rešitev je samostojna spletna aplikacija, ki samodejno prevzame uporabnika Windows za svojega uporabnika. Povezava do podatkovne baze poteka prek strežnika Active Directory (v nadaljevanju AD). Uporabnik mora biti definiran tako na strežniku AD kot na podatkovnem strežniku. Uporabniške vloge so definirane v podatkovni bazi. Pravice za dodeljevanje vlog ima le administrator programa. Programska rešitev ima pet nivojev uporabnikov:

- Uporabnik1 – vodja naročila,
- Uporabnik2 – tajništvo,
- Uporabnik3 – finance,
- Uporabnik4 – pravna služba,
- Uporabnik5 – administrator.

Uporabnik1 je najpogostejši uporabnik programske rešitve. Uporabnik nima pravic potrjevanja naročil in knjiženja računov. Uporabnik2 ima enake pravice kot Uporabnik1, le da ima dodeljene tudi pravice potrjevanja naročil. Uporabnik3 ima enake pravice kot Uporabnik1, le da ima dodeljene tudi pravice potrjevanja naročil in knjiženja računov. Uporabnik4 ima enake pravice kot Uporabnik1, le da ima dodeljene tudi pravice potrjevanja naročil. Uporabnik5 je administrator in ima dostop do vseh funkcionalnosti programske rešitve. Administrator spreminja pravice uporabnikov v meniju »administracija«, tako da poišče določenega uporabnika strežnika AD in mu dodeli ustrezne uporabniške pravice. Meni »analiza podatkov« je dostopen vsem uporabnikom programske rešitve.

3.4 Izdelava podatkovnega modela

Podatkovna baza je vzpostavljena na strežniku MS SQL 2012. Pri vseh tabelah so definirani primarni indeksi. Na ravni podatkovne baze so opredeljene tudi relacije med tabelami in referenčna integriteta podatkov. Podatkovni model je zasnovan okoli glavne tabele »predlogi«, ki opredeljuje vse podatke predloge naročila. Zasnovan je kot relacijski model podatkov. Podatkovni model sem zasnoval v naslednjih korakih:

- podatkovna analiza – normalizacija,
- proučitev podatkovnega modela,
- postavitve podatkovnega modela na strežniku – določitev podatkovnih tipov,
- prepis starih podatkov v nov podatkovni model.

3.4.1 Podatkovna analiza – normalizacija

Postopki podatkovne analize – normalizacije za posamezne dokumente so opisani v Tabeli 20.

Tabela 20: Podatkovna analiza – normalizacija

Dokument: predlog za naročilo	
Opis aktivnosti	Predlog rešitve
1. Podatkovni elementi predloga za naročilo	številka_predloga, številka_naročila, datum_potrditve, organizacijska_enota, številka_dobavitelja, šifra_postavke, predmet_naročila, znesek, znesek_ddv, šifra_finančne_postavke, obrazložitev_izbora
2. Določitev ključa relacije	Številka_predloga
3. Ponavljajoče se skupine podatkovnih elementov	Jih ni.
4. Prevedba nenormalizirane relacije v prvo normalno obliko	Novi podatkovni elementi ne obstajajo. Nove relacije ni (ne obstajajo ponavljajoče se skupine podatkovnih elementov), zato ključ ostane številka_predloga.
5. Prevedba iz prve normalne oblike v drugo	Ni potrebno.
6. Prevedba iz druge normalne oblike v tretjo. Proučimo odnose med atributi v relacijah (če poznam atribut »A«, ali poznam atribut »B«) in odstranimo vzajemno soodvisne attribute v nove relacije.	
Številka_naročila	Nova podatkovna tabela »naročilnica«.
Organizacijska_enota	Podatek se prebere iz imenika AD.
Številka_dobavitelja	Podatek se prebere iz finančne baze.
Šifra_postavke	Nova podatkovna tabela postavke_naročila.
Sifra_finančne_postavke	Podatek se prebere iz finančne baze.
Tip_dokumenta	Nova podatkovna tabela naziv_dokumenta.
Dokument: naročilnica	
Opis aktivnosti	Predlog rešitve

se nadaljuje

nadaljevanje

1. Podatkovni elementi naročilnice	Večina podatkovnih elementov naročilnice je zajetih že v predlogu za izdajo naročilnice. Ostali podatkovni elementi so: številka_naročila, datum_izdaje, izdajatelj, davčna_številka, davčni_zavezanec, številka_dobavitelja, odpremiti_naslovu, količina, enota_mere, predmet_naročila, odpremljeno, naročeno, datum_naročila.
2. Določitev ključa relacije	številka_naročila
3. Ponavljajoče se skupine podatkovnih elementov	Prva skupina: številka_naročila, datum_izdaje, izdajatelj, davčna_številka, davčni_zavezanec, številka_dobavitelja, odpremiti_naslovu. druga skupina: količina, enota_mere, predmet_naročila, odpremljeno.
4. Prevedba nenormalizirane relacije v prvo normalno obliko	Nov ključ je številka_predmeta_naročila.
5. Prevedba iz prve normalne oblike v drugo	Odstraniti je treba odvisnosti od delov ključa. številka_predmeta_naročila. Za vsako relacijo s ključem za vsak atribut je treba preveriti, ali je atribut odvisen od ključa. Če ni odvisen ga izločimo v novo relacijo.
6. Prevedba iz druge normalne oblike v tretjo	številka_naročila, številka_predmeta_naročila, datum_izdaje, izdajatelj, davčna_številka, davčni_zavezanec, številka_dobavitelja, odpremiti_naslovu, naročeno, datum_naročila
Številka_dobavitelja	Podatek se prebere iz finančne baze
Dokument: pogodba	
Opis aktivnosti	Predlog rešitve
1. Podatkovni elementi pogodbe	številka_pogodbe, naročnik, naslov_naročnika, matična_številka, davčna_številka, številka_dobavitelja, številka_člena_pogodbe, opombe_člena, opis_člena, datum_dobave, datum_naročila
2. Določitev ključa relacije	Številka_pogodbe
3. Ponavljajoče se skupine podatkovnih elementov	številka_člena_pogodbe, opombe_člena, opis_člena

se nadaljuje

nadaljevanje

4. Prevedba nenormalizirane relacije v prvo normalno obliko	Nov ključ je številka_člena_pogodbe.
5. Prevedba iz prve normalne oblike v drugo	Ni potrebno.
6. Prevedba iz druge normalne oblike v tretjo. številka_pogodbe, naročnik, naslov_naročnika, matična_številka, davčna_številka, številka_dobavitelja, številka_člena_pogodbe, datum_dobave, datum_naročila.	
Številka_dobavitelja	Podatek se prebere iz finančne baze
Dokument: račun	
Opis aktivnosti	Predlog rešitve
1. Podatkovni elementi pogodbe	številka_računa, številka_predloga, številka_naročila, številka_pogodbe, vodja_fakture, znesek_ddv, znesek_brez_ddv, skupaj_znesek, datum_knjiženja
2. Določitev ključa relacije	številka_računa
3. Ponavljajoče se skupine podatkovnih elementov	Jih ni.
4. Prevedba nenormalizirane relacije v prvo normalno obliko	Ni potrebno.
5. Prevedba iz prve normalne oblike v drugo	Ni potrebno.
6. Prevedba iz druge normalne oblike v tretjo Ni vzajemno soodvisnih elementov.	

3.4.2 Proučitev podatkovnega modela

Normaliziran podatkovni model (Tabela 21) za programsko rešitev vsebuje primarne tabele: »predlogi«, »naročilnica«, »pogodba« in »račun«. Ostale podatkovne tabele so: stranke, finančne_postavke, naziv_dokumenta, postavke_naročila, predmet_naročila.

Tabela 21: Proučitev podatkovnega modela

Opis aktivnosti	Predlog rešitve
-----------------	-----------------

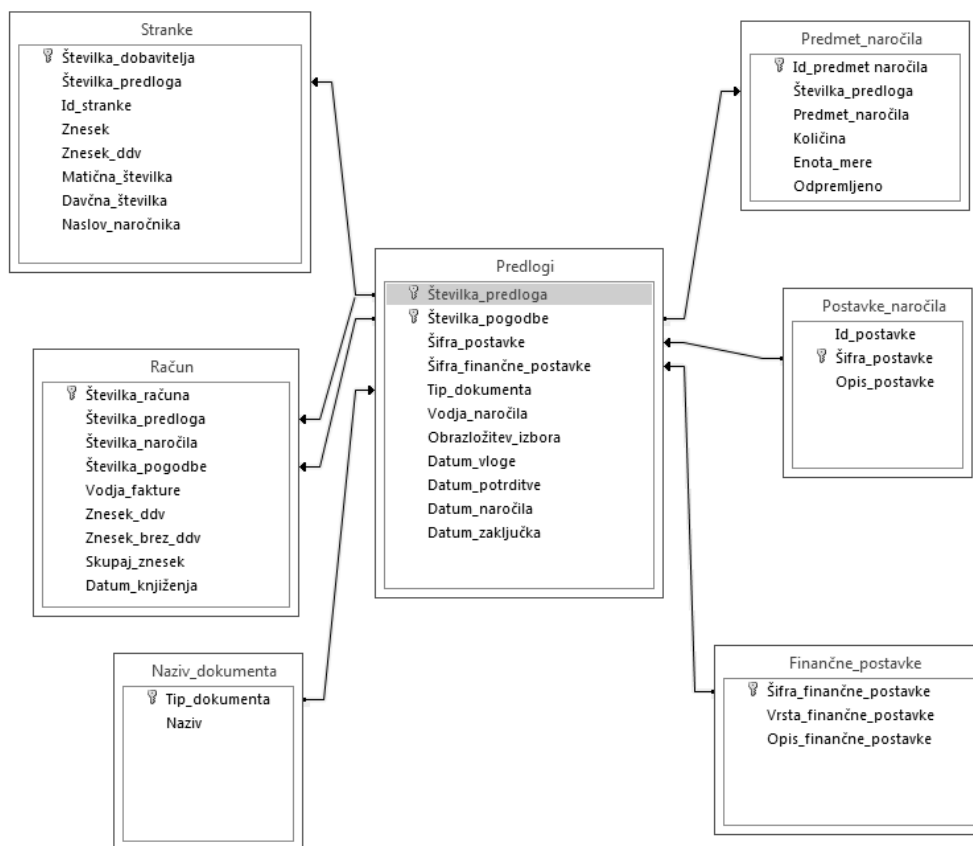
se nadaljuje

nadaljevanje

1. Združitev podatkovnih tabel »predlogi«, »naročilnica« in »pogodba«	Relacije med tabelami niso smiselne, saj lahko za en predlog izdamo natanko eno naročilnico oziroma pogodbo.
2. Odstranitev podvojenih atributov tabele »predlogi«	Podvojena atributa sta številka_naročila in številka_dobavitelja.
3. Odstranitev enakopomenskih in enovrednostnih atributov	Enakopomenski atributi: datum_izdaje = datum_naročila in predmet_naročila = številka_predmeta_naročila. Enovrednostni atributi so: izdajatelj, davčna številka, davčni zavezanec, odpremiti_naslovu, odpremljeno, naročeno.
4. Preimenovanje in ustreznost razvrstitev atributov	Nekatere attribute preimenujemo zaradi lažjega razločevanja med predlogom, naročilnico in pogodbo. Attribute razporedimo v logično zaporedje (primarni ključ, tuj ključ ...).

Dokončen podatkovni model je prikazan na Sliki 22.

Slika 22: Podatkovni model programske rešitve



Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (funkcijska specifikacija), 2015c, str. 30.

3.4.3 Prepis starih podatkov v novi podatkovni model

Pred prepisom starih podatkov v novi podatkovni model je bilo treba glede na vsebino logično povezati vse stare attribute z novimi atributi. Pripravljene so bile procedure za prepis podatkov:

- procedura_stranke,
- procedura_predlogi,
- procedura_perdmet_naročila,
- procedura_postavke,
- procedura_računi.

3.5 Izdelava programske rešitve

Programska rešitev za javna naročila po evidenčnem postopku na tržišču ne obstaja kot paket, ki bi ga bilo mogoče kupiti. V primeru, da bi se v izbrani organizaciji odločili za izdelavo programske rešitve v okviru zunanjšega razvoja, bi to predstavljalo bistveno večje stroške kot v primeru notranjega razvoja.

Za izdelavo programske rešitve sem porabil 440 ur dela, kar sem razporedil v okvir treh mesecev. Faze izdelave programske rešitve:

- vzpostavitev testnega okolja,
- pravice uporabnikov,
- urejanje šifrantov,
- urejanja registra strank,
- vnosne maske,
- povezava z dokumentnim sistemom,
- analitične poizvedbe in poročila.

V fazo izdelava programske rešitve lahko uvrstim tudi predhodne korake, ki jih je bilo treba izvesti: izdelava podatkovnega modela, postavitve podatkovne baze in prepis podatkov ter napolnitev podatkovne baze s testnimi podatki.

3.5.1 Vzpostavitev testnega okolja

Vsa programska orodja in strežniško infrastrukturo za vzpostavitev testnega okolja so zagotovili v izbrani organizaciji. Za izdelavo programske rešitve sem uporabil programska orodja MS Visual Studio in MS SQL Server 2008.

Testno okolje je postavljeno na dveh računalnikih, od katerih je eden predstavljen kot spletni in drugi kot aplikacijski strežnik. Na spletnem strežniku so nameščeni: operacijski sistem Windows 7 Professional, razvojno orodje MS Visual Studio in internetni brskalnik Google Chrome. Na aplikacijskem strežniku so nameščeni: MS Windows Server 2008, MS SQL Server 2008, IIS 7.0 in MS Framework 3.5.

3.5.2 Pravice uporabnikov

Namestitev strežnika MS Windows Server 2008 je izvedena kot domenski strežnik, na katerem je nameščen imenik AD. V imeniku AD so kreirani testni uporabniki, ki se lahko prijavijo v izbrano domeno.

Na nivoju programske rešitve je izdelan vmesnik za vpisovanje in spreminjanje pravic uporabnikov, ki so hkrati uporabniki imenika AD in uporabniki aplikacije s pravicami, opisanimi v podpoglavju 4.2.2 v razdelku Uporabniške vloge.

Za vpisovanje pravic uporabnikov se uporabi obstoječa podatkovna tabela (šifrant uporabnikov), ki jo uporabljajo v izbrani organizaciji. Vmesnik za urejanje pravic uporabnikov je prikazan na Sliki 23.

Slika 23: Vmesnik za urejanje pravice uporabnikov

Vir: Izbrana organizacija, Javna naročila po evidenčnem postopku (tehnična specifikacija), 2015č, str. 22.

3.5.3 Urejanje šifrantov

Šifranti združujejo nabor objektov za pregled in spreminjanje šifrantov, ki so osnova za uporabo skupnih podatkov na več mestih in omogočajo povezovanje med različnimi podatkovnimi skupinami, poročanje in analize podatkov. Šifrante lahko v programu urejajo le uporabniki z ustreznimi delovnimi pooblastili, pregled in uporaba pa sta praviloma omogočena vsem uporabnikom.

Šifrante v programski rešitvi urejajo administratorji. Program mora preprečiti izbris zapisov, ki imajo relacijo s katerim koli zapisom v podatkovni bazi. Dovolj se sprememba, ki mora imeti za posledico ustrezne spremembe v vseh referenčnih tabelah. Referenčna integriteta mora biti implementirana na ravni podatkovne baze in ne na ravni programske logike. Programska rešitev ima razvite vmesnike za pregled in spreminjanje šifrantov:

- šifrant uporabnikov,
- šifrant organizacijskih enot,
- šifrant strank,
- šifrant finančnih postavk,
- šifrant postavk naročil,
- šifrant vrst dokumentov.

Vmesnik za šifrant uporabnikov je opisan v predhodni točki 4.5.2. Šifrant organizacijskih enot je narejen v obliki seznama, ki prikazuje vse organizacijske enote izbrane organizacije. Urejanje poteka z gumbom »uredi«. Uporabnik ima možnost dodajanja novih organizacijskih enot, brisanje ni mogoče. Za potrditev vnesenih podatkov je na voljo gumb »shrani«. Šifrant strank se prepisuje iz SAOP. Izdelan je tudi vmesnik za spreminjanje podatkov o stranki v podatkovni bazi SAOP. Šifrant finančnih postavk se prepisuje iz SAOP. Izdelan je tudi vmesnik za spreminjanje podatkov o finančnih postavkah v podatkovni bazi SAOP. Šifrant postavk naročil je narejen v obliki seznama, ki prikazuje postavke naročil. Urejanje poteka z gumbom »uredi«. Uporabnik ima možnost dodajanja novih postavk naročil, brisanje ni mogoče. Za potrditev vnesenih podatkov je na voljo gumb »shrani«. Šifrant vrst dokumentov je narejen v obliki seznama, ki prikazuje seznam vseh vrst dokumentov. Urejanje poteka z gumbom »uredi«. Uporabnik ima možnost dodajanja novih vrst dokumentov, brisanje ni mogoče. Za potrditev vnesenih podatkov je na voljo gumb »shrani«.

3.5.4 Urejanje registra strank

Ker finančne stranke v aplikaciji MS Access niso bile povezane s finančnimi strankami v SAOP, je bilo treba stranke predhodno poenotiti. V aplikaciji MS Access namreč ni bilo vgrajene nobene kontrole pri vpisu strank, zato so se te multiplicirale. V razviti programske rešitvi je enolični identifikator stranke enotna matična številka ali enotna davčna številka iz SAOP.

Stare stranke iz aplikacije MS Access so usklajene z novimi strankami v SAOP. V primeru, da stranke ni bilo mogoče poenotiti s stranko v SAOP, programska rešitev ohranja zapis stranke. Poenotenih je približno 30.000 strank.

3.5.5 Vnosne maske

3.5.6 Povezava z dokumentnim sistemom

Povezava z dokumentnim sistemom je vzpostavljena prek gumba »dodaj dokument«. V dokumentni sistem se lahko dodajajo vsi dokumenti iz programske rešitve. Dokumenti se vpišejo pod določeno zaporedno klasifikacijsko številko. Nova zadeva se vedno ustvari v dokumentnem sistemu, šele potem je mogoče dodajanje dokumentov iz programske rešitve za javna naročila po evidenčnem postopku. Dokument se doda prek vnosne maske »dodajanje dokumenta v dokumentni sistem«. Uporabnik vpiše številko zadeve, na katero želi pripeti dokument, in potrdi zahtevo z gumbom »shrani«. V aplikaciji je dodan tudi gumb »odpri dokumentni sistem« za neposredno povezavo z aplikacijo dokumentnega sistema.

3.5.7 Analitične poizvedbe in poročila

Analitične poizvedbe in poročila združujejo izpise različnih vrst, ki jih potrebujejo uporabniki za svoje delo in odločanje. Nekatera pogosto uporabljena poročila, ki imajo standardizirano obliko, so vgrajena v sistem na način, ki omogoča oblikovan izpis na papir, nekatera so izdelana bolj v smislu filtriranja in prikaza podatkov na zaslonu. Vsa pa omogočajo izpis podatkov v datoteki .pdf in nadaljnjo obdelavo v zunanji aplikaciji (npr. MS Excel).

3.6 Testiranje rešitve

Testiranje rešitve je bilo izvedeno v postavljenem testnem okolju v izbrani organizaciji, opisanem v podpoglavju 4.5.1. Pri testiranju programske rešitve sem se osredotočil na:

- pravilnost delovanja glede na opisane funkcionalne zahteve,
- postopke nameščanja programske rešitve in ostale programske opreme,
- delovanje vseh kontrol in varnosti,
- delovanje v povezavi z ostalimi aplikacijami,
- odzivnost programske rešitve pri določenih obremenitvah.

V okviru testiranja funkcionalnosti so bile znova natančno preverjene vse opisane funkcionalnosti, ki so bile podane. Preverjeni so bili vsi uporabniški vmesniki, ustrezne funkcije v programski kodi in procedure v podatkovni bazi. Vnovič sem preveril postopke namestitve programske rešitve in ostalo programsko opremo, ki je nujno potrebna za delovanje:

- napolnitev podatkovne baze s pripravljenimi podatki predhodnega stanja,
- konfiguracija IIS 7.0,
- namestitev MS Internet Explorerja in Google Chroma.

Na nivoju podatkovne baze in uporabniških vmesnikov sem preveril delovanje vseh kontrol. Preverjeni so bili podatkovni tipi in možnosti za vnos napačnih podatkov na nivoju programske rešitve, avtorizacija uporabnikov in prikazovanje različnih nivojev glede na pravice uporabnikov ter vse povezave, ki se uporabljajo v povezavi s SAOP in z aplikacijo za dokumentni sistem.

Izvedeni so bili določeni testi pri različnih obremenitvah programske rešitve in podatkovne baze. Obremenitvene teste sem izvedel tako, da sem na istem računalniku simuliral več uporabnikov uporabniške rešitve in meril odzivne čase. Teste sem izvedel za primer, ko programsko rešitev hkrati uporablja deset uporabnikov. Na nivoju podatkovne baze sem kreiral podobne procedure, kot bi jih izvajal uporabnik na nivoju programske rešitve za različne podatkovne tabele. Pri izvajanju testov odzivnosti sem za različne poizvedbe meril:

- čas trajanja poizvedbe na nivoju uporabniške rešitve in podatkovne baze;
- porabo sistemskih virov procesorja in spomina na strani odjemalca in podatkovnega strežnika.

3.7 Izdelava dokumentacije

Dokumentacija je bila izdelana glede na posamezne faze razvoja programske rešitve in opisuje programsko rešitev, ostalo programsko opremo, vsebinske in tehnične lastnosti, uporabniška navodila in navodila za namestitvev. Dokumentacija je namenjena tako vzdrževalcem uporabniške rešitve kot uporabnikom.

Pri izdelavi dokumentacije sem upošteval, da vsi dokumenti vsebujejo: naslovnico, kazalo, seznam slik, uvod, predstavitev in slovar.

V naslovnico vsakega dokumenta sem vključil podatke: naslov projekta, različica, avtor, datum zadnjega popravka, tip dokumenta, kdo je dokument odobril, datum odobritve, uporabniki dokumenta, ključne besede, pravice razmnoževanja in simbol organizacije. Izdelana je bila dokumentacija, ki je prikazana v Tabeli 22.

Za namen spremljanja projekta je bila pripravljena drevesna struktura z naslednjo vsebino:

- <docx> (vsa dokumentacija tipa *.doc),
- <deploy> (vse namestitvene datoteke programske rešitve),
- <src> (vsa izvorna koda programske rešitve vključno s testno kodo),
- <baza> (izvorna koda temeljnih objektov, skripte za namestitvev).

Tabela 22: Izdelana dokumentacija

Naziv dokumenta	Vsebina
Javna naročila po evidenčnem postopku (vzpostavitev projekta)	Vključuje ključne informacije za izvedbo projekta. Določene so podrobnosti glede organizacije projekta, sodelujočih, terminskega plana in finančne opredelitve projekta.
Javna naročila po evidenčnem postopku (funkcijska specifikacija)	Predstavlja načrt izgradnje programske rešitve. V načrtu so opisani poslovni proces, posamezni postopki in pravila, uporabniški vmesnik, podatkovni model ipd.
Javna naročila po evidenčnem postopku (specifikacija dizajna)	Opisuje predlog izdelave programske rešitve.
Javna naročila po evidenčnem postopku (tehnična dokumentacija)	Opisuje izdelan uporabniški vmesnik programske rešitve.
Javna naročila po evidenčnem postopku (sistemska dokumentacija)	Opisuje celotno okolje, ki je pomembno za delovanje programske rešitve. Opisane so namestitve: programske rešitve podatkovnega strežnika in podatkovnega modela, konfiguracija IIS 7.0 ipd.
Javna naročila po evidenčnem postopku (uporabniška navodila)	Opisana so podrobna uporabniška navodila za uporabo programske rešitve (opis uporabniškega vmesnika, posameznih menijev, pravice uporabnikov ...).

3.8 Usposabljanje uporabnikov

Za uporabo programske rešitve je bilo treba usposobiti 44 uporabnikov. Izvedene so bile tri vrste usposabljanj:

- informativno usposabljanje,
- usposabljanje uporabnikov aplikacije,
- usposabljanje vzdrževalcev informacijskega sistema.

Objavljena in vsem zaposlenim predana je bila vsa dokumentacija izdelane programske rešitve. Dokumentacijo so zaposleni prejeli vsaj en teden pred prvim predvidenim izobraževalnim tečajem.

Najprej so bili izvedeni enodnevni tečaji za uporabnike programske rešitve. Opremo in prostore za izobraževanje so zagotovili v organizaciji. Zaposlenim v oddelku za informacijsko tehnologijo je bila v okviru enodnevnega tečaja predstavljena sistemska

dokumentacija. Tečajji so bili izvedeni v obliki predavanj s predstavitvijo prek projektorja. V informacijski sistem za prijavo napak je bila vključena tudi prijava napak s področja uporabe programske rešitve za javna naročila po evidenčnem postopku.

3.9 Prehod v produkcijo

Prehod v produkcijo je bil izveden po naslednjem scenariju:

- prepis podatkov na testno podatkovno bazo,
- namestitev testne različice programske rešitve,
- preverjanje primernosti IIS 7.0, Internet Explorerja in Google Chroma,
- potrditev testne različice programske rešitve in odprava napak,
- prepis podatkov v produkcijsko bazo,
- namestitev uradne različice programske rešitve.

Za prepis podatkov iz podatkovne baze MS Access in MS SQL 2008 so bile pripravljene procedure, ki so samodejno prepisale podatke v testno podatkovno bazo. Zagon procedur za namestitev testne baze se je izvedel v popoldanskem času.

Nameščena je bila testna različica programske rešitve. V testno različico so bili glede na pravice vpisani vsi uporabniki programske rešitve, ki so bili obenem tudi domenski uporabniki. Preverjena je bila konfiguracija MS IIS 7.0 in različice programskih orodij MS Internet Explorer in Google Chrome na strani odjemalcev, torej za vsak računalnik posebej.

Oblikovana je bila testna skupina, ki je testirala vse funkcionalnosti programske rešitve in poročala o napakah v delovanju. Napake, ki so se pojavile pri testiranju, so bile odpravljane, pripombe glede dopolnitev so bili uvrščene v načrt nadgradenj programske rešitve.

Ob potrditvi testne različice programske rešitve oziroma ob odpravi vseh pomanjkljivosti se je pripravil načrt prehoda v produkcijo. Izveden je bil prepis zadnjega stanja podatkov v novi podatkovni strežnik in nameščena uradna različica programske rešitve.

3.10 Vzdrževanje programske rešitve

Redno vzdrževanje programske rešitve je začelo teči z dnem uvedbe v produkcijo. Zagotavlja se za obdobje dveh let. V tem času so zagotovljene štiri ure za vzdrževanje programske rešitve mesečno.

SKLEP

Da bi se laže odločili, kateremu izvajalcu zaupati izdelavo informacijskega sistema, je treba zelo natančno določiti korake in aktivnosti izgradnje informacijskega sistema, kar predstavlja izhodišče za izbiro primerne metodologije razvoja. Prva faza pri izdelavi informacijskega sistema je vedno analiza, ki je najpomembnejša, saj predstavlja osnovo za določitev natančnih potreb pri izdelavi informacijskega sistema. Pomembno je, da pri analizi sodeluje čim večje število zaposlenih, saj se le na ta način lahko natančno opišejo obravnavani poslovni procesi organizacije in opredelijo morebitne spremembe. V fazi načrtovanja se izdelava podroben načrt uvedbe informacijskega sistema, ki poleg opisa vseh funkcionalnosti vsebuje stroškovni in terminski načrt projekta. Treba je poudariti, da natančen načrt izvedbe v veliki meri določa, ali bo izvedba informacijskega sistema uspešna ali ne. V fazi izvedbe projekta se informacijski sistem implementira in izvedejo se vsa potrebna testiranja in odpravijo morebitne napake. Pomembno v tej fazi je, da se pripravijo in izvedejo določeni scenariji testiranja, da imamo potem pri prehodu v produkcijo čim manj težav. V fazi prehoda v produkcijo izvedemo vse namestitve ostale potrebne programske opreme in uvedemo nov informacijski sistem. Najpogostejši prehod v produkcijo, ki sem ga uporabil tudi v našem primeru, je neposredni prehod, ki pa je tudi najbolj tvegan. Tovrsten prehod mora biti skrbno načrtovan, vključno z najmanjšimi podrobnostmi prehoda.

Lahko bi rekel, da so zaposleni v organizaciji tisti, ki določijo vsebino poslovnih procesov, informatiki pa vsebino »prevedejo« v računalniško obliko oziroma v informacijski sistem. Uspešna »prevedba« vsebine v informacijsko obliko pa je odvisna od znanja, ki ga imajo informatiki.

Zavedati se je tudi treba, da se trg informacijskih rešitev razvija neverjetno hitro, zato se tudi dogaja, da je pri iskanju kadrov zelo težko dobiti dobrega informatika. Vodstva v različnih organizacijah od informatikov zahtevajo podrobna vsebinska znanja s področja poslovnih procesov in informatike. Informatik mora imeti različna znanja s področja upravljanja, informacijske tehnologije in poslovanja. Informatik v organizaciji naj bi torej imel širok nabor različnih znanj, kar pa je nemogoče zagotoviti v kratkem času.

Dandanes je značilno, da se v različnih organizacijah in tudi v javni upravi vse bolj poslužujejo zunanjih izvajalcev s področja informatike. Eden izmed vzrokov je prav gotovo ta, da organizacije vse manj vlagajo v znanje notranje zaposlenih informatikov in se namesto tega poslužujejo dragih zunanjih izvajalcev. V našem primeru sem nazorno prikazal, da zna biti zunanji razvoj programske rešitve približno štirikrat dražji od notranjega razvoja ob predpostavki, da imajo zunanji razvijalci programskih rešitev enako vsebinsko znanje z obravnavanega področja kot razvijalci programskih rešitev v izbrani organizaciji. Razvijalci programskih rešitev pa vemo, da to največkrat ne drži. Zunanji razvijalci programskih rešitev v postopku analize lahko porabijo neprimerno več časa, da se seznanijo z obravnavano vsebino, kot notranji razvijalci. Predvsem v različnih javnih organizacijah je opaziti trend

ukinjanja delovnih mest razvijalcev programskih rešitev. Vodstva se raje odločajo za najem dragih zunanjih izvajalcev, kar pa dolgoročno lahko predstavlja veliko težav pri vzdrževanju programskih rešitev. Težave niso samo visoki stroški vzdrževanja, temveč tudi iskanje primernih vzdrževalcev za vzdrževanje programske rešitve ob preteku vzdrževalnih pogodb. Vse to je po moji presoji posledica neustreznega poslovnega načrta organizacije, kar pomeni, da se vodstvo ne zaveda pomembnosti notranjega razvoja informatike, kar pa je odraz nekompetentnosti vodstva.

V organizacijah prav zaradi pomankanja strokovnega znanja s področja informatike niso sposobni sami opredeliti zahteve za izdelavo informacijskega sistema. Prav tako se lahko zgodi, da zunanji izvajalci ne znajo pravilno opredeliti zahteve naročnika, ampak se zanašajo na svoj poslovni model razvoja programske rešitve. V takem primeru načrtovanje programske rešitve poteka neustrezno, zato je tudi razvita programska rešitev neprimerna in draga za vzdrževanje zaradi nenehnih popravkov oziroma dopolnitev. Dolgoročno sodelovanje naročnika in izvajalca programske rešitve je tako že v samem začetku vprašljivo.

Konkreten projekt izgradnje informacijskega sistema, bi lahko za marsikatero organizacijo, ki se srečuje s podobno problematiko predstavljal ustrezen model reševanja.

Pri razvoju informacijske rešitve je zelo pomembno vodenje. To pomeni, da je že na začetku treba določiti vloge zaposlenih v projektu. Pri razvoju programske rešitve za javna naročila po evidenčnem postopku so sodelovali vodja projekta, nadzorni organ in vsebinski razvijalci. Kot vodja projekta in glavni razvijalec sem spoznal, da je zelo pomembna organizacija projekta, stopnja vključenosti zaposlenih, motivacija in komunikacija. Rečem lahko, da sem užival popolno zaupanje nadzornikov projekta in vsebinskih razvijalcev, kar je bilo tudi ključno, da se je projekt uspešno zaključil, saj so bili zastavljeni cilji izpolnjeni.

LITERATURA IN VIRI

1. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile Software Development Methods: Review and Analysis*. Vuorimiehentie: VTT Publications.
2. Avison, E. D., & Fitzgerald, G. (1995). *Information Systems Development: Methodologies, Techniques and Tools* (2nd ed.). London: McGraw-Hill.
3. *Bagya Tech*. Najdeno 18. februarja 2016 na spletnem naslovu <http://www.bagyatech.com/advantages-and-disadvantages-of-spiral-model/>
4. Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. *Journal Computer*, 21(5), 61–72.
5. Boehm, A. (2007). *Murach's Ado.net 2.0 database programming with VB 2005*. Mike Murach & Associates, Inc.
6. Boehm, B., & Turner, R. (2004). *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston: Pearson Education, Inc.
7. Bucanac, C. (1999). *The V-Model*. Karlskrona: University of Karlskrona/Ronneby.
8. Colnar, M. (2000). Strateško planiranje razvoja ali prenove informacijskih sistemov v državnih organih Republike Slovenije. *Uporabna informatika*, 8(1), 42–46.
9. Components of .Net Framework. (b.l.). V *DeveloperIn.net*. Najdeno 20. januarja 2016 na spletnem naslovu <http://www.developerin.net/a/39-Intro-to-.Net-FrameWork/23-Components-of-.Net-Framework>
10. Crinnion, J. (1991). *Evolutionary Systems Development, a practical guide to the use of prototyping within a structured systems methodology*. New York: Plenum Press.
11. Evjen, B., Hollis, B., Sheldon, B., & Sharkey, K. (2008). *Professional Visual Basic 2008*. 10475 Crosspoint Boulevard Indianapolis: Wiley Publishing, Inc.
12. Gospodarska zbornica Slovenije. (2008, 23. april). Referenčni cenik storitev s področja informacijskih storitev. Najdeno 5. marca 2016 na spletnem naslovu <https://www.gzs.si/pripone/oei31701d19151a511a9550a.pdf>
13. Gradišar, M. (2003). *Uvod v informatiko*. Ljubljana: Ekonomska fakulteta.
14. Gradišar, M., & Resinovič, G. (2001). *Informatika v poslovnem okolju*. Ljubljana: Ekonomska fakulteta.
15. Gradišar, M., Jaklič J., & Turk, T. (2007). *Osnove poslovne informatike*. Ljubljana: Ekonomska fakulteta.
16. Gričar, J. (1994) Reorganiziranje procesov kot sestavina spreminjanja organizacij, *Uporabna informatika*, 2(2), 12–16.
17. Gričar, J. (2002). *Poslovni informacijski sistem*. Kranj: Fakulteta za organizacijske vede.
18. Guimarães, L. R., & Vilela, P. R. S (2005). *Comparing software development models using CDM*. New Jersey: Newark, Sigite.
19. *I answer 4 U*. Najdeno 18. februarja 2016 na spletnem naslovu <http://www.ianswer4u.com/2011/11/prototype-model-advantages-and.html#axzz420XMCboG>
20. IIS, Web Server and Windows Part 1. (b.l.). V *Tenouk*. Najdeno 12. januarja 2016 na spletnem naslovu <http://www.tenouk.com/download/pdf/setupiisnweb.pdf>

21. Incremental Model. (2008, 8 November). V *Testing Excellence*. Najdeno 18. februarja 2016 na spletnem naslovu <http://www.testingexcellence.com/incremental-model/>
22. Isaias, P., & Issa, T. (2015). *High Level Models and Methodologies for Information Systems*. Perth: Springer.
23. Izbrana organizacija. (2015a). *Poročilo o delu in zaključni račun za leto 2014*. Ljubljana: Izbrana organizacija.
24. Izbrana organizacija. (2015b). *Navodilo za oddajo naročil pod mejo zakonskih vrednosti ZJN-2* (interno gradivo). Škofja Loka: Izbrana organizacija.
25. Izbrana organizacija. (2015c). *Javna naročila po evidenčnem postopku (funkcijska specifikacija)* (interno gradivo). Škofja Loka: Izbrana organizacija.
26. Izbrana organizacija. (2015č). *Javna naročila po evidenčnem postopku (tehnična dokumentacija)* (interno gradivo). Škofja Loka: Izbrana organizacija.
27. Izbrana organizacija. (2015d). *Javna naročila po evidenčnem postopku (uporabniška navodila)* (interno gradivo). Škofja Loka: Izbrana organizacija.
28. Izbrana organizacija. (2015e). *Javna naročila po evidenčnem postopku (specifikacija dizajna)* (interno gradivo). Škofja Loka: Izbrana organizacija.
29. Javornik, M., Železnik, M., & Čerič, D. (2006). *Priročnik za izvajanje javnih naročil*. Maribor: Inštitut za lokalno samoupravo in javna naročila.
30. Jayaswal, B. K., & Patton, P. C. (2006). *Software Development Methodology Today*. Ney Jersey: Prentice Hall.
31. Johnson, B. (2013). *Professional Visual Studio 2012*. 10475 Crosspoint Boulevard Indianapolis: John Wiley & Sons, Inc.
32. Kosmač, H. (2010, 24. maj). Predstavitev novosti na področju javnega naročanja. Najdeno 12. januarja 2016 na spletnem naslovu http://www.zdruzenjeobcin.si/e_files/datoteke/18/ZJN-2-K-%20HELENA%20KOSMAC%2024.5.2010.ppt
33. Kovačič, A. (1998). *Informatizacija poslovanja*. Ljubljana: Ekonomska fakulteta.
34. Kovačič, A., & Vintar, M. (1994). *Načrtovanje in gradnja informacijskih sistemov*. Ljubljana: DZS.
35. Kranjc, M. (2003). Kako optimizirati IS?. *Monitor, priloga Sistem, oktober*, 22–24.
36. LeBlanc, P. (2013). *Microsoft SQL Server 2012 Step by Step*. California: O'Reilly Media.
37. MacDonald, M. (2006). *The book of Visual Basic 2005 .net insight for classic VB developers*. William Pollock.
38. McConnell, S. (2004). *Code Complete* (2nd ed.). Washington: Microsoft Press.
39. Mesojedec, U. (1998). *Visual Basic: sodobno programiranje*. Ljubljana: Pasadena.
40. Michelbrink, P. (2004, 15 januar). Visual Basic .NET Tutorial. Najdeno 12. januarja 2016 na spletnem naslovu <https://www.iusb.edu/computerscience/employment/docs/VisualStack.pdf>
41. Microsoft. (2012), *Visual Studio 2012 Product Guide*. Redmond: Microsoft.
42. Microsoft. (2016) Najdeno 20. januarja 2016 na spletnem naslovu <https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx>
43. Moore, A. (2010). *Visual Studio 2010 All in One for Dummies*. Indianapolis (IN): Willey Publishing, Inc.

44. Nusdorfer, A. (2004). *Informacijski sistem za javna naročila* (magistrsko delo). Ljubljana: Ekonomska fakulteta.
45. Parnas, D. (1986). A Rational Design Process. How and why to fake it. *Ieee transactions on software engineering*, 12(2), 251–257.
46. PCMAG. (b. l.). V *PC Encyclopedia*. Najdeno 12. januarja 2016 na spletnem naslovu <http://www.pcmag.com/encyclopedia/term/54342/web-server>
47. Primec, B. (2008). *Zakon o javnih naročilih s komentarjem*. Ljubljana: Bonex.
48. Randolph, N., & Gardner, D. (2008). *Professional Visual Studio 2008*. 10475 Crosspoint Boulevard Indianapolis: Wiley Publishing, Inc.
49. *Software Development Life Cycle*. (2016). Najdeno 18. februarja 2016 na spletnem naslovu <https://www.sdlc.ws/incremental-model/>
50. *Software Testing Help*. (2016). Najdeno 22. februarja 2016 na spletnem naslovu <http://www.softwaretestinghelp.com/>
51. Srabotič, R. (2002). *Strateško načrtovanje integriranih informacijskih sistemov v slovenskih majhnih in srednje velikih podjetjih* (magistrsko delo). Ljubljana: Ekonomska fakulteta.
52. Stal, M. (2008, 7 november). *Why the waterfall model does not work*. *Blogspot*. Najdeno 5. januarja 2016 na spletnem naslovu www.stal.blogspot.si
53. Stanek, W. R. (2008). *Internet Information Services (IIS) 7.0*. Washington: Microsoft Press.
54. Syverson, B., & Murach, J. (2008). *Murach's SQL Server 2008 for developers*. B. k.: Mike Murach & Associates, Inc.
55. Šuler, A. (2012). *Spoznajmo Visual Basic .NET 2010*. Šempeter pri Gorici: Flamingo.
56. *TutorialsPoint*. Najdeno 12. januarja 2016 na spletnem naslovu <http://www.tutorialspoint.com/>
57. Tylee, L. (1998). *Learn Visual Basic 6.0*. Bellevue: KIDware.
58. Web Server. (b. l.). V *Webopedia*. Najdeno 12. januarja 2016 na spletnem naslovu http://www.webopedia.com/TERM/W/Web_server.html
59. Web Server. (b. l.). V *Technopedia*. Najdeno 12. januarja 2016 na spletnem naslovu <https://www.techopedia.com/definition/4928/web-server>
60. Web Server. (b. l.). V *TechTarget*. Najdeno 12. januarja 2016 na spletnem naslovu <http://whatis.techtarget.com/definition/Web-server>
61. Werber, B., & Zupančič, J. (2002). Ključni dejavniki uspeha informacijskih sistemov v malih podjetjih v Sloveniji. *Organizacija*, 35(2), 82–89.
62. Werber, B., & Zupančič, J. (2002). Uporaba informacijske tehnologije v malih podjetjih v Sloveniji. *Organizacija*, 35(1), 23–32.
63. Winston, W. R. (1970). Managing the Development of Large Software Systems. *IEEE WESCON*, avgust, 328–338.
64. Zakon o javnem naročanju – ZJN-1. *Uradni list RS* št. 39/2000, 102/2000, 2/04.
65. Zakon o javnem naročanju – ZJN-2. *Uradni list RS* št. 128/2006, 16/08, 19/10, 18/11, 43/12, 90/12, 12/13, 19/14, 90/14.

66. Zornada, L. (2002). Razvoj informacijskega sistema – od strateškega načrta do realizacije. *Strokovni posvet Visoke šole za management v Kopru z mednarodno udeležbo* (str. 223–236). Koper: Visoka šola za management v Kopru.

PRILOGE

Priloga 1: Seznam kratic

AD	Active Directory
ASP	Active Server Pages
C	programski jezik C
C#	programski jezik C#
C++	programski jezik C++
CLR	Common Language Runtime
CLS	Common Language Specification
CSS	Cascading Style Sheets
CTS	Common Type System
DDV	davek na dodano vrednost
F#	programski jezik F#
FCL	Framework Class Library
GUI	Graphical User Interface: grafični uporabniški vmesnik
GWS	Google Web Server
HKOM	državno komunikacijsko omrežje
HP	Hewlett-Packard
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
IDE	Integrated Development Environment
IIS	Internet Information Services
IP	Internet Protocol
MS	Microsoft Corporation
MSIL	Microsoft Intermediate Language
MVC	Model View Controller
.NET	ogrodje za razvijanje programske opreme
NMV	naročila male vrednosti
OSI	Open Systems Interconnection Model
PHP	Personal Home Page Tools: programski jezik za spletne strani
SAOP	računovodska poslovna aplikacija
SQL	Structured Query Language: strukturirani poizvedovalni jezik
TCP/IP	Transmission Control Protocol/Internet Protocol
VB	Visual Basic: programski jezik
WEB	splet
XHTML	Extensible Hypertext Markup Language: označevalni jezik, ki je prilagojen XML
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations
ZJN	Zakon o javnem naročanju