

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

**ANALIZA INFORMACIJSKE REŠITVE ZA E-RAČUNE V IZBRANEM
PODJETJU**

Ljubljana, julij 2016

ŠTEFAN ZUPANČIČ

IZJAVA O AVTORSTVU

Podpisani Štefan Zupančič, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom ANALIZA INFORMACIJSKE REŠITVE ZA E-RAČUNE V IZBRANEM PODJETJU, pripravljenega v sodelovanju s svetovalcem prof. dr. Mirkom Gradišarjem,

IZJAVLJAM,

1. da sem predloženo delo pripravil samostojno;
2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;
3. da je besedilo predloženega dela 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 oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;
4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;
5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;
6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;
7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;
8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;
10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne 20. julij 2016

Podpis študenta:

KAZALO

UVOD	1
1 METODOLOGIJE RAZVOJA INFORMACIJSKEGA SISTEMA	5
1.1 Delitev metodologij	6
1.1.1 Delitev metodologij glede na njihov tip.....	7
1.1.2 Delitev metodologij glede na njihovo težo	9
1.1.3 Delitev metodologij glede na njihovo utežitev	11
1.2 Tradicionalne metodologije.....	13
1.2.1 Zaporedni ali slapovni model.....	14
1.2.2 Iterativni model	16
1.2.3 Prototipni model.....	17
1.2.4 Inkrementalni model	18
1.3 Agilne metodologije	19
1.3.1 Ekstremno programiranje – model XP.....	22
1.3.2 Scrum	24
1.3.3 Model RUP	26
1.3.4 Funkcionalno voden razvoj informacijskih rešitev – model FDD.....	27
1.3.5 Dinamični sistemski model razvoja informacijskih rešitev – model DSDM...	28
1.3.6 Prilagodljivi razvoj programskih rešitev – model ASD.....	29
1.3.7 Metodologije družine Crystal.....	30
1.4 Primerjava tradicionalnih z agilnimi metodologijami	31
1.4.1 Omejitve tradicionalnih metodologij	32
1.4.2 Omejitve agilnih metodologij	34
2 ELEKTRONSKO POSLOVANJE	35
2.1 Oblike in področja elektronskega poslovanja.....	36
2.2 Prednosti in slabosti elektronskega poslovanja	37
2.2.1 Prednosti elektronskega poslovanja	37
2.2.2 Slabosti elektronskega poslovanja	40
2.3 Elektronski račun	41
2.4 Zakonodaja	43
2.4.1 Elektronsko poslovanje in podpis	43
2.4.2 Poslovanje z e-računi	44
2.4.3 Izmenjava elektronskih računov s proračunskimi uporabniki	45
2.4.4 Arhiviranje elektronskih dokumentov	48
3 ANALIZA INFORMACIJSKE REŠITVE	49
3.1 Predstavitev Izbranega podjetja in situacije	49
3.2 Metodologija razvoja v Podjetju za računalniški inženiring	52
3.3 Analiza aplikacije eRačun	54
3.3.1 Shema povezav nove aplikacije	54

3.3.2	Funkcionalnosti nove aplikacije eRačun	55
3.3.3	Opis mask, polj in gumbov	56
3.3.4	Postopek dela s prejetimi računi	58
3.3.5	Postopek dela z izdanimi računi	59
3.3.6	Urejanje prilog računov	60
3.4	Analiza aplikacije ePoslovanje.....	61
3.4.1	Funkcionalnosti nove aplikacije ePoslovanje	62
3.4.2	Delovni tok dokumenta s prehodi stanj in vloge uporabnikov	62
3.4.3	Opis uporabniškega vmesnika in menijev	66
3.4.4	Maska za delo z dokumenti	71
SKLEP.....		74
LITERATURA IN VIRI.....		76

KAZALO TABEL

Tabela 1: Primerjava tradicionalnih metod z agilnimi	32
--	----

KAZALO SLIK

Slika 1: Delitev metodologije glede na tip	7
Slika 2: Določitev teže metodologije	10
Slika 3: Zaporedni ali slapovni model.....	14
Slika 4: Iterativni model	16
Slika 5: Prototipni model – RAD	17
Slika 6: Inkrementalni model	18
Slika 7: Življenjski cikel modela XP.....	24
Slika 8: Model Scrum.....	25
Slika 9: Model RUP	27
Slika 10: Model FDD	28
Slika 11: Model DSDM.....	29
Slika 12: Model ASD	30
Slika 13: Metodologije družine Crystal	31
Slika 14: Shema procesa razvoja v Podjetju za računalniški inženiring.....	53
Slika 15: Shema povezav nove aplikacije	55
Slika 16: Maska za pregled e-računov	57
Slika 17: Maska za pregled prilog.....	60
Slika 18: Maska za pregled prilog.....	61
Slika 19: Delovni tok s prehodi stanj, vlogami in možnimi akcijami	63
Slika 20: Meni domov	67
Slika 21: Meni prejeti računi	68
Slika 22: Meni izpisi	69

Slika 23: Zavihek dokument.....	72
Slika 24: Zavihek DDV	72
Slika 25: Zavihek postavke.....	73
Slika 26: Zavihek priloge	73

UVOD

Opredelitev problema. Podjetja se pri svojem poslovanju dnevno srečujejo z neko obliko metodologije, saj je ta prisotna pri vsaki obliki organiziranega dela. Vse, kar redno počnemo za dosego želenega rezultata, ki je lahko izdelek ali storitev oziroma dosežemo zastavljeni cilj, je zajeto v metodologijo, ki jo pri tem uporabljamo. Za primer razvoja programske opreme to ne pomeni samo postopkov, ki so neposredno povezani z razvojem, temveč zajema tudi podporne postopke, načine komunikacije med sodelujočimi, pravila odločanja in podobno. Gledano s tega vidika lahko metodologijo opredelimo tudi kot množico dogovorov, s katerimi se projektna skupina/organizacija strinja (Krisper et al., 2003).

Sociološko komponento metodologije ponazarja dejstvo, da je metodologija kot takšna prežeta s filozofijo, načeli, idejami in pogledi organizacije ter njenih članov in ni nekaj, kar lahko nastane neodvisno od ljudi, katerim je namenjena. Kljub temu, da obstajajo številne formalne metodologije, ki temeljijo na preizkušenih postopkih, ki so se v praksi izkazali kot dobri, si jih organizacije prilagodijo tako, da ustrezajo njihovem načinu dela ter percepciji domene, za katero so vzpostavljene. Uporabniki se z uporabo metodologije učijo in pridobivajo nove izkušnje, posledično pa se s tem bogati tudi metodologija. Poleg formalno opredeljenih elementov, ki so lahko zapisani v obliki postopkov, pravil, napotkov, smernic, priročnikov itd., so v metodologijo zajeti tudi drugi neformalni elementi, ki niso dokumentirani. Eden od njih je znanje, ki za podjetje predstavlja tudi konkurenčno vrednost. Ločnice med formalnim in neformalnim delom metodologije ni enostavno postaviti. Obsežen formalni del metodologije lahko povzroči težave pri njegovem vzdrževanju, kar lahko privede do zastaranja metodologije in njene neustreznosti dejanskemu stanju. Za organizacijo lahko predstavlja tveganje tudi obsežna neformalna metodologija, saj je ta popolnoma odvisna od njenih uporabnikov. To lahko privede do težav pri standardizaciji metodologije, postopek pa lahko postane nepregleden in posledično neobvladljiv (Bajec & Krisper, b.l., str. 2–4).

Obstaja več metodologij razvoja programske opreme, ki so danes v uporabi. Nekatera podjetja za razvoj programske opreme uporabljajo lastne (prilagojene) metodologije, vendar pa se večinoma govori o uporabi dveh vrst metodologij, in sicer težki (tradicionalni) in lahki (gibki oziroma agilni). Tradicionalna metodologija za razvoj programske opreme predpisuje uporabo celovitega načrtovanja in podrobne dokumentacije. Gibke oziroma agilne metodologije so v zadnjih letih vse bolj priljubljene, saj za razliko od tradicionalnih metod, ki poudarjajo detajlno dokumentacijo, temeljijo na kratkih ponavljajočih se ciklih razvoja in znanju skupine, ki je v njih udeležena (Awad, 2005, str. 2).

Tradicionalne metodologije razvoja programske opreme temeljijo na zaporednem nizu korakov in postopkov, ki predstavljajo opredelitev zahtev, izgradnjo rešitev, testiranje ter

implementacijo. Tradicionalne metodologije že v začetni fazi projekta zahtevajo detajlno opredelitev zahtev in podrobno dokumentacijo, ki se jo upošteva v celotni fazi izgradnje (Awad, 2005, str. 10).

Hvala (2011, str. 30) ugotavlja, da je skupna značilnost tradicionalnih metodologij močna reguliranost in formaliziranost ter dejstvo, da ne dopuščajo nobene svobode pri prilagajanju dejanskim razmeram, v katerih se odvija razvoj programske opreme. Tradicionalne metodologije v vseh fazah razvoja zahtevajo ogromno dokumentacije, saj ta predstavlja komunikacijski kanal med razvijalci in naročnikom. Z napredovanjem po fazah je uvajanje morebitnih sprememb v metodologiji vse težje, predvsem pa dražje. Zaradi striktno določenih pravil in ogromne količine dokumentacije so tradicionalne metodologije časovno precej zahtevne, hkrati pa traja precej časa, da prinesejo želeni rezultat.

Kot rezultat kritičnega pogleda na tradicionalne metodologije, ki razvoj informacijskega sistema predpišejo do najmanjše podrobnosti in so večinoma težko obvladljive, se je pojavil pojem agilnost. Agilne metodologije se v praksi lažje in hitreje implementira, saj so hitro ter enostavno prilagodljive dejanskemu stanju in zahtevajo manj birokracije ter dokumentacije (Bajec & Krisper, b.l., str. 1).

Agilne metodologije temeljijo na iterativnem in inkrementalnem razvoju, pri katerem se tako zahteve kot rešitve razvijajo skozi več ciklov. Njihova značilnost je, da nalogo razbijejo na manjše korake (inkremente), za katere je potrebno samo kratkoročno načrtovanje in nič dolgoročnega (Hvala, 2011, str. 31).

Agilne metodologije so se razvile kot odziv na načela in striktne predpise tradicionalnih metodologij oziroma kot odziv na obsežno birokracijo, ki so jo te zahtevale. Nove metode razvoja programske opreme poskušajo poiskati koristen kompromis v zahtevani količini predpisov, navodil in dokumentacije, ki privede do končnega rezultata. Te metodologije se manj osredotočajo na zahtevano dokumentacijo oziroma je ta za posamezno nalogo precej manjša in pogosto vsebuje tudi programsko kodo (Fowler, 2005).

Doseganje visoke stopnje učinkovitosti in uspešnosti v informacijski družbi je pogojeno z vzpostavitvijo učinkovitih oblik poslovanja, organiziranja, sodelovanja ter povezovanja. Poslovanje je lahko učinkovito le, če je podprto s sodobnimi dosežki s področja informacijskih in telekomunikacijskih tehnologij. Elektronsko poslovanje predstavlja višjo obliko in kakovosten dvig poslovanja, ki ga mora poslovna informatika v informacijski družbi tudi podpreti. Elektronsko poslovanje lahko opredelimo kot način delovanja, organiziranja in notranjega ter zunanjega sodelovanja organizacijskega sistema, ki je podprt z dosežki informacijskih in telekomunikacijskih tehnologij: aplikacijami, storitvami in tehnološko infrastrukturo (Krisper et al., 2003).

Elektronsko poslovanje lahko opredelimo kot uporabo internetnega omrežja za povezavo poslovnih procesov, organizacijsko komuniciranje in sodelovanje podjetja s svojimi strankami, dobavitelji ter drugimi interesnimi skupinami. Elektronsko poslovanje uporablja internet, intranet in druga omrežja za podporo svojih poslovnih procesov (Combe, 2006, str. 1). Chaffey (2009, str. 10–15) ugotavlja, da se izraz elektronsko poslovanje v organizacijah uporablja na dva načina. Prva uporaba izraza se nanaša na koncept, ki se uporablja za strategijo in poslovanje. Druga uporaba izraza pa se nanaša na opis delovanja podjetja, kjer to večinoma posluje prek spleta.

Elektronsko poslovanje (e-poslovanje) se v angleški literaturi pojavlja pod pojmom »electronics commerce« in »e-business«, ki pa se v pomenu razlikujeta. E-commerce v angleški literaturi predstavlja proces nakupa, prodaje, prenosa ali izmenjave izdelkov, storitev in informacij po internetu ter drugih računalniških omrežjih. Takšna opredelitev elektronskega poslovanja je omejena le na procese trgovanja, zato se je pojavil izraz s širšim pomenom, in sicer »e-business«, ki poleg naštetega vključuje tudi servisiranje potrošnikov, sodelovanje s poslovnimi partnerji, učenje na daljavo ter sodelovanje zaposlenih znotraj posamezne organizacije (Gradišar, Jaklič, Damij & Baloh, 2005, str. 138–139).

Elektronski račun (v nadaljevanju e-račun) je listina, ki jo izdajatelj računa izda svojemu dolžniku oziroma prejemniku računa v elektronski obliki in enakovredno zamenjuje račun v papirni obliki. Izmenjava elektronskega računa je proces posredovanja računa od izdajatelja do prejemnika tega računa. Prejemnik je proračunski uporabnik ali fizična ali pravna oseba, ki ji je namenjen e-račun. Izdajatelj e-računa je proračunski uporabnik ali fizična ali pravna oseba, ki takšen račun izda (Uprava Republike Slovenije za javna plačila, b.l.).

Ko je račun izdan in prejet v katerikoli elektronski obliki zapisa lahko govorimo o elektronskem računu. Račun je lahko v obliki strukturiranih sporočil (kot je XML) z ustrezno vizualizacijo ali drugih elektronskih oblikah, kot npr. e-pošta s prilogo PDF ali sporočilo, prejeta po telefaksu v elektronski obliki in ne v papirni obliki. Vendar vsi računi, ki so v elektronski obliki, ne štejejo za elektronske račune. Takšen primer so računi, ustvarjeni z računovodsko programsko opremo ali programsko opremo za obdelavo besedil, ki so poslani in prejeti v papirni obliki. Kot e-računi se štejejo računi, ki so ustvarjeni v papirni obliki in ki se optično preberejo, pošljejo in prejmejo pa prek elektronske pošte. Torej ni pomembna vrsta elektronske oblike računa, ampak dejstvo, da je račun v elektronski obliki, ko je izdan in prejet. To omogoča pošiljanje in prejemanje e-računov v eni obliki zapisa, ki se lahko nato pretvori v drugo (Elektronski računi, 2016).

Uveljavitev Zakona o spremembah in dopolnitvah Zakona o opravljanju plačilnih storitev za proračunske uporabnike je za proračunske uporabnike pomenilo prelomnico v njihovem poslovanju. Zakon je namreč z dnem 1. januar 2015 proračunskim uporabnikom nalagal

obvezen prejem vseh računov v elektronski obliki (Uprava Republike Slovenije za javna plačila, 2014).

Zakon o opravljanju plačilnih storitev za proračunske uporabnike (Ur.l. RS, št. 59/2010, 111/2013, v nadaljevanju ZOPSPU-A) je začel veljati 11. januarja 2014 in proračunskim uporabnikom nalaga, da s 1. januarjem 2015 začnejo prejemati račune in njihovo spremljajočo dokumentacijo izključno v elektronski obliki. Pravne in fizične osebe lahko proračunskim uporabnikom za dobavljeno blago in izvedene storitve pošljejo le račune v elektronski obliki. Izdajatelji neposredno ali prek ponudnika storitev pošiljajo elektronske račune na spletni portal Uprave Republike Slovenije za javna plačila (v nadaljevanju UJP), medtem ko lahko vse javne ustanove izdajajo račune le v elektronski obliki prek UJP.

Namen in cilj dela. Z magistrskim delom smo analizirali trenutne metodologije razvoja informacijskih sistemov, ki jih lahko razvijalci informacijskih rešitev uporabijo pri svojem delu. Analizirali smo tudi različne oblike in področja elektronskega poslovanja organizacij, s posebnim poudarkom na e-računih. Izbrali smo ustrezno metodologijo za primer razvoja aplikacije za e-račune.

Namen magistrskega dela je z agilnimi metodologijami izboljšati obstoječi informacijski sistem.

Končni cilj magistrskega dela je analiza informacijske rešitve, ki bo Izbranemu podjetju omogočala prevzem in obdelavo e-računov, skladno z ZOPSPU-A. Gre za dopolnitev obstoječe informacijske rešitve, ki jo podjetje že uporablja. Rešitev bo omogočala likvidaturo prejetih e-računov glede na nastavljeni delovni tok in stopenjsko obdelavo, hkrati pa bo uporabnikom omogočala, da bodo v celotni fazi obdelave imeli pregled nad vizualizirano sliko elektronskega računa ter morebitnimi spremnimi dokumenti.

Metode dela. Pri izdelavi magistrske naloge je uporabljeno teoretično znanje, pridobljeno v času študija, in praktično znanje z analitičnega področja, pridobljeno na podlagi večletnega dela v podjetju, ki razvija informacijske rešitve.

V prvem delu magistrske naloge je uporabljen poglobljen teoretično-analitičen pregled strokovne literature, knjig, člankov in spletnih virov s področja metodologij razvoja informacijskih sistemov ter elektronskega poslovanja. Pri preučevanju teoretične podlage smo uporabili deskriptivno, komparativno analitično metodo in metodo kompilacije, s katero smo do različnih spoznanj prišli s pomočjo stališč drugih avtorjev. V drugem, empiričnem delu je prikazana analiza informacijske rešitve za elektronske račune.

Celotna magistrska naloga je razdeljena na tri dele. V prvem delu so predstavljene tradicionalne in agilne metodologije razvoja informacijskih sistemov ter primerjava med

obema skupinama metodologij. Teoretično so obravnavane tudi omejitve, ki jih prinaša posamezna metodologija.

Drugi del je namenjen teoretičnemu pregledu elektronskega poslovanja ter oblik in področij elektronskega poslovanja. Teoretično so obdelani zakonski okviri, ki regulirajo elektronske račune in njihovo izmenjavo. Prav tako so prikazane prednosti in slabosti e-računov.

V tretjem delu je predstavljeno Izbrano podjetje in situacija, v kateri se je podjetje znašlo ob uvedbi ZOPSPU-A. Ker je podjetje posredni proračunski uporabnik, je zavezano vse račune prejemati v elektronski obliki, kar je s stališča informacijskega sistema pomenilo dograditev obstoječe rešitve. V tem delu je predstavljena tudi analiza informacijske rešitve, ki podjetju omogoča poslovanje skladno z zakonodajo. Obstoječa likvidatura prejetih računov zaradi obsega dela in velikosti podjetja ter števila uporabnikov, ki so vključeni v likvidaturo, poteka stopenjsko. Analiza rešitve temelji na obstoječem delovnem toku dokumenta oziroma njegovi stopenjski obdelavi, hkrati pa omogoča pregled nad vizualizirano sliko računa in vso spremno dokumentacijo, ki jo uporabniki potrebujejo za uspešno likvidaturo prejetega računa.

1 METODOLOGIJE RAZVOJA INFORMACIJSKEGA SISTEMA

Razvita informatika podjetja oziroma ustrezna informacijska tehnologija, ki je vpeta v podjetje, na trgu predstavlja konkurenčno prednost podjetja. Metodologije razvoja informacijskih sistemov zato stremijo h gradnji sistemov, ki podjetju prinašajo poslovno priložnost, s katero nastopa na trgu in ga razlikuje od njegove konkurence (Kovačič, 1998, str. 40).

Informacijski sistem predstavlja množico ljudi, stroje, postopke in aktivnosti, ki omogočajo pridobivanje ter uporabo informacij, na podlagi katerih se njegovi uporabniki odločajo glede aktivnosti v podjetju. Poslovni informacijski sistem omogoča shranjevanje informacij, njihovo organiziranje in preoblikovanje podatkov v informacije, ki zaposlenim omogočajo sprejemanje odločitev ter izvajanje poslovnih procesov v podjetju (Gradišar et al., str. 40–41). Podobno definicijo postavlja tudi Rupnik (b.l., str. 33), ki informacijski sistem opredeljuje kot množico komponent (strojna oprema, komunikacijska oprema, programska oprema in ljudje), ki so medsebojno odvisne. Te komponente zbirajo, procesirajo, hranijo in porazdeljujejo podatke ter s tem podpirajo temeljne in odločitvene procese v organizaciji.

Informacijski sistem je zasnovan tako, da odpravlja vrsto nalog, pri katerih so rezultati vnaprej definirani oziroma pričakovani. Prenova informacijskega sistema, ki že v obstoječi obliki zagotavlja pravilne in potrebne rezultate, je zahtevno ter dolgotrajno delo skrbno načrtovanih procesov, ki bodo končnemu izdelku zagotovili tako visoko stopnjo integritete

in robustnosti kot tudi sprejetost med njegovimi uporabniki (Yu, Wooi, Wai & Soo, 2012, str. 162).

K uspešni prenovi poslovnih procesov in informatizaciji podjetja veliko prispevajo tudi metodologije razvoja informacijskih sistemov, katerih uporaba omogoča skrajšanje razvojnega časa, lažje vodenje ter obvladovanje projekta in zadovoljstvo uporabnikov po implementaciji rešitve (Zornada, Veselič & Ježovnik, 2002, str. 223).

Metodologija razvoja informacijskega sistema oziroma razvojna metodologija obsega tehnike in postopke, ki se uporabljajo za strukturiranje, načrtovanje ter nadzor nad procesom razvoja informacijskega sistema (Software Development Methodologies, 2016).

Podrobneje metodologijo načrtovanja in izgradnje informacijskih sistemov opredeljujejo tudi Gradišar et al. (2005, str. 267–268), in sicer jo pojmujejo kot skupek tehnik, orodij, postopkov ter dokumentacijskih pripomočkov, ki jih razvijalci uporabljajo pri načrtovanju in implementiranju, hkrati pa omogočajo tudi upravljanje, kontroliranje ter vrednotenje projektov izgradnje informacijskega sistema. Metodologijo sestavljajo posamezne faze in njihove podfaze, ki razvijalcem sistema omogočajo izbiro najprimernejše tehnike izgradnje za vsako fazo projekta.

1.1 Delitev metodologij

V teoriji se pojavljajo različne delitve metodologij. Eno izmed delitev postavlja Gradišar et al. (2005, str. 268–269), ki metodologije razvoja informacijskih sistemov delijo na naslednje skupine:

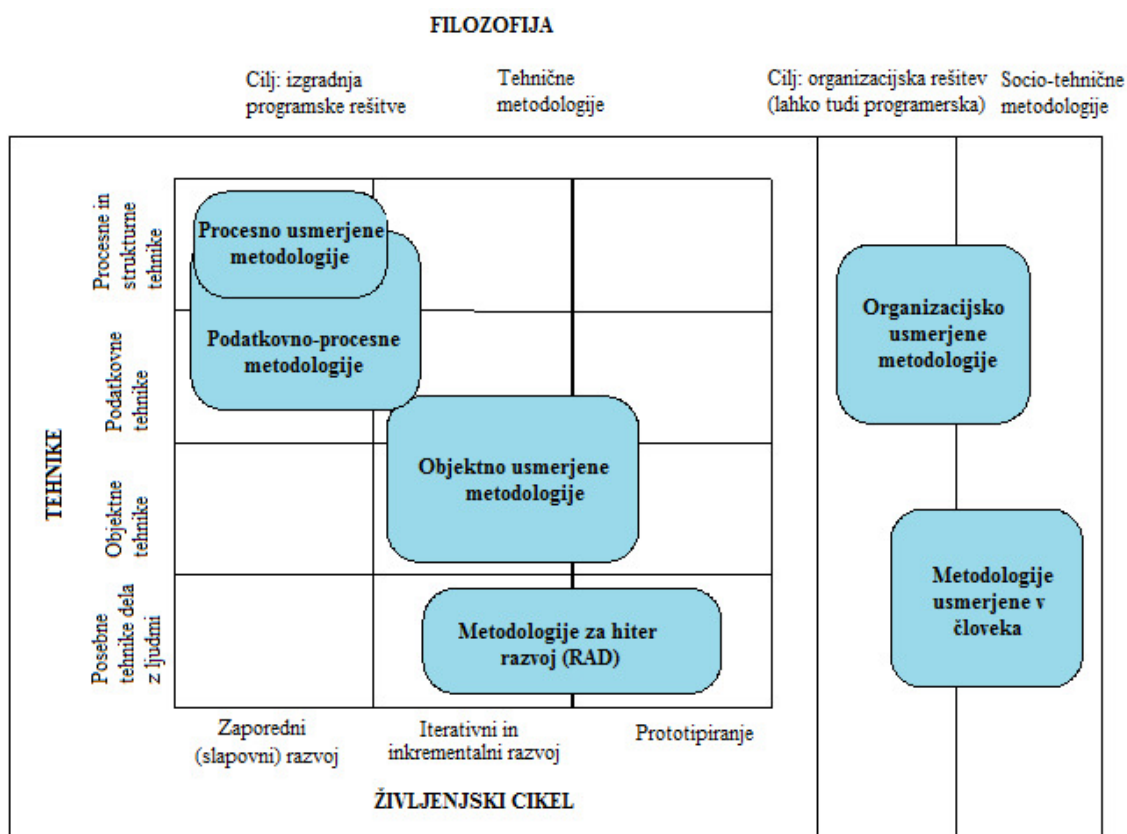
- **Tradicionalne metodologije** za razvoj informacijskih sistemov uporabljajo vnaprej definirane faze razvoja, zato se imenujejo tudi zaporedne. Proces razvoja pri teh metodologijah je linearen, vsaka nadaljnja faza razvoja se lahko začne šele po zaključku predhodne faze.
- **Prototipni pristop** uporablja metode in tehnike modeliranja, znane že iz tradicionalnih metodologij, z namenom, da se že v začetku razvije delujoč vzorec rešitve, ki se ga nato lahko zavrže ali pa izpopolni do končne rešitve. Razvijalci s prototipom predstavijo prihodnji informacijski sistem, ki ga nato s prihodnjimi uporabniki dopolnjujejo in spreminjajo.
- **Objektni pristop** modelov razvoja informacijskih sistemov ne ločuje več na podatkovne in procesne, ampak obravnava oba vidika skupaj. Primera objektnega pristopa sta UML (angl. *Unified Modeling Language*) in TAD (angl. *Tabular Application Development*).

Krisper et al. (2003) ločujejo delitev metodologij razvoja informacijskih sistemov v tri skupine, in sicer glede na tip metodologije, težo metodologije ter utežitev metodologije.

1.1.1 Delitev metodologij glede na njihov tip

Pri delitvi metodologije razvoja programske opreme glede na njihov tip gre za delitev na podlagi filozofije metodologije, kjer so upoštevani sociološka komponenta in cilji metodologije. Metodologije so tako razdeljene na dve glavni skupini, in sicer na socio-tehnične, katerih cilj je tudi organizacijska rešitev, ter na tehnične metodologije, ki se na podlagi življenjskih ciklov in tehnik še dodatno delijo, kar je prikazano na Sliki 1 (Krisper et al., 2003).

Slika 1: Delitev metodologije glede na tip



Vir: M. Krisper, R. Rupnik, M. Bajec, A. Rožanec, A. Zrnc & D. Vavpotič, *Enotna metodologija razvoja informacijskih sistemov*, 2003, 1. zvezek.

Metodologije razvoja informacijskih sistemov se glede na njihov tip delijo na naslednji dve skupini (Krisper et al., 2003):

- **socio-tehnične metodologije**, ki obsegajo metodologije, usmerjene v človeka in organizacijsko usmerjene metodologije ter

- **tehnične metodologije**, ki zajemajo procesno usmerjene, podatkovno-procesno usmerjene metodologije, objektno usmerjene metodologije in metodologije za hiter razvoj.

Za **socio-tehnične metodologije** je značilno, da se ne osredotočajo samo na tehnično plat razvoja sistema, ampak kot pomembno komponento pri razvoju upoštevajo tudi ljudi in njihove lastnosti. Cilj takšnih metodologij je zato poleg programske rešitve lahko tudi organizacijska rešitev. Metodologije, pri katerih je glavni poudarek na organizacijski rešitvi, uvrščamo med organizacijsko usmerjene, metodologije, pri katerih je poudarek na sociološki komponenti, pa med metodologije, usmerjene v človeka (Krisper et al., 2003):

- **Organizacijsko usmerjene metodologije** imajo za cilj modeliranje organizacije kot celote, kar pomeni, da je teoretično njihov cilj lahko tudi samo organizacijska rešitev, čeprav je ta največkrat razvita poleg novo razvite programske rešitve. Zaradi upoštevanja sociološke komponente se te metodologije pogosto prepletajo z metodologijami, usmerjenimi v človeka, hkrati pa tudi z nekaterimi tehničnimi metodologijami (objektno usmerjene metodologije, podatkovno in procesno usmerjene metodologije), ker uporabljajo njim podobne tehnike.
- **Metodologije, usmerjene v človeka**, pri razvoju programskih rešitev uporabljajo socio-tehnični pristop, saj poleg tehnične upoštevajo tudi tehnološko komponento. Metodologije so nastale kot kritika strogo tehničnih metodologij, ki zanemarjajo sociološki vidik. Značilnost teh metodologij je, da v proces odločanja o načrtu in delovanju sistema vključujejo vse ljudi, na katere bo končna programska rešitev ali informacijski sistem vplival. Na eni strani tako vključujejo neposredne uporabnike, ki bodo sistem uporabljali, na drugi strani pa posredne uporabnike, ki se bodo s sistemom srečevali (vodstvo, dobavitelji, stranke). Zaradi širokega kroga ljudi, vpetih v odločanje, te metodologije modelirajo tudi organizacijo kot celoto, zato pogosto vključujejo tudi lastnosti organizacijsko usmerjenih metodologij.

Tehnologije, pri katerih je glavni cilj izgradnja programske rešitve, se uvrščajo med **tehnične metodologije**. Na podlagi uporabljenih tehnik in življenjskega cikla se te metodologije delijo na (Krisper et al., 2003):

- **Procesno usmerjene metodologije:** temeljijo na modeliranju dogajanja in procesov v sistemu, ob tem pa uporabljajo procesne tehnike, kot so odločitvena drevesa, diagrami podatkovnih tokov in akcijski diagrami. Poleg tega uporabljajo tudi strukturne diagrame, ki omogočajo prikaz razgradnje in strukture sistema. Te metodologije največkrat uporabljajo zaporedni (slapovni) življenjski cikel z modifikacijami.
- **Podatkovno-procesno usmerjene metodologije:** nadgrajujejo procesno usmerjene, ker poleg modeliranja procesov sistema vključujejo tudi modeliranje podatkov, ki v

sistemu nastopajo. Tehnike, ki jih uporabljajo te metodologije, vključujejo procesne in strukturne tehnike, kakor tudi tehnike podatkovnega modeliranja. Tako kot procesne tudi podatkovno-procesne metodologije uporabljajo zaporedni (slapovni) življenjski cikel z modifikacijami.

- **Objektno usmerjene metodologije:** zelo se razlikujejo od procesno in podatkovno-procesno usmerjenih metodologij, saj ima pri njih osrednjo vlogo modeliranje objektov, ki hkrati obsegajo podatke in procese ter so osnovni gradniki informacijskega sistema. Usmerjenost na modeliranje objektov zahteva drugačen pristop k analizi in načrtovanju ter uporabo objektno usmerjenih programskih jezikov in orodij. Te metodologije uvajajo poenotenje razvoja programske opreme, ker se s skupnimi koncepti srečujejo skozi celoten proces razvoja, hkrati pa so neločljivo povezane z inkrementalno-iterativnim razvojem. Skupni koncepti omogočajo možnost ponovne uporabe programske kode, modelov in znanja, pridobljenega pri analizi ter načrtovanju. Objektno usmerjene metodologije pri razvoju informacijskega sistema uporabljajo iterativen in inkrementalen življenjski cikel, sistem gradijo v več iteracijah ter na osnovi preizkušenih gradnikov.
- **Metodologije za hiter razvoj** (angl. *Rapid Application Development*, v nadaljevanju RAD): nastale so kot posledica tehnoloških sprememb, ki hitro spreminjajo zahteve informacijskih sistemov. Metodologije temeljijo na iterativnem življenjskem ciklu in prototipiranju ter omogočajo hiter razvoj programske opreme. To jim omogočata uporaba orodij za hiter razvoj programske opreme in poudarek na implementaciji ter testiranju.

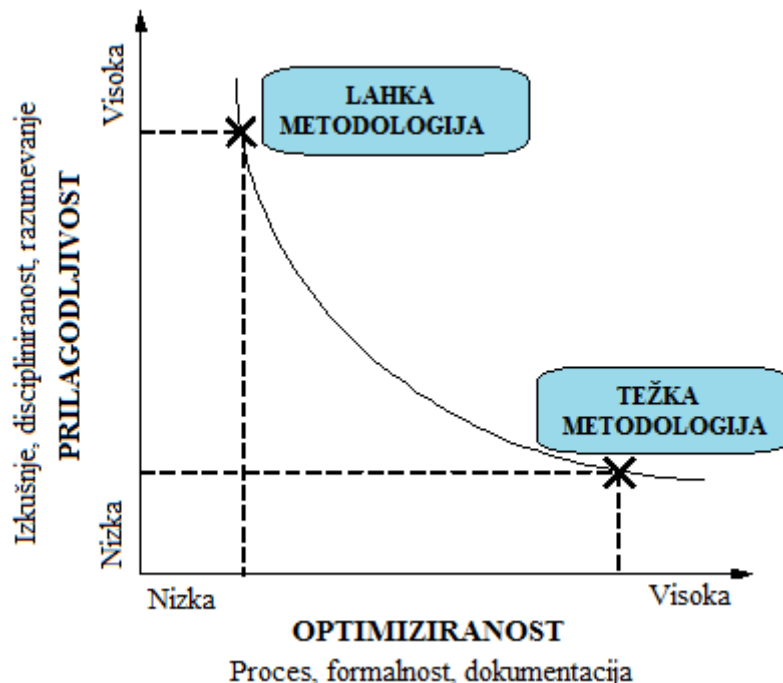
1.1.2 Delitev metodologij glede na njihovo težo

Z uveljavljanjem novih pristopov k razvoju programske opreme, nastalih kot kritika obsežnim formalnim metodologijam, se je pojavila tudi nova delitev metodologij, in sicer na težke ter lahke metodologije. Teža metodologije je določena kot zmnožek njene gostote in obsega, kar je prikazano na Sliki 2. Z gostoto metodologije definiramo zahtevani nivo podrobnosti oziroma formalnosti, iz česar izhaja, da so metodologije z višjo gostoto bolj formalne, njihovi elementi pa so podrobneje definirani kot pri metodologijah z nižjo gostoto, kjer so odločitve prepuščene interpretaciji posameznika. Drugi element za definiranje teže metodologije je njen obseg, in sicer je ta določen glede na število različnih elementov (aktivnosti, vloge, izdelki, standardi), ki jih metodologija opisuje (Krisper et al., 2003).

Tudi Highsmith (2002, str. 202–203) ugotavlja, da je pri delitvi glede na težo metodologije treba upoštevati in primerjati tri dejavnike, in sicer postopek v primerjavi z izkušnjami, formalnost v primerjavi z disciplino ter dokumentacijo v primerjavi z razumevanjem. Pri določanju teže metodologije se primerjata skupek dejavnikov in njihov vpliv na

metodologijo, kar se odraža kot primerjava optimiziranosti metodologije z njeno prilagodljivostjo.

Slika 2: Določitev teže metodologije



Vir: J. Highsmith, *Agile Software Development Ecosystem*, 2002, str. 204.

Višja kot sta obseg in gostota, težja je metodologija. Za **težke metodologije** so značilni visoka stopnja formaliziranosti in dobro optimizirani procesi, ki so podprti z natančno pripravljeno dokumentacijo ter predpisanimi orodji in tehnikami. Zaradi podrobne dokumentacije je takšna metodologija visoko optimizirana in primerna za izgradnjo kritičnih programskih rešitev, pri katerih sodeluje večja razvojna skupina (Krisper et al., 2003). Highsmith (2002, str. 203) dodatno ugotavlja, da so pri večjih in obsežnejših projektih potrebne dodatne formalnosti ter obsežnejša dokumentacija in hkrati poudarja, da je za pripravo slednjih na začetku projekta potrebna velika količina znanja ter izkušenj.

Težke metodologije uporabimo v primeru, ko (Krisper et al., 2003):

- je s strani naročnika pri izdelavi dokumentacije zahtevana visoka stopnja formalizma;
- so zahteve dobro definirane in stabilne ter se ne bodo spreminjale;
- pri izgradnji sodelujejo manj izkušeni razvijalci, vendar lahko pomanjkanje njihovega znanja in izkušenj nadomestimo s formalno opredeljenimi pravili;
- je poleg razvoja programske rešitve dodaten cilj tudi prenova organizacijskega sistema;
- razvijamo visoko kritičen sistem, ki zahteva izdelavo podrobnih načrtov in dokumentacije.

Krisper et al. (2003) ugotavljajo, da so metodologije z manjšo gostoto in manjšim obsegom **lahke metodologije**, ki so primerne za okolja, kjer ključni cilj predstavlja razvoj programske rešitve, pri čemer zapostavljajo organizacijske, poslovne ter sociološke vidike sistema. Lahke metodologije temeljijo na dobrem znanju in izkušnjah razvijalcev ter sodelovanju in disciplini med vsemi udeleženci v projektu, hkrati pa so visoko prilagodljive, zato se lahko sproti odzivajo na morebitne spremembe zahtev, ki se pojavijo v postopku razvoja rešitve. Lahke metodologije, v nasprotju s težkimi, niso obremenjene z visoko stopnjo formalnosti in so kot takšne primerne predvsem za manjše razvojne skupine.

Lahke metodologije so uporabne v naslednjih primerih (Krisper et al., 2003):

- zahteve za programske rešitve so nepredvidljive in se spreminjajo;
- razvijalci, ki sodelujejo pri projektu, so izkušeni, odgovorni, disciplinirani ter seznanjeni s tehnikami dela;
- razvoj programske rešitve je glavni in dokončno opredeljeni cilj;
- naročnik razume bistvo lahke metodologije in je vključen v razvoj programske rešitve;
- cilj razvoja je nekritičen sistem, ki ga je mogoče razviti z majhno razvojno ekipo.

1.1.3 Delitev metodologij glede na njihovo utežitev

Zadnja delitev metodologije deli glede na utežitev. Ideja za to delitev izhaja iz dejstva, da je razvoj programske opreme sestavljen iz različnih aktivnosti in zaporedja izvajanja le teh. Pred dejanskim razvojem programske opreme je treba pripraviti analizo in načrt potrebnih sprememb, ki mu sledita kodiranje ter testiranje. Glede na to, katere aktivnosti metodologija poudarja, lahko izvedemo delitev na spredaj utežene metodologije, ki poudarjajo prvi dve aktivnosti, ter na zadaj utežene metodologije, pri katerih je poudarek na zadnjih dveh aktivnostih. Če metodologija pri razvoju nekaterih delov informacijskega sistema poudarja prvi aktivnosti, pri drugih delih pa zadnji, lahko govorimo o uravnoteženi metodologiji. Spredaj utežene metodologije so se v praksi izkazale kot težke metodologije, medtem ko se zadaj utežene metodologije izkazujejo kot lahke (Krisper et al., 2003).

Spredaj utežena metodologija poudarja aktivnosti analize in načrtovanja, kar se odraža v dejstvu, da takšna metodologija omogoča zelo podrobno izdelane systemske zahteve in analize ter zelo podrobno definirane načrte izgradnje. Metodologija sicer dopušča kasnejše spreminjanje in dopolnjevanje načrtov izvedbe, vendar se končni načrt bistveno ne razlikuje od prvotno zastavljenega. Podrobno izdelane analize in načrti izgradnje omogočajo, da se pri izdelavi informacijskega sistema opravlja le rutinsko kodiranje v načrtu opredeljenih metod, hkrati pa je pri teh metodah zahtevanega tudi manj končnega testiranja, ker dobro definirani načrti izničujejo možnosti za napake. Klasične metodologije poudarjajo pomembnost analize in načrta, kar pomeni, da jih lahko večino uvrstimo v

skupino spredaj uteženih metodologij. Glavni poudarek spredaj uteženih metodologij pa je tudi njihova slabost, saj podrobne analize in načrti pogosto vodijo v preveliko količino časa, končni rezultat pa je zaradi hitrega spreminjanja zahtev lahko celo neuporaben (Krisper et al., 2003).

Krisper et al. (2003) ugotavljajo, da so spredaj utežene metodologije primerne za razvoj:

- sistemov s stabilnimi zahtevami, ki so dobro specificirane. Ker je poudarek na analizi in načrtovanju, so lahko poznejše spremembe načrta zelo drage, hkrati pa lahko porušijo arhitekturo prvotno načrtovanega sistema;
- kompleksnih sistemov, ki ga razvija velika razvojna skupina, v kateri sta potrebna delitev dela in sodelovanje med člani, kar zahteva podrobno izdelane načrte izvedbe;
- kritičnih sistemov, pri katerih sta analiza in načrtovanje ključnega pomena, saj se z njima predvidi tudi alternativne scenarije, odstopanja in napake, ki se jih vključi v načrt izgradnje.

V nasprotju s spredaj uteženimi metodologijami pa **zadaj utežene metodologije** poudarjajo pomembnost postopkov izvedbe in testiranja. Analiza in načrtovanje služita le za izdelavo osnovnega načrta izgradnje, brez podrobnosti. Te razvijalci definirajo šele v fazi dejanske izgradnje, ki ji sledi obsežno testiranje in popravljanje kode, katere rezultat je lahko relativno stabilen izdelek, ki pa ima slabšo arhitekturo ter je težje razširljiv. V to skupino metodologij uvrščamo metodologije prototipiranja, katerih glavna kritika je, da niso primerne za razvoj obsežnih sistemov (Krisper et al., 2003).

Zadaj utežene metodologije so primerne za razvoj (Krisper et al., 2003):

- sistemov, ki se bodo najverjetneje še spreminjali in imajo v izhodišču slabo definirane zahteve, ker jih v fazi načrtovanja še ni mogoče predvideti;
- manjših in enostavnejših sistemov, ki jih razvija majhna, a izkušena razvojna skupina. Takšna skupina lahko s svojim znanjem in izkušnjami nadomesti slabo definiran načrt izgradnje;
- sistemov, temelječih na tehnologiji, ki je razvojnikom nepoznana. Zadaj utežena metodologija omogoča, da razvojniki tehnologijo najprej preizkusijo in spoznajo;
- sistemov, ki niso kritični in pri izgradnji omogočajo manj podrobno analizo ter načrtovanje, kar privede do slabe arhitekture sistema, ki ga je težje vzdrževati in nadgrajevati.

Uravnotežene metodologije omogočajo, da na eni strani za dele sistema z dobro znanimi in stabilnimi zahtevami izvedemo podrobno analizo ter načrt in na drugi strani za dele sistema, ki so nam nepoznani ali pa zahteve niso jasno opredeljene ter stabilne, takoj pričnemo s kodiranjem in testiranjem prototipne rešitve. Pri izgradnji tako kombiniramo

oba skrajna pristopa, kar pa ni nujno idealna odločitev, saj obstajajo procesi, ki nimajo vseh značilnosti spredaj ali vseh značilnosti zadaj uteženih procesov (Krisper et al., 2003).

Različni avtorji navajajo različne delitve metodologij, ki se uporabljajo pri razvoju informacijskih sistemov, vendar jih lahko večino razdelimo na dve veliki skupini, in sicer na tradicionalne (težke) in agilne (gibke, lahke). Pri razvoju informacijskih rešitev v zadnjem času vodilno vlogo prevzemajo sodobne agilne metode, ki pa niso primerne za vse oblike razvoja sistemov, zato tradicionalnih metodologij ne bodo nikoli popolnoma nadomestile.

1.2 Tradicionalne metodologije

Proces izgradnje in vzdrževanja informacijskega sistema temelji na sosledju določenih faz oziroma sledi življenjskemu ciklu razvoja sistema. Zanj je značilno, da poleg predhodne analize vključuje različne faze, ki sledijo izgradnji in testiranju (Yu et al., 2012, str. 162). Bajec (b.l., str. 17) ugotavlja, da življenjski cikel razvoja informacijskega sistema določa sosledje faz oziroma pove, na kakšen način si v okviru razvoja informacijskega sistema sledijo posamezne faze.

Proces razvoja informacijskega sistema in vsake njegove generacije, pri čemer se kot generacija šteje nova različica programske opreme, gre skozi posamezne razvojne faze oziroma sledi življenjskemu ciklu razvoja sistema. Vsaka faza se zaključi z definiranim časovnim mejnikom in zahteva sprejem novih odločitev. Faze življenjskega cikla razvoja informacijskega sistema so naslednje (Krisper et al., 2003):

- **Začetna faza** definira projekt s poslovnega stališča in določi njegov obseg. V tej fazi se poišče vse zunanje entitete, s katerimi bo sistem sodeloval, in določi kriterije uspeha oziroma neuspeha projekta, oceni se tveganja ter izdelava fazni načrt, ki vključuje glavne časovne mejnike projekta.
- **Faza zbiranja informacij** je namenjena analizi problemskega področja, izdelavi projektnega plana in razrešitvi morebitnih tveganjih elementov ter izdelavi načrta izgradnje.
- **Faza konstrukcije** obsega izdelavo celotnega proizvoda, definiranega v analizi in načrtu izgradnje. Ta faza vključuje tudi testiranje in pripravo dokumentacije ter uporabniških navodil.
- **Faza prevzema** predstavlja predajo naročniku oziroma implementacijo novega sistema ali programske opreme v obstoječi sistem. V tej fazi se lahko pojavi potreba po dodatnih dopolnitvah ali popravkih.

Bajec (b.l., str. 17) postavlja podobno definicijo in ugotavlja, da tako kot večina razvojnih procesov tudi razvoj informacijskega sistema sledi določenemu življenjskemu ciklu

oziroma razvojnemu modelu, ki določa zaporedje faz razvoja, ki jih definira kot analizo, načrtovanje, implementacijo, testiranje, uvedbo in vzdrževanje.

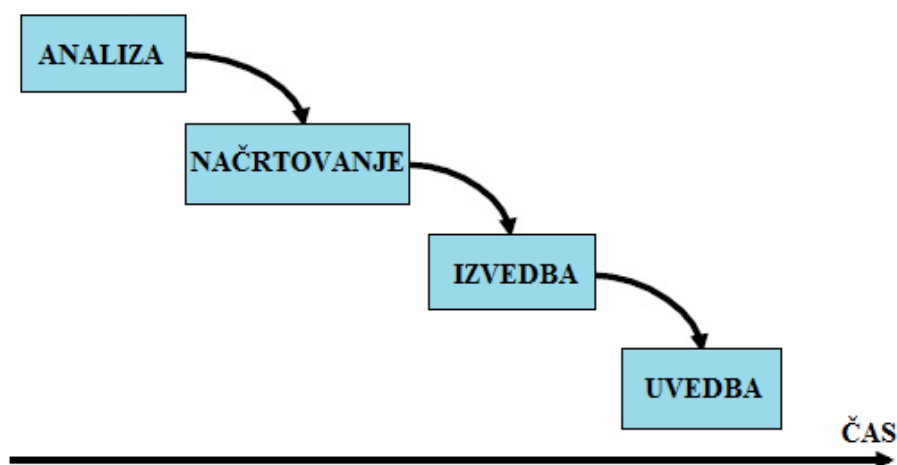
Tradicionalne metode razvoja informacijskih sistemov, temelječe na življenjskem ciklu razvoja, se uvrščajo med težke metodologije, za katere je značilen zaporeden niz korakov, ki največkrat predstavljajo analizo rešitve, njeno izgradnjo, testiranje in implementacijo. Tradicionalne metodologije že na začetku življenjskega cikla, ki predstavlja analizo rešitve, zahtevajo pripravo podrobne dokumentacije in specifikacij, ki se jim sledi skozi celoten življenjski cikel (Yu et al., 2012, str. 162).

Pri tradicionalnem planiranem načinu razvoja, ki temelji na življenjskem ciklu razvoja, so se razvili modeli, prikazani v nadaljevanju. V praksi se največkrat uporablja kombinacijo različnih modelov oziroma izpeljanke zaporednega modela, ki temelji na sosledju posameznih aktivnosti.

1.2.1 Zaporedni ali slapovni model

Pri zaporednem ali slapovnem modelu (angl. *Waterfall Model*), predstavljenem na Sliki 3, gre za model, pri katerem si posamezne razvojne faze informacijskega sistema sledijo po vnaprej določenem zaporedju, pri čemer prekrivanja med posameznimi fazami naj ne bi bilo, ker se naslednja faza začne šele po tem, ko je predhodna faza v celoti končana (Hvala, 2011, str. 30).

Slika 3: Zaporedni ali slapovni model



Vir: R. Rupnik, *Osnove informacijskih sistemov*, b.l., str. 76.

Gre za najstarejši razvojni model, značilen za prve oblike strukturiranega pristopa, pri katerem si faze analize, načrtovanja, izvedbe in uvedbe sledijo zaporedno ter vračanje nazaj ni mogoče. Slapovni model je primeren za obsežne in kompleksne projekte, pri katerih so zahteve že na začetku dobro definirane in se med samim projektom ne bodo

bistveno spreminjale. Zaradi obsežne spremljajoče se dokumentacije ta model omogoča natančne vodenje in pregled nad projektom (Bajec, b.l., str. 22).

Gradišar et al. (2005, str. 270–272) metodologijo življenjskega cikla razvoja informacijskega sistema ponazarjajo s slapovnim modelom, ki je sestavljen iz naslednjih faz oziroma korakov:

- **Študija izvedljivosti** predstavlja prvi korak razvoja informacijskega sistema, s katerim se opredeli stroške izvedbe in koristi nove rešitve, razišče se s strani naročnika sporočene zahteve ter morebitne probleme, za katere se pripravi možne alternativne rešitve.
- **Raziskava sistema** služi pripravi specifikacije in analize zahtev uporabnikov, v kateri se evidentira tudi omejitve ter probleme obstoječega sistema.
- **Analiza** predstavlja tretjo fazo razvoja, v kateri se izdela logični model podatkov in procesov sistema.
- **Načrtovanje** v predhodni fazi pripravljen logični model spremeni v načrt prihodnjega informacijskega sistema, pri tem pa definira vhodne in izhodne podatke sistema, njihovo strukturo ter način zajema podatkov.
- **Implementacija** je predzadnja faza, ki je namenjena izgradnji sistema, ki ji na koncu sledi prehod iz starega v novi sistem ter izobraževanje osebja, ki bo uporabljalo novi sistem.
- **Vzdrževanje sistema** je zadnja faza v metodologiji razvoja, ki temelji na življenjskem ciklu, in sicer se ta izvaja po tem, ko je bil prehod na novi sistem že opravljen. Ta faza zagotavlja učinkovito delovanje novega sistema, to pa zagotavlja s sprotnim odpravljanjem odkritih napak in morebitnimi prilagoditvami sistema.

Bajec (b.l., str. 23–24) ugotavlja, da se **prednost slapovnega modela** kaže v manjši količini porabljenega režijskega dela, ki ni neposredno povezano z izdelavo programske opreme (vodenje projekta), ker je mogoče načrtovanje v celoti izvesti vnaprej. Hvala (2011, str. 30) dodaja, da visoka stopnja formalizacije tega modela skoraj vedno zagotavlja visoko kakovost končnega izdelka.

Slabosti slapovnega modela se kažejo v njegovi nefleksibilnosti, saj vsaka naknadna sprememba zahtev terja veliko dodatnega napora. Informacijsko okolje se hitro spreminja, zato zahteve nikoli niso statične. Naslednjo slabost predstavlja dejstvo, da je pred začetkom naslednje faze treba v celoti zaključiti predhodno fazo, saj vračanje nazaj ni več mogoče, ker je v praksi skoraj neizvedljivo. Ker vračanje k predhodnim fazam vse do konca projekta ni mogoče, obstaja veliko tveganje, da razviti sistem ne bo ustrezal dejanskim zahtevam (Rupnik, b.l., str. 77).

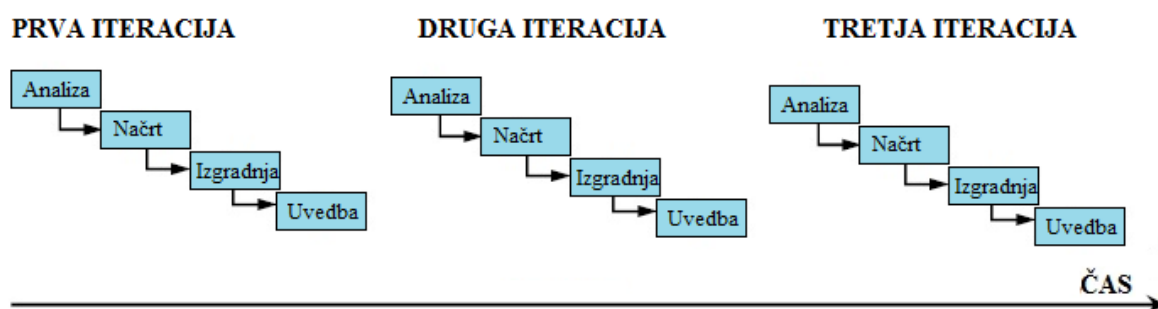
Podobno ugotavlja tudi Hvala (2011, str. 30), ki kot slabost slapovnega modela navaja nezmožnost prilagajanja razmeram na trgu in ogromno količino dokumentacije, ki spremlja vsako posamezno fazo.

1.2.2 Iterativni model

Iterativni modeli so bili razviti kot odziv na pomanjkljivosti slapovnega modela, ki onemogoča ponavljanje posameznih faz in vračanje na predhodne faze. Njihova skupna značilnost je postopen razvoj, kar pomeni, da posameznih faz ne končujejo v celoti, ampak le delno, pri tem pa se celoten cikel ponavlja toliko časa, dokler projekt ni končan. Med najbolj poznane iterativne modele se uvršča **spiralni model** (angl. *Spiral Model*) (Krisper et al., 2003). Awad (2005, str. 6) meni, da spiralni model združuje tako elemente načrtovanja kot tudi elemente prototipiranja in iteracij, ki jih ponavljamo toliko časa, dokler projekt ni končan.

Bajec (b.l., str. 25–28) ugotavlja, da iterativni model razvoja poteka v več iteracijah oziroma zaporedju aktivnosti, ki so izvedene na osnovi načrta in z določenimi kriteriji vrednotenja, ki se končajo z realizacijo izdelka. V vsaki iteraciji, ki gre navadno čez vse faze življenjskega cikla razvoja informacijskega sistema (analiza, načrtovanje, izgradnja, uvedba), se navadno razvije določen del funkcionalnosti celotnega sistema, pri čemer običajno najbolj tvegane dele sistema razvijemo v začetnih iteracijah. Proces razvoja z iterativnim modelom je prikazan na Sliki 4. Posamezna iteracija običajno traja od sedem do štirinajst dni, med njenim izvajanjem pa se ne sprejema sprememb zahtev. Po uspešnem zaključku prve iteracije se na osnovi njenih rezultatov določi vsebino naslednje iteracije in začne z njeno izvedbo.

Slika 4: Iterativni model



Vir: R. Rupnik, *Osnove informacijskih sistemov*, b.l., str. 80.

Prednosti iterativnega modela v primerjavi s slapovnim se kažejo v tem, da so lahko najbolj tvegani deli projekta razrešeni še preden postane investicija prevelika, poleg tega pa iteracije omogočajo predajo izvedenega dela projekta, še preden je dokončan celoten projekt. S tem se zagotovi povratne informacije s strani uporabnikov, ki jih lahko

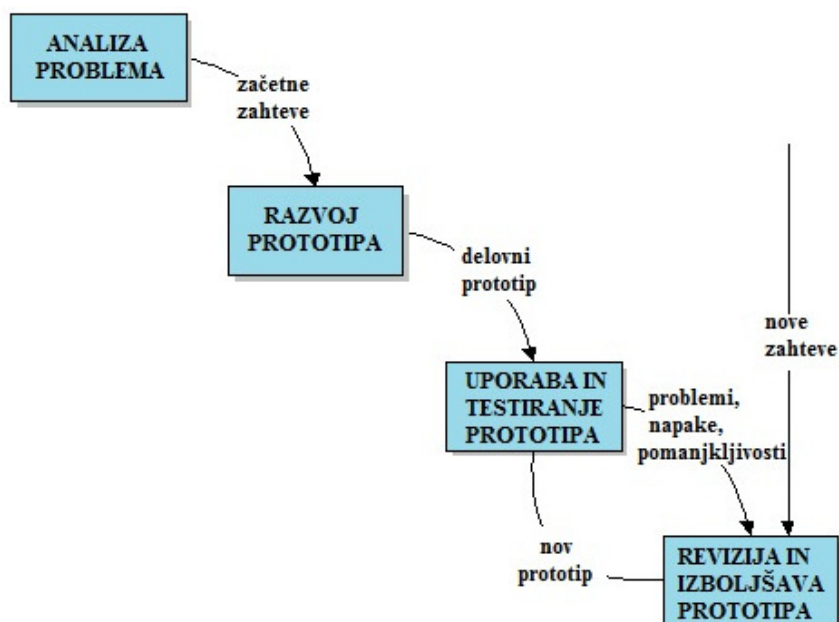
upoštevamo pri načrtovanju naslednjih iteracij. Posamezne iteracije merimo z ocenjevanjem izvedenega dela in predstavljajo ciljne mejnike, kar nam omogoča, da se osredotočamo samo na eno iteracijo in ne na celoten projekt razvoja (Rupnik, b.l., str. 82).

Slabost iterativnega modela se kaže v težkem in nepredvidljivem načrtovanju projekta, saj vnaprej ni mogoče točno predvideti, koliko posameznih iteracij bo potrebnih za razvoj dokončnega produkta. Posledica te ovire je izredno zahtevno vodenje projekta (Bajec, b.l., str. 31).

1.2.3 Prototipni model

Pri prototipnih modelih gre za eno izmed različic iterativnega modela, ki temelji na izdelavi prototipov, ki so predhodno izdelane in navadno še nepopolne različice sistema ali dela sistema (Bajec, b.l., str. 32). Rupnik (b.l., str. 84) ugotavlja, da se prototipi lahko uporabljajo na dva načina. Pri prvem načinu se uporabljajo kot del specifikacije sistema, s katero se pridobi jasnejšo podobo prihodnjega sistema in se v nadaljevanju zavržejo. Pri drugem načinu uporabe pa prototipi predstavljajo osnovo za izdelavo produkcijskega sistema, kar je prikazano na Sliki 5.

Slika 5: Prototipni model – RAD



Vir: M. Krisper, R. Rupnik, M. Bajec, A. Rožanec, A. Zrnc & D. Vavpotič, Enotna metodologija razvoja informacijskih sistemov, 2003, 1. zvezek.

Prototipi v veliki meri olajšajo komuniciranje z uporabnikom, zato se jih uporablja v različnih fazah razvoja. Analitik z naročnikom najprej uskladi zahteve sistema, nato pa izdelava prototip ter ga da uporabniku v testiranje. Uporabnik prek povratnih informacij in

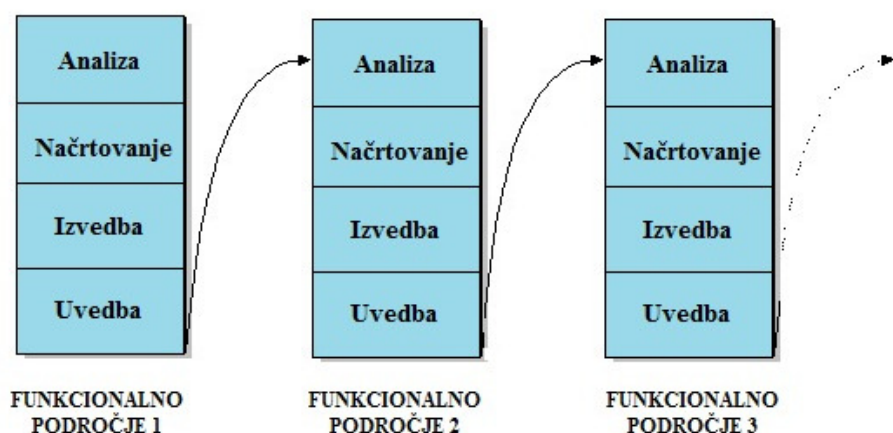
priporočam analitiku sporoči, kaj mu je pri prototipu všeč in kaj ne, ta pa nato na podlagi prejetih informacij izdela novi prototip ter postopek ponavlja, dokler naročnik ni zadovoljen s končnim rezultatom (Krisper et al., 2003).

1.2.4 Inkrementalni model

Inkrementalni model temelji na postopni gradnji informacijske rešitve, pri katerem te ne razvijamo v celoti, temveč se omejimo na posamezen sklop, ki ga razvijemo v celoti in ob predaji uporabniku povežemo z ostalimi sklopi (Bajec, b.l., str. 34–35).

Krisper et al. (2003) ugotavljajo, da gre za pogosto uporabljeni model razvoja informacijskega sistema, pri katerem se celoten problem razdeli na podprobleme ali module, ki se rešujejo posebej in neodvisno od ostalih. Tako razviti deli aplikacij, ki zajemajo le del funkcionalnosti celotnega sistema, so dovolj samostojni, da se lahko vključijo v produkcijo. Celoten projekt se lahko razdeli na funkcionalna področja, kar je razvidno na Sliki 6. Ta področja predstavljajo posamezne inkremente, ki se jih na koncu združi v celotno rešitev.

Slika 6: Inkrementalni model



Vir: M. Krisper, R. Rupnik, M. Bajec, A. Rožanec, A. Zrnc & D. Vavpotič, Enotna metodologija razvoja informacijskih sistemov, 2003, 1. zvezek.

Prednost inkrementalnega modela se kaže v dejstvu, da naročnik lažje sledi napredku celotnega projekta, saj informacijsko rešitev prejema postopoma, sam pa lahko sodeluje pri testiranju že razvitih sklopov (Bajec, b.l., str. 39). Krisper et al. (2003) kot prednost navajajo tudi povečano možnost za odkrivanje morebitnih napak in lažje odpravljanje le teh, saj se del končnega produkta že relativno zgodaj preda v redno uporabo. Ugotavljajo tudi, da je pri tem modelu razvoj informacijskega sistema v splošnem cenejši in manj tvegan od razvoja celotnega sistema v enem kosu.

Slabosti inkrementalnega modela so naslednje: ni uporaben pri vseh projektih, saj nekaterih rešitev ni moč predati v uporabo po delih. Pri načrtovanju posameznih sklopov informacijskega sistema se lahko zgodi, da sklope narobe razporedimo, kar neugodno vpliva na izdelavo rešitve (Bajec, b.l., str. 39).

1.3 Agilne metodologije

Podjetja, ki se ukvarjajo z razvojem informacijskih sistemov in programskih rešitev, delujejo v dinamičnem ter konkurenčnem okolju, v katerem so tehnološke novosti stalnica. Poleg sprememb v tehnologijah pa se spreminjajo tudi zahteve in pričakovanja kupcev glede pričakovane življenjske dobe produktov. Za to, da lahko podjetja uspešno konkurirajo na takšnem trgu, morajo biti sposobna obvladovati neenakomerno in nepredvidljivo povpraševanje, kar pa jim uporaba tradicionalnih razvojnih metodologij ne omogoča. Kot odgovor na težave tradicionalnih metodologij in njihovo nezmožnost prilagajanja so se začele razvijati agilne metodologije (Kettunen, 2009, str. 408).

Informacijska tehnologija, ki se naglo razvija in pri razvoju omogoča nove možnosti ter omejitve tradicionalnih metodologij razvoja informacijskih sistemov, je privedla do razvoja alternativnih metod, kot so prototipni razvoj, hiter razvoj aplikacij, skupen razvoj aplikacij, ekstremno programiranje, evolutivni razvoj, inkrementalni razvoj, objektni pristop in agilni razvoj aplikacij. Skupna lastnost alternativnih metodologij je želja po odpravi pomanjkljivosti tradicionalnih metodologij in težnja po hitrejšem razvoju, v katerega je vključen tudi končni uporabnik (Gradišar, Jaklič & Turk, 2007, str. 172).

Agilni razvoj ni postopek, ki bi mu lahko sledili, temveč predstavlja filozofijo oziroma način razmišljanja, ki ga uporabljamo pri razvoju informacijskih sistemov. Agilne metode so sestavljene iz posameznih elementov oziroma priporočil, ki so že dolgo poznana, vendar so pri agilnih metodah združena v unikatno kombinacijo (Shore & Warden, 2008, str. 9).

Ključna prednost agilnih metodologij je njihova preprostost in hitrost, saj se razvojna skupina osredotoča le na izgradnjo nujnih funkcionalnosti, potrebnih za hitro dostavo delujočega proizvoda. Metodologijo lahko označimo za agilno, ko (Pekka, Outi, Jussi & Juhani, 2002, str. 17):

- pri razvoju uporablja inkrementalni pristop z dostavo več manjših različic in več ponovitvami;
- temelji na tesnem sodelovanju med razvojno ekipo in naročnikom;
- je metoda lahka za uporabo in prilagodljiva različnim razvojnim okoljem;
- je metoda visoko prilagodljiva pri upoštevanju spremenjenih zahtev.

Skupina sedemnajstih strokovnjakov, tako ali drugače znanih s področja lahkih metodologij, se je decembra 2001 sestala z namenom, da bi ugotovili skupne točke

posameznih metodologij ter na njihovi podlagi postavili skupne metodološke osnove lahkih oziroma agilnih metodologij. Strokovnjaki so se združili v zvezo agilnih (angl. *Agile Alliance*) in v Manifestu agilnega razvoja programske opreme (angl. *The Agile Manifesto*) objavili trend novih agilnih metodologij, znanih po svoji učinkovitosti in prilagodljivosti (Bajec & Krisper, b.1., str. 4).

V svojem manifestu so člani zveze svoje cilje definirali kot iskanje boljših načinov razvoja programske opreme tako, da jo sami razvijajo in pri tem pomagajo tudi drugim. Iz zastavljenega cilja so izpeljali naslednje vrednote (Manifesto for Agile Software Development, 2016):

- posamezniki in interakcije med njimi so pomembnejše od procesov ter orodij;
- delujoča programska oprema je pomembnejša od natančne dokumentacije;
- sodelovanje s stranko je pomembnejše kot pogajanje na osnovi pogodb;
- odziv na spremembe je pomembnejši od togega sledenja načrtom.

Prvo načelo zveze agilnih – **posamezniki in interakcije med njimi so pomembnejše od procesov in orodij** izhaja iz mnenja, da se je treba individualnim članom projekta, njihovemu znanju in željam posvetiti v večji meri. Strogo določena porazdelitev vlog, določenih s procesom, ni smiselna, če za to nimamo ustreznih ljudi, ki bi te vloge uspešno opravljali. Prvo načelo poudarja tudi pomembnost interakcije in komunikacije med člani projekta, saj sta ti ključni za uspešnost projekta. Na takšen način lahko dosežemo dobre rezultate, kljub temu, da udeleženci ne upoštevajo nobenega predpisanega procesa, kot pa v primeru, ko so procesi natančno predpisani, a med udeleženci ni komunikacije (Bajec & Krisper, b.1., str. 5).

Bajec in Krisper (b.1., str. 5) drugo načelo – **delujoča programska oprema je pomembnejša od podrobne dokumentacije** razlagata kot dejstvo, da je delujoči program največ, kar lahko končni uporabnik dobi in je pomembnejši kot podrobna dokumentacija. Naročnik kljub prejeti popolni dokumentaciji ne bo zadovoljen, če ne bo najprej videl delujočega sistema. Dobra dokumentacija pa po drugi strani olajša vzdrževanje sistema in komunikacijo med naročnikom ter izvajalcem tako, da se je ne sme popolnoma zanemariti. Pomembno je, da se dokumentacijo pripravlja v pravem obsegu oziroma na način, da je vsak del dokumentacije utemeljen.

Tretje načelo zveze agilnih – **sodelovanje s stranko je pomembnejše kot pogajanje na osnovi pogodb** posveča pozornost odnosu med naročnikom in izvajalcem, ki je velikokrat preveč strog ter formalen. Če se naročnik in izvajalec dobro ujameta, je to lahko ključnega pomena za uspešnost projekta, saj naročnik najbolje ve, kaj potrebuje, čeprav tega pogosto ne zna izraziti in razložiti. Tehnološko zahteven projekt lahko močno olajša dobro ujemanje med naročnikom in izvajalcem, na drugi strani pa lahko neujemanje ter strog pogodben odnos oteži že izvedbo enostavnih projektov (Bajec & Krisper, b.1., str. 5–6).

Bajec in Krisper (b.l., str. 6) zadnje načelo – **odziv na spremembe je pomembnejši od togega sledenja načrtom** navajata kot enega od ključnih za uspešnost projekta. Spremembe in nove zahteve mora metodologija razvoja informacijskega sistema obvladovati v celotni fazi projekta. Projektni plani so koristni element, vendar morajo omogočati spremembe vsaj znotraj dogovorjenega okvirja, poleg tega pa je utopično pričakovati, da se bo lahko v začetnih fazah projekta zajelo vse možne spremembe. Planiranje in sledenje načrtom je pomembno in se ga ne sme zapostavljati, vendar načrtom sledimo le do stanja, ko se to ne razlikuje preveč od dejanskega.

Iz osnovnih štirih načel, ki veljajo kot osnova agilnih metodologij, so bila izpeljana priporočila, ki naj bi bila v pomoč pri gradnji in vrednotenju vseh tistih metodologij, ki se deklarirajo kot agilne. Ta priporočila so naslednja (Manifesto for Agile Software Development, 2016):

- Najvišja prioriteta mora biti zadovoljstvo stranke, ki ga dosežemo s pravočasnimi in nepretrganimi različicami programske opreme.
- Spremembe zahtev je treba sprejeti tudi v poznejših fazah razvoja, saj lahko z njihovo realizacijo svoji stranki zagotovimo konkurenčnost.
- Delujočo programsko opremo je treba izdajati pogosto, znotraj obdobja nekaj tednov do nekaj mesecev, pri čemer so zaželeni časovni intervali, ki so krajši.
- Skozi celoten projekt je zaželeno dnevno sodelovanje med naročnikom in razvojno ekipo.
- Projekte se izvaja z motiviranimi posamezniki, ki se jim zagotovi ustrezno delovno okolje in nudi podporo, ki jo potrebujejo, hkrati pa jim je treba tudi zaupati, da bodo dobro opravili svoje delo.
- Osebni stik je najučinkovitejša in najboljša metoda posredovanja informacij razvojni ekipi, pa tudi znotraj ekipe.
- Osnovno merilo napredka projekta je delujoča programska oprema.
- Agilne metodologije promovirajo trajnostni razvoj programske opreme, kar za razvojno ekipo, podpornike in uporabnike rešitev zahteva zagotavljanje konstantnega tempa za nedoločen čas.
- Neprekinjena težnja k dobrim načrtom in tehnični odličnosti povečuje agilnost metodologij.
- Preprostost in enostavnost je bistvenega pomena, saj vodi k zmanjšanju količine nepotrebnega dela.
- Ekipe, ki so se samoorganizirale, lahko pripravijo najboljše arhitekture, zahteve in načrte.
- Ekipe naj v rednih časovnih presledkih iščejo načine, kako postati učinkovitejše, nato pa naj svoje delovanje prilagodijo ugotovitvam.

Hvala (2011, str. 31) ugotavlja, da v skupino metod z agilno metodologijo razvoja informacijskih sistemov spada več metod, ki imajo skupne nekatere lastnosti, kot so usmerjenost v razvoj, tesno sodelovanje in prilagajanje procesov trenutnim razmeram znotraj življenjskega cikla projektov. Metode pri načrtovanju sistema posamezne naloge razbijejo na manjše korake ali inkremente, ki zahtevajo le kratkoročno planiranje.

Med najpogostejšimi agilnimi metodologijami razvoja informacijskih sistemov se največkrat navajajo naslednje (Abrahamsson, Salo & Ronkainen, 2002, str. 18–83):

- ekstremno programiranje – Model XP (angl. *Extreme Programming*);
- model Scrum;
- model RUP (angl. *Rational Unified Process*);
- funkcionalno voden razvoj informacijskih rešitev – model FDD (angl. *Feature Driven Development Programming*);
- dinamični sistemski model razvoja informacijskih rešitev – model DSDM (angl. *Dynamic System Development Method*);
- prilagodljivi razvoj programskih rešitev – model ASD (angl. *Adaptive Software Development*);
- metodologije družine Crystal (angl. *Crystal family of methodologies*).

1.3.1 Ekstremno programiranje – model XP

Metoda ekstremnega programiranja se je razvila kot odgovor na težave, ki jih je prinašal dolg razvojni cikel tradicionalne metode življenjskega cikla. Metoda je znana po kratkih razvojnih ciklih, inkrementalnem planiranju, stalnemu povratnemu toku informacij in dobri komunikaciji (Awad, 2005, str. 8).

Ekstremno programiranje je prilagodljiva, lahka, predvsem pa napovedljiva metodologija razvoja programske opreme, ki prinaša minimalno tveganje. Metodologija prinaša hitre rezultate, tok povratnih informacij je stalen in stvaren, inkrementalni pristop k razvoju pa omogoča hiter razvoj delujoče rešitve, ki jo naročnik prejme v testiranje. Ker gre za agilno metodologijo, se lahko hitro odziva in prilagaja spremenjenim zahtevam naročnika, avtomatizirani testi razvijalcev ter uporabnikov pa omogočajo hitro odkrivanje in popravek napak (Beck, 1999, str. 1–9).

Highsmith (2002, str. 167–170) navaja dvanajst osnovnih principov ekstremnega programiranja, ki mu omogočajo primerjalne prednosti v primerjavi z drugimi metodologijami. Principi so naslednji:

- igra planiranja,
- majhne in pogoste izdaje,

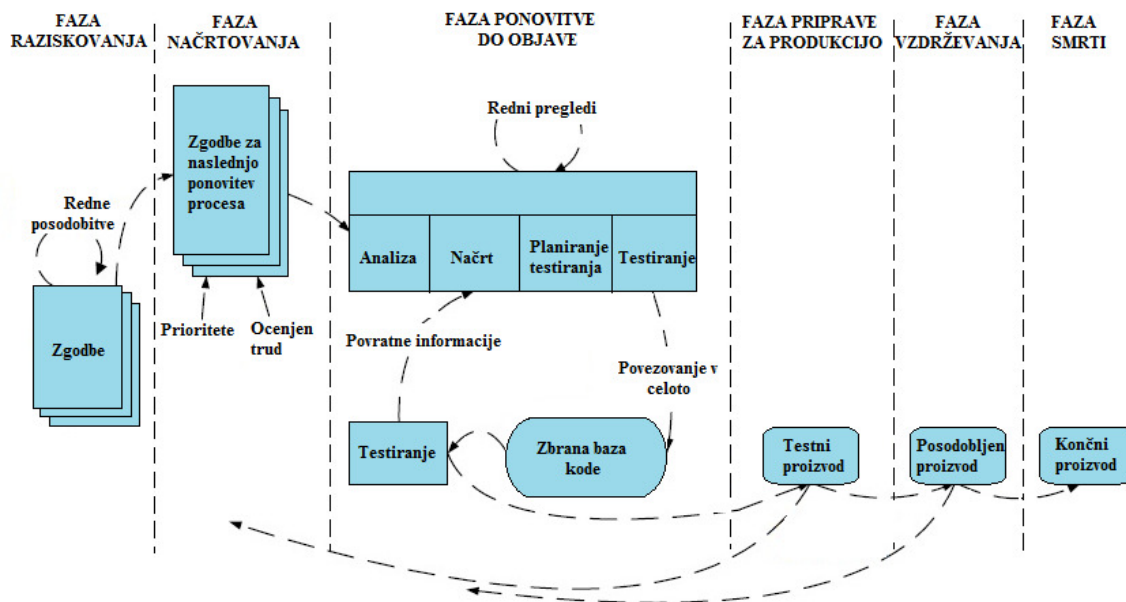
- metafora,
- preprost načrt,
- testiranje,
- preoblikovanje,
- programiranje v parih,
- skupno lastništvo programske kode,
- stalna integracija,
- 40-urni delovni teden,
- stranka ob strani in
- standardi kodiranja.

Življenjski cikel metode razvoja ekstremnega programiranja je sestavljen iz šestih faz, ki so prikazane na Sliki 7 (Beck, 1999, str. 100–105):

- **Faza raziskovanja** je namenjena temu, da naročnik opiše zgodbe in zahteve, ki jih želi imeti v prvi različici rešitve, pri čemer vsaka zgodba predstavlja posamezno funkcijo končnega programa. Razvojna ekipa se v tej fazi spozna z orodji, tehnologijo in praksami, ki jih bodo pri projektu potrebovali, hkrati pa s pomočjo prototipa preveri arhitekturne možnosti. Trajanje te faze je odvisno od obstoječega znanja in izkušenj razvojne ekipe, običajno pa traja od nekaj tednov do nekaj mesecev.
- **Faza načrtovanja** je namenjena postavitvi prioritet in določitvi vrstnega reda razvoja posamezne zgodbe – funkciji programa. Razvijalci za vsako zgodbo ocenijo potrebno količino časa in pripravijo terminski načrt. Trajanje te faze je omejeno na nekaj dni, terminski načrt pa ne predvideva več kot dva meseca dela.
- **Faza ponovitve do objave** vključuje več ponovitev sistema pred objavo prve različice. Pripravljen terminski načrt se razdeli na določeno število ponovitev, implementacija vsake pa traja od enega do štirih tednov. Prva ponovitev vsebuje arhitekturo celotnega sistema, kar se doseže z izbiro zgodbe, ki se navezuje na celoten sistem. Naročnik za vsako ponovitev – iteracijo izbere zgodbo, nato pa zanjo pripravi funkcionalni test, ki se izvede na koncu iteracije. Ob koncu zadnje iteracije je sistem pripravljen na produkcijo.
- **Faza priprave za produkcijo** obsega dodatno testiranje in preverjanje zmogljivosti rešitve, preden je ta predana naročniku. Faza dopušča dodatne spremembe, za katere se določi, ali bodo vključene v trenutno ali v naslednjo različico.
- **Faza vzdrževanja** nastopi, ko je prva različica rešitve primerna za produkcijo. V tej fazi nastopi tudi uporabniška podpora, razvija se še dodatne zahteve, ki so bile ugotovljene v predhodni fazi, kar pomeni nove iteracije.

- **Faza smrti** nastopi, ko naročnik nima več dodatnih zgodb, ki bi jih lahko implementirali, kar pomeni, da rešitev zadovoljuje naročnika. Ta faza lahko nastopi tudi v primeru, ko rešitev ne zadovoljuje naročnika ali pa je dodaten razvoj postal predrag.

Slika 7: Življenjski cikel modela XP



Vir: A. Pekka, S. Outi, R. Jussi & W. Juhani, *Agile software development methods*, 2002, str. 19.

1.3.2 Scrum

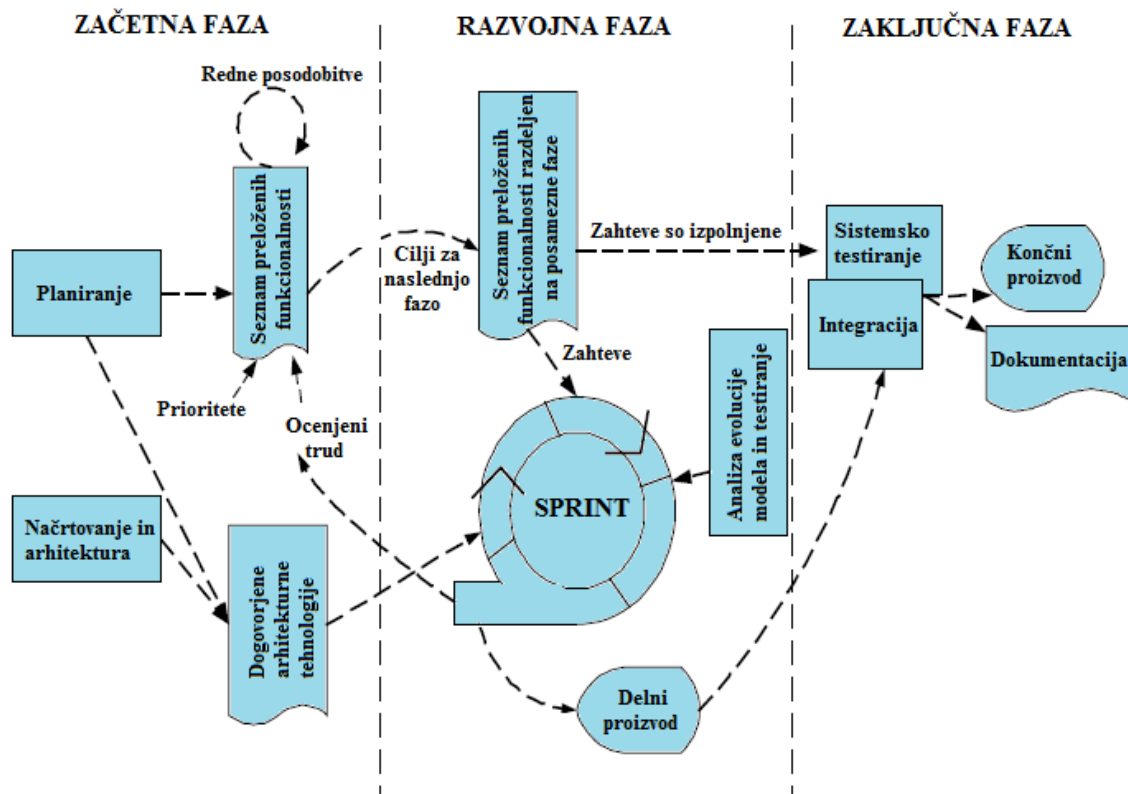
Scrum je iterativen in inkrementalen proces razvoja informacijskih rešitev, osredotočen na delo razvojne skupine. Osredotoča se na to, kako naj člani skupine delujejo, da pripravijo dovolj prilagodljivo rešitev, ki bo kljubovala nenehno spreminjajočemu se okolju (Awad, 2005, str. 10).

Osnovna ideja te metodologije je razviti informacijsko rešitev po delih oziroma inkrementih, pri čemer se vsak posamezni del skozi celoten proces razvoja stalno dopolnjuje in izboljšuje skozi ponavljajoče se cikle. Posamezni deli projekta so razdeljeni na sprints, ki predstavljajo 30-dnevne cikle, kar je prikazano na Sliki 8. Razvojna ekipa na začetku vsakega sprinta oceni delo, ki ga bodo v tem ciklu lahko opravili, nato pa na podlagi kratkih dnevnih sestankov spremljajo napredovanje zastavljenih nalog. Rezultat posameznega sprinta predstavlja del delujoče in testirane informacijske rešitve (Abrahamsson et al., 2002, str. 27–36).

Highsmith (2002, str. 134) podrobneje opisuje fazo 30-dnevnega cikla sprinta, ki jo definira kot naslednjo: na začetku sprinta si razvojna ekipa razdeli naloge, nato pa si vsak

posameznik prizadeva, da bo svojo nalogo uspešno opravil. Pri svojem delu se ne zanašajo na formalne plane in razvojno dokumentacijo, temveč na svoje izkušnje ter znanje, o napredku pa ekipa poroča na kratkih dnevnih sestankih.

Slika 8: Model Scrum



Vir: P. Abrahamsson, O. Salo & J. Ronkainen, Agile software development methods: Review and analysis, 2002, str. 28.

Schwaber (1996, str. 3) ugotavlja, da so skupne lastnosti podjetij, ki pri svojem razvoju informacijskih rešitev uporabljajo metodo Scrum, naslednje:

- velike produkte razdelijo na manjše, obvladljive dele, ki jih lahko majhna skupina razvojnikov ustvari v kratkem času;
- celoten projekt se nadaljuje sistematično, kljub temu, da člani skupine na začetku ne morejo določiti popolnega in stabilnega dizajna rešitve;
- velikim skupinam omogoča delo kot majhne skupinice, kar dosegajo s porazdelitvijo dela na manjše postopke, sinhronizacijo med skupinicami ter nenehnim iskanjem in odpravljanjem napak.

V fazi posameznega sprinta spremembe zahtev niso mogoče in se jih ne sprejema, kar metodi Scrum zagotavlja potrebno stabilnost (Highsmith, 2002, str. 134). Schwaber (1996,

str. 4) dodatno ugotavlja, da kljub povečani prožnosti in tveganju metodologija vseeno zagotavlja ohranjanje nadzora nad projektom, kar ji omogoča sledenje naslednjim načelom:

- prilagodljivost na tehnične spremembe in spremembe zahtev trga omogočajo implementacijo najboljšega možnega izdelka;
- majhne razvojne skupine omogočajo boljšo komunikacijo in izmenjavo tihega, neformalnega znanja, kar posledično zmanjšujejo režijske stroške;
- celotno delo se razdeli na posamezne naloge, ki jih opravijo posamezne skupine;
- sposobnost, da lahko kadarkoli posreduješ produkt, ki ga zahteva stranka ali konkurenca.

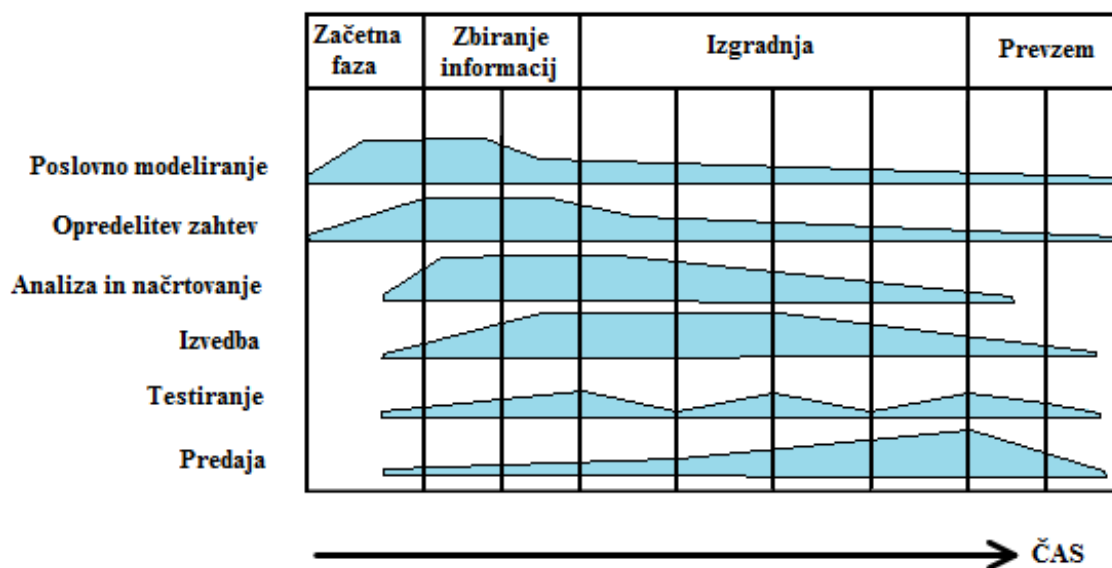
1.3.3 Model RUP

Model RUP so razvili avtorji poenotene jezika modeliranja (angl. *Unified Modeling Language*, v nadaljevanju UML), z namenom združiti najboljše postopke obstoječih procesov razvoja informacijskih rešitev. Za model je značilen iterativen pristop, kjer se v vsaki ponovitvi razvije del funkcionalnosti informacijske rešitve. Naslednje značilnosti modela so uporaba vizualnega modeliranja s pomočjo UML, obvladovanje zahtev, ki se jih spremlja od začetka do konca razvoja in vključuje tudi njihove morebitne spremembe ter razbitje celotne rešitve na več delov, ki se jih razvija ločeno. Uspešnost modela zagotavlja stalno preverjanje kakovosti in funkcionalnosti rešitev ter nadzor nad spremembami. Model RUP ni model v pravem pomenu, temveč predstavlja ogrodje, v katerem si lahko razvojno podjetje prilagodi proces in tehnike, ki so najbolj primerne za razvoj njegove rešitve. Na Sliki 9 so prikazane faze v ciklu razvoja informacijskega sistema, ki si sledijo zaporedno, vendar se med seboj zaradi iterativne narave modela prepletajo. V vsaki fazi se ponavljajo naslednje ključne aktivnosti: poslovno modeliranje, opredelitev zahtev, analiza in načrtovanje, izvedba, testiranje ter predaja (Abrahamsson et al., 2002, str. 55–61).

Abrahamsson et al. (2002, str. 55–61) definirajo naslednje faze modela:

- začetna faza,
- zbiranje informacij,
- izgradnja in
- prevzem.

Slika 9: Model RUP



Vir: P. Abrahamsson, O. Salo & J. Ronkainen, *Agile software development methods: Review and analysis*, 2002, str. 56.

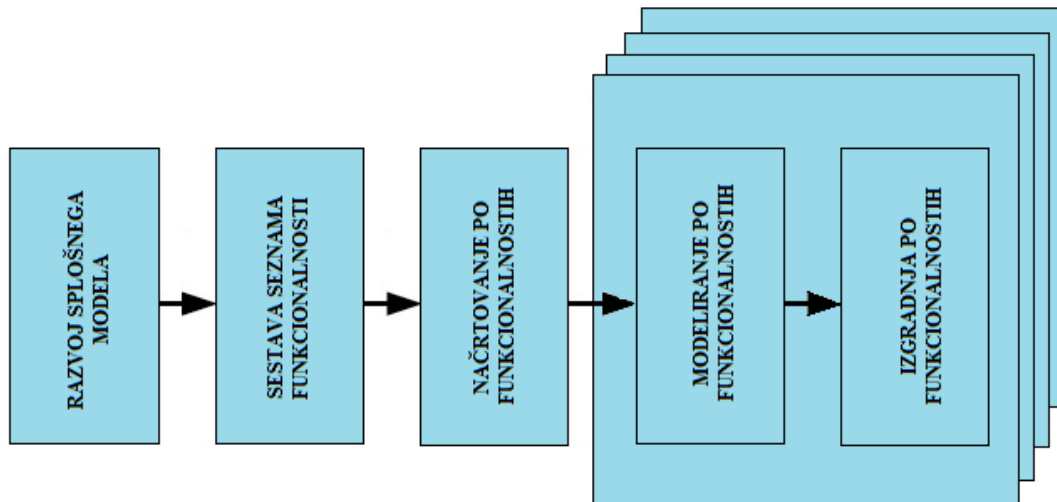
1.3.4 Funkcionalno voden razvoj informacijskih rešitev – model FDD

Gre za okreten in prilagodljiv pristop za razvoj informacijskih sistemov ter programskih rešitev, za katerega je značilno, da ne zajame hkrati celotne informacijske rešitve, temveč celoten problem (informacijsko rešitev) razdeli na manjše funkcionalno zaokrožene celote, ki jih je možno razviti v časovnem okviru dveh tednov (Abrahamsson et al., 2002, str. 47–55). Pekka et al. (2002, str. 47) ga definirajo kot agilen in prilagodljiv pristop k razvoju informacijskih sistemov, ki ne pokriva izgradnje celotne informacijske rešitve, temveč se osredotoča predvsem na fazi načrtovanja in izgradnje, čeprav je zasnovan tako, da se ga lahko uporabi tudi v drugih fazah.

Za model je značilno, da je razdeljen na pet glavnih aktivnosti ali faz, ki se razvijajo iterativno, kar je prikazano na Sliki 9. Začetne tri faze se opravi na začetku projekta, zadnji dve pa sta ponavljajoči, kar modelu omogoča, da informacijsko rešitev hitro prilagodi spremembam zahtev (Ambler, 2005). Awad (2005, str. 12) ugotavlja, da so faze modela FDD naslednje:

- razvoj splošnega modela,
- sestava seznama funkcionalnosti,
- načrtovanje po funkcionalnostih,
- modeliranje po funkcionalnostih in
- izgradnja po funkcionalnostih.

Slika 10: Model FDD



Vir: J. Highsmith, *Agile Software Development Ecosystem*, 2002, str. 152.

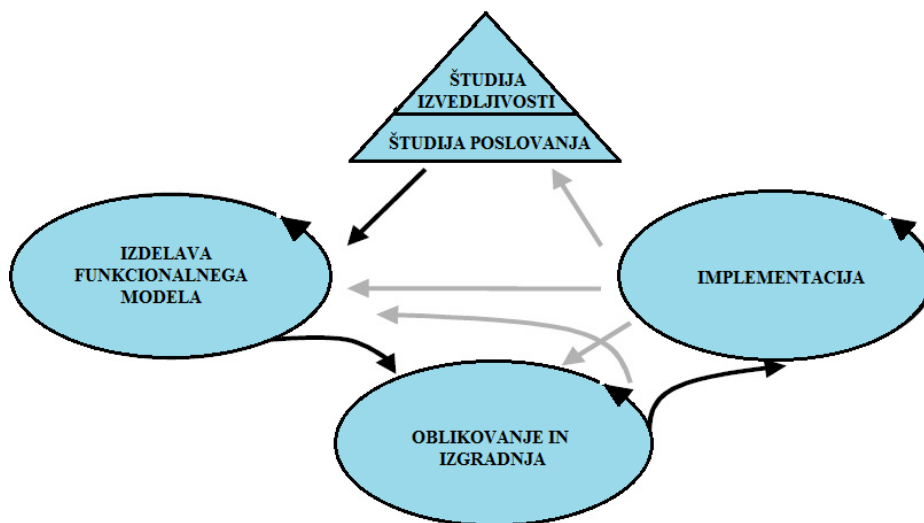
1.3.5 Dinamični sistemski model razvoja informacijskih rešitev – model DSDM

Pri razvojnem modelu DSDM se najprej opredeli potreben čas in sredstva, ki so na voljo za izgradnjo informacijske rešitve, nato pa se opredeli funkcionalnosti, ki jih je možno razviti v teh okvirih (Pekka et al., 2002, str. 61).

Clifton in Dunlap (2003) ugotavljata, da se model DSDM osredotoča na zagotavljanje poslovnih rešitev in ne le na aktivnosti ekipe, kar zagotavlja poslovni namen projekta, še preden je ta dejansko končan.

Model sestavlja pet faz, ki so vidne na Sliki 10. Prvi dve fazi – študija izvedljivosti in študija poslovanja sta zaporedni ter se izvedeta samo enkrat, medtem ko se ostale tri faze, ki predstavljajo izdelavo funkcionalnega modela, oblikovanje in izgradnjo ter implementacija v iteracijah, ponavljajo vse do zaključka projekta. Čas trajanja projekta je že na začetku vnaprej določen, običajno pa traja od nekaj dni do nekaj tednov (Abrahamsson et al., 2002, str. 61–68).

Slika 11: Model DSDM



Vir: J. Highsmith, *Agile Software Development Ecosystem*, 2002, str. 140.

Alexandrou (2011) ugotavlja, da za model DSDM velja naslednjih devet načel, s katerimi se strinjata tudi Clifton in Dunlap (2003), ki dodata še deseto načelo:

- sodelovanje in vključevanje prihodnjih uporabnikov rešitve v razvoj;
- razvojna ekipa mora biti pooblaščen za sprejemanje odločitev;
- osredotočanje na pogosto dobavo novih različic;
- primernost rešitve in njena uporaba predstavlja kriterij kakovosti;
- inkrementalni in iterativni razvoj zagotavljata realizacijo rešitve;
- spremembe v razvoju so reverzibilne;
- najtežje zahteve se razdela in razjasni na začetku procesa, pred dejansko izgradnjo;
- testiranje je integrirano skozi celoten življenjski cikel;
- v projekt so vpete vse interesne skupine, na katere se bo rešitev nanašala;
- pravilo 80 % proti 20 %, ki predpostavlja, da se 80 % rešitve razvije v 20 % potrebnega časa za celoten razvoj rešitve.

1.3.6 Prilagodljivi razvoj programskih rešitev – model ASD

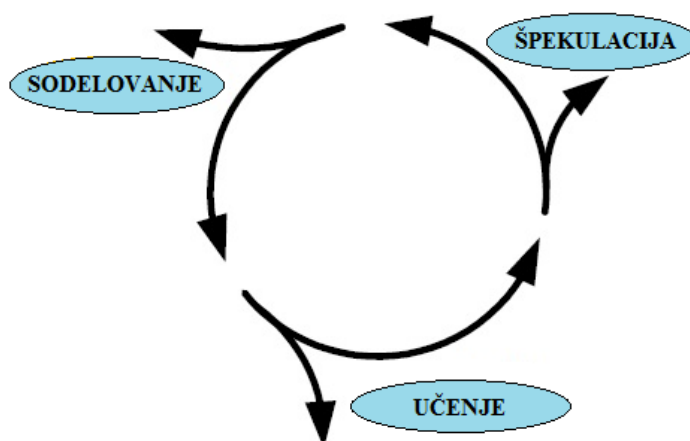
Za uspeh v informacijskem okolju, polnem sprememb in tehnoloških novosti, je treba imeti pristop, ki se je zmožen na spremembe odzivati in se jim tudi prilagoditi. Model ASD sprememb ne zavrača, temveč se jim prilagodi in jih vključi v razvoj (Highsmith, 2002, str. 173).

Model ASD se osredotoča na težave, ki nastajajo pri razvoju večjih in zapletenih rešitev ter projektov s hitrimi in velikimi spremembami. Pristop k razvoju je v tem modelu prilagojen nepredvidljivemu okolju, v katerem je planiranje projekta zelo oteženo, zato je statični plan zamenjan z dinamičnim in špekulativnim razvojnim ciklom informacijske rešitve. Metoda spodbuja inkrementalni in iterativni razvoj s konstantno izdelavo prototipov (Abrahamsson et al., 2002, str. 68–73).

Na Sliki 11 je prikazan razvojni cikel modela ASD, v katerem se ponavljajo naslednje faze (Awad, 2005, str. 14):

- špekulacija: v tej fazi se predvidi nejasnosti in neznani del projekta;
- sodelovanje: pomen timskega dela je ključen;
- učenje: predstavlja zmožnost prepoznavanja in odzivanja na napake ter sprejemanje zahtev, ki jih bodo odpravile.

Slika 12: Model ASD



Vir: M. A. Awad, Comparison between Agile and Traditional Software Development Methodologies, 2005, str. 15.

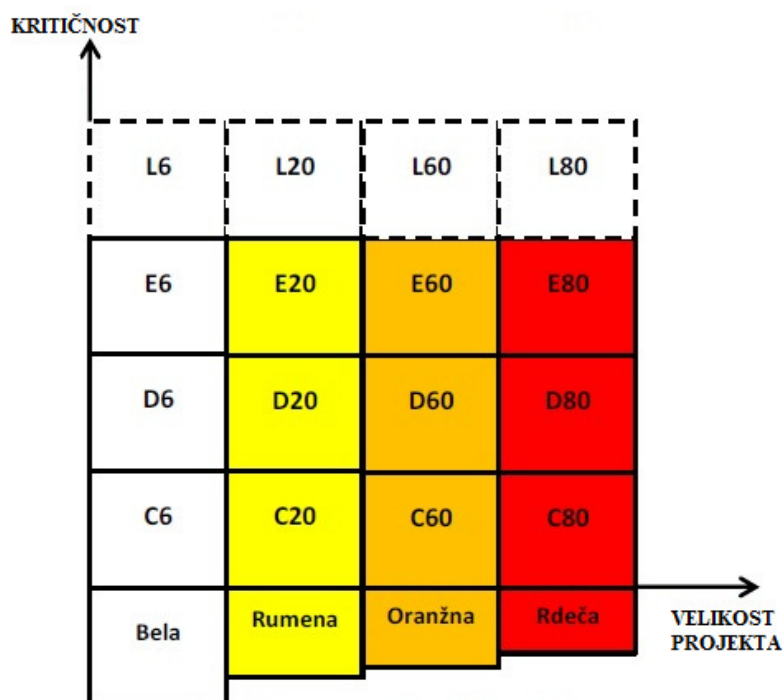
1.3.7 Metodologije družine Crystal

Metodologije družine Crystal vsebujejo veliko različnih agilnih metodologij, med katerimi se lahko odločamo in glede na značilnosti posameznega projekta izberemo tisto, ki je najustreznejša. V družino metodologij so poleg metod izgradnje vključeni tudi principi za prilagoditev metodologije potrebam projekta (Pekka et al., 2002, str. 36).

Metodologije so označene z barvami, po katerih se nato tudi imenujejo in predstavljajo težavnost metodologije, kar je razvidno na Sliki 12. Metodologije se ločijo glede na dva kriterija, in sicer sta to težavnost ter kritičnost. Večji in težavnejši projekti zahtevajo težjo

metodologijo in predstavljajo tudi večjo kritičnost izgub podatkov na informacijsko rešitev. Kritičnost je označena z naslednjimi črkami: C (angl. *Confort*) – izguba udobnosti; D (angl. *Discretionary*) – izguba denarja; E (angl. *Essenital money*) – izguba pomembnega dela denarja in L (angl. *Life*) – izguba življenja (Abrahamsson et al., 2002, str. 36–46):

Slika 13: Metodologije družine Crystal



Vir: P. Abrahamsson, O. Salo & J. Ronkainen, *Agile software development methods: Review and analysis*, 2002, str. 37.

1.4 Primerjava tradicionalnih z agilnimi metodologijami

Meja med tradicionalnimi in agilnimi metodologijami ter področji, kjer jih lahko uporabljamo, ni točno določena, čeprav praksa kaže, da določene metode niso primerne za vse posameznike ali projekte (Pekka et al., 2002, str. 8).

Kot ugotavljajo Yu et al. (2012, str. 163) je ena izmed razlik med tradicionalnimi in agilnimi metodologijami ta, da lahko slednje hitro ter poceni realizirajo rešitve relativno kompleksnim sistemov, kljub zelo slabo definiranim zahtevam, česar tradicionalne niso zmožne. Agilne metodologije poudarjajo sodelovanje razvojne ekipe, sodelovanje z naročnikom, hitro odzivanje in prilagoditve na spremembe, česar ne moremo trditi za tradicionalne metodologije. Te v nasprotju z agilnimi poudarjajo pogodbe, načrte, dokumentacijo in predpisujejo orodja za izgradnjo.

Tradicionalne metode razvoja informacijskih sistemov so prisotne že zelo dolgo časa. Pri velikih in manjših projektih se je najpogosteje uporabljal slapovni model razvoja, ki se je izkazal za zelo uspešnega. Kljub uspehom pa je imel spiralni model pomanjkljivosti, kot so izredna linearnost, nefleksibilnost ob spremenjenih zahtevah in potreba po formaliziranih postopkih ter obsežni dokumentaciji, ne glede na velikost projekta. Odgovor na pomanjkljivosti tradicionalnih metod je predstavljala metodologija ekstremnega programiranja, ki velja za prvo agilno metodo. Agilne metode lahko z uporabo več različnih tehnik upoštevajo in implementirajo nestabilne ter hitre spremembe zahtev. Vse agilne metodologije poudarjajo sodelovanje med razvojno ekipo in naročniki ter podpirajo hitro dobavo programskih rešitev. Razlike med tradicionalnimi in agilnimi metodami so prikazane v Tabeli 1 (Awad, 2005, str. 35).

Tabela 1: Primerjava tradicionalnih metod z agilnimi

	Tradicionalne metode	Agilne metode
Pristop	predvidljiv	prilagodljiv
Merjenje uspeha	skladno s planom	poslovna vrednost
Velikost projekta	veliki	majhni
Način vodenja	avtorski	decentralizirani
Pogled na spremembe	nesprejemljivost sprememb	prilagodljivost na spremembe
Kultura	ukazovanje – nadzor	vodenje - sodelovanje
Dokumentacija	visoka stopnja in veliko	nizka stopnja in malo
Usmerjenost	na procese	na ljudi
Cikličnost	omejena	na številna obdobja
Domena	predvidljivo	nepredvidljivo/raziskovalno
Načrtovanje vnaprej	celovito	minimalno
Donosnost naložb	na koncu projekta	na začetku projekta
Velikost skupine	velika	majhna/kreativna

Vir: M. A. Awad, A Comparison between Agile and Traditional Software Development, 2005, str. 35.

1.4.1 Omejitve tradicionalnih metodologij

Vzroki napak pri razvoju informacijskih sistemov z uporabo tradicionalnih metodologij so ključni tudi pri sodobnem razvoju, kjer neuspeh razvoja največkrat ni posledica slabe metodologije ali napak v informatizaciji, temveč pomanjkanje poznavanja poslovnega okolja ter napak, narejenih pri analizi in modeliranju poslovnega okolja (Zornada et al., 2002, str. 223).

Tradicionalne metodologije so prisotne že zelo dolga časa in so znane po svoji uspešnosti. Manj znane pa so njihove slabosti. Ena izmed kritik teh metodologij je, da so zelo birokratske, saj predpisujejo ogromno količino spremljajoče dokumentacije projekta, ki jo je treba voditi, kar privede do tega, da se zaradi nje upočasni tempo razvoja. Druga kritika izhaja iz dejstva, da tradicionalne metodologije temeljijo na zaporednem razvoju, kar

pomeni, da so povratne informacije stranke in testiranje postavljene v zadnjo fazo življenjskega cikla projekta. Neustreznost ali pomanjkljivost rešitve se tako ugotovi šele, ko je ta že popolnoma končana, kar je lahko tudi kritično (Fowler, 2005).

Pri tradicionalnih metodah je treba še pred začetkom izgradnje definirati vse zahteve in potrebne spremembe, poleg tega pa se lahko na začetku definirane zahteve v času izgradnje tudi spremenijo, kar v časih hitrega spreminjanja in hude konkurence predstavlja omejitev teh tehnologij. Ostale omejitve teh tehnologij so (Gradišar et. al., 2005, str. 272–273):

- Neupoštevanje potreb managerjev podjetja, ki jim sistemi ne nudijo dovolj podpore pri odločanju in ne vključujejo strateških ter taktičnih ciljev.
- Razvoj obsežnejših informacijskih sistemov lahko traja več let in lahko se zgodi, da informacijski sistem ob implementaciji ne zadovoljuje uporabnikovih zahtev ter potreb, ki so se v času izgradnje že spremenile, kar vodi v nezadovoljstvo končnih uporabnikov.
- Dokumentacija projekta je namenjena osebam, vključenim v analizo ter izgradnjo in ne končnim uporabnikom.
- Neambiciozno načrtovanje in nepopolnost sistemov je lahko posledica tega, da se pogosto avtomatizira le delo, ki se je prej opravljalo ročno, poleg tega pa takšno načrtovanje ne ustvarja sistemov, ki bi prinesli radikalne spremembe ter dejanske koristi.
- Zaradi dolgotrajnosti razvoja sistema se lahko zgodi, da se nujne zahteve, ki morajo biti realizirane, vgradi v že obstoječe sisteme brez ustreznega načrtovanja, kar pripelje do neobvladljivih sistemov, ki se težko vzdržujejo.

Uspeh ali neuspeh projektov programske opreme je odvisen od sposobnosti odzivanja na spremembe zahtev in trga, ki jo ima metodologija vgrajeno. Sprejemanje sprememb predstavlja glavno razliko med tradicionalnimi in agilnimi metodologijami (Williams & Cockburn, 2003, 39–43). S trditvijo se strinjajo tudi Gradišar et al. (2005, str. 269), ki ugotavljajo, da so dolgotrajnost, visoka cena in zelo omejene možnosti spreminjanja glavne omejitve tradicionalnih metodologij razvoja informacijskega sistema. Te so zato primerne le za nekatere vrste rešitev v individualiziranih poslovnih sistemih, pri katerih so zahtevane spremembe zelo dobro definirane.

Tudi Fowler (2005) ugotavlja, da je neupoštevanje sprememb glavna omejitev tradicionalnih metodologij. Te namreč v življenjskem ciklu razvoja ne dovoljujejo sprememb funkcionalnosti projekta, to pa je v današnjem okolju, ki se v tehnološkem smislu zelo hitro spreminja, skoraj nemogoče.

Tradicionalne metodologije so usmerjene na postopke in procese, ne pa na ljudi. Cilj tradicionalne metodologije je pripraviti proces razvoja rešitve, ki bo uporaben za vse – ne glede na njihovo znanje in izkušnje. V realnosti pa dobro predpisan razvojni proces ne

more nadomesti pomanjkanja znanja razvojne ekipe, medtem ko lahko spretnost in izkušnje slednje nadomestijo slabo predpisani proces (Fowler, 2005). Podobno menita tudi Cockburn in Highsmith (2001, str. 131–133), ki ugotavljata, da so ljudje najpomembnejši dejavnik pri razvoju programske opreme.

1.4.2 Omejitve agilnih metodologij

Čeprav agilne metodologije v primerjavi s tradicionalnimi prinašajo mnogo prednosti se je treba zavedati tudi njihovih slabosti, ki so se pokazale pri njihovi praktični uporabi. Ena izmed slabosti agilnih metodologij je znatno zmanjšanje oziroma v nekaterih primerih tudi zavračanje dokumentacije, saj trdijo, da jo uspešno nadomešča programska koda in v njej evidentirane opombe razvijalcev (Yu et al., 2012, str. 164).

Zapostavljanje dokumentacije izpostavlja tudi Boehm (2002, str. 64–69), ki ugotavlja, da tradicionalne metodologije zmanjšujejo tveganje projekta z vlaganjem v arhitekturo rešitve, ki lahko zunanjim strokovnim pregledovalcem olajša vpogled v planiranje in uporabo programske rešitve, čeprav to podraži celoten projekt. Agilne metodologije zavračajo pretirano birokracijo in se opirajo na spretnosti, izkušnje in komunikacijo med člani. Znanja ne zapisujejo v dokumentacijo, temveč uporabljajo delitev tihega znanja znotraj članov, kar pa lahko vodi do arhitekturnih napak, ki jih zunanji pregledovalci zaradi pomanjkljive dokumentacije ne morejo odkriti.

Fowler (2005) ugotavlja, da komunikacija na štiri oči, ki je značilna za agilne metodologije, postaja vse težja, ko je razvijalcev več kot dvajset in začne celo predstavljati oviro. Ravnanje z večjimi skupinami predstavlja glavno omejitev agilnih metodologij, kar ugotavljata tudi Cockburn in Highsmith (2001, str. 131–133), ki menita, da je agilni razvoj za večje skupine težji.

Eno izmed načel agilnih metodologij pravi, da je največja prednost zadovoljiti kupca s hitro in konstantno dostavo delujoče programske rešitve, kar pa ni nujno prednost oziroma lahko postane tudi ovira. Pregled zgodnjih rezultatov prejete programske rešitve lahko privede do ugotovitev, da bodo potrebne večje predelave. Načrtno usmerjen proces je najbolj potreben visoko zanesljivi programski opremi, kjer pridejo do izraza cilji tradicionalnih teorij, kot so predvidljivost, ponovljivost in optimizacija. Večina agilnih tehnik ne podpira pregledov in inšpekcijskih pregledov kode v življenjskem ciklu projekta, temveč zaupa svojim ocenam kontrolnih mehanizmov in jim sledi (Boehm, 2002, str. 64–69).

Cockburn in Highsmith (2001, str. 131–133) uspeh projekta pripisujeta tudi ljudem. Če so ljudje na projektu dovolj dobri, lahko uporabimo kateri koli postopek in izpolnili bodo vse naloge. Če niso dovolj dobri, ne bo noben postopek popravil in prikriil njihove

neustreznosti. Awad (2005) se z njima strinja in ugotavlja, da je uspeh agilnih metodologij treba pripisati ekipi dobrih ljudi, ne pa načelom metodologij.

Agilne metodologije poudarjajo vključevanje naročnika v proces razvoja programske opreme in ga obravnavajo kot nujni del ekipe, ki s svojim znanjem pomaga razvoju in testiranju. Do problema lahko pride v primeru, ko naročnikovo znanje ni na ravni, ki bi zadostovala potrebam celotnega razvojnega obdobja. Dodaten problem lahko predstavljajo naročniki, ki niso usklajeni ali pa med njimi obstajajo konflikti (Boehm, 2002, str. 64–69).

2 ELEKTRONSKO POSLOVANJE

Elektronsko poslovanje lahko opredelimo kot način delovanja, organiziranja in notranjega ter zunanjega sodelovanja organizacijskega sistema, podprtega z dosežki informacijskih in telekomunikacijskih tehnologij, ki obsegajo aplikacije, storitve ter tehnološko infrastrukturo (Krisper et al., 2003). Resnik (2010) dodaja, da gre pri elektronskem poslovanju za uporabo informacijske tehnologije za potrebe poslovnih komunikacij med različnimi skupinami udeležencev – podjetji, zaposlenimi, državo in potrošniki.

Zhang, Johnson, Seltzer in Bichard (2010, str. 78–82) ugotavljajo, da lahko o elektronskem poslovanju govorimo takrat, ko podjetje s pomočjo informatizacije prenovi ali posodobi del ali celotno poslovanje.

Vse, kar delamo v sklopu poslovne dejavnosti podjetja s pomočjo računalniških omrežij in aplikacij, imenujemo elektronsko poslovanje. Elektronsko poslovanje predstavlja katerokoli obliko poslovanja, pri katerem med partnerji ni fizičnega stika, temveč je vse njuno delovanje na elektronski ravni (Gradišar et al., 2005, str. 138–139). Skrt (2002) elektronsko poslovanje pojmuje kot vse izvajanje poslovnih transakcij, upravljanje odnosov s strankami, komuniciranje znotraj podjetja in med različnimi podjetji, kupci ter državno upravo, ki poteka prek zasebnih ali javnih omrežij.

Organizacije v današnjem času doživljajo velik pritisk na zniževanje cen, zato morajo stremeti k poenostavitvi procesov in povezljivosti svojih procesov s procesi poslovnih partnerjev, kar pa jim omogoča elektronsko poslovanje (Bogataj, Pucihar & Puc, 2007). Gradišar et al. (2007, str. 204) še ugotavljajo, da uvedba elektronskega poslovanja za podjetje ne pomeni le prenosa obstoječih komunikacij s partnerji iz fizične oblike ali klasičnih medijev na internet, temveč tudi spremembe v poslovanju, ki mu omogočajo, da v polni meri izkoristi možnosti in prednosti nove informacijske tehnologije.

Doseganje učinkovitosti in uspešnosti v informacijski družbi je pogojeno z vzpostavitvijo učinkovitih oblik poslovanja, ki obsega organiziranje, sodelovanje ter povezovanje. Za vse oblike poslovanja velja, da lahko učinkovito delujejo le v primeru, ko so podprte s sodobnimi tehnološkimi dosežki s področja informacijskih in telekomunikacijskih

tehnologij. Za takšne oblike poslovanja se je uveljavil pojem elektronsko poslovanje, ki predstavlja višjo obliko oziroma kakovosten dvig poslovanja (Krisper et al., 2003).

Elektronsko poslovanje je široko področje, ki zajema tehnološke, tehnične, ekonomske, pravne in organizacijske okvirje. Prav zaradi širine pojmovanja se v angleški literaturi pojavljata dva izraza, ki pa se pomensko razlikujeta. Najprej se je pojavil pojem »electronic commerce« ali krajše »e-commerce«, ki predstavlja proces nakupa, prodaje, prenosa, izmenjave izdelkov ali storitev prek internega ali drugih omrežij. Ker je takšna opredelitev e-poslovanja omejena le na proces trgovanja, se je v angleški literaturi pojavil še izraz »e-business«, ki dodatno vključuje še sodelovanje s poslovnimi partnerji, servisiranje potrošnikov, učenje na daljavo in tesnejše sodelovanje zaposlenih znotraj organizacijske ravni. Slovenski izraz e-poslovanje se vsebinsko ujema z angleškim izrazom »e-business«, ki hkrati vsebuje tudi »e-commerce« (Chaffey, 2009, str. 9–13; Gradišar et al., 2005, str. 138–139).

2.1 Oblike in področja elektronskega poslovanja

Podjetja, državne in javne službe ter posamezniki kot potrošniki oziroma končni uporabniki so tri glavne skupine udeležencev e-poslovanja. Poslovanje poteka med skupinami, kakor tudi znotraj posamezne skupine. Glede na vrsto udeležencev in obliko sodelovanja med njimi ločujemo naslednje vrste e-poslovanja (Combe, 2006, str. 61–67; Gradišar et al., 2005, str. 140–146):

- **Med podjetji** (angl. *Business to Business*, v nadaljevanju B2B). Gre za največjo in najstarejšo obliko e-poslovanja, ki je bila v prvotni obliki računalniška izmenjava podatkov. Ta je podjetjem omogočala, da si poslovne dokumente, kot so naročilo, potrditev naročila, dobavnica in račun, niso več izmenjevala v papirni obliki, temveč elektronsko. To je obema udeleženi stranema omogočalo cenejše in hitrejša poslovne transakcije, hkrati pa so si lahko po elektronski poti izmenjevale tudi ostale informacije, ki so jim omogočale optimiranje materialnih tokov. Danes B2B zajema povezave med podjetjem in njegovimi partnerji, elektronsko bančništvo za pravne osebe, sodelovanje v virtualnih organizacijah in elektronske ali e-tržnice, pri katerih si množica ponudnikov in povpraševalcev izmenjuje informacije ter sklepa posle.
- **Med podjetji in potrošniki** (angl. *Business to Consumer*, v nadaljevanju B2C). Ta vrsta e-poslovanja je potrošnikom najbližja in tudi najbolj poznana, saj predstavlja elektronsko bančništvo za fizične osebe, e-trgovine kot tudi izobraževanja na daljavo. Najbolj prepoznavna oblika B2C e-poslovanja so internetne trgovine ali tudi e-trgovine, v katerih končni potrošniki izdelke kupujejo prek svetovnega spleta. Nadgradnja e-trgovine sta e-blagovnica, ki vključuje več neodvisnih e-trgovin, in e-tržnica, na kateri se srečujejo različni kupci ter prodajalci.

- **Med potrošniki** (angl. *Consumer to Consumer*, v nadaljevanju C2C). Elektronsko poslovanje v tem primeru poteka med končnimi potrošniki. Pri C2C gre za elektronsko komunikacijo med potrošniki, z uporabo različnih e-klepetalnic in v primeru prodaje ter nakupa tudi e-tržnic ali e-dražb, na katerih prodajalci posredujejo informacijo o ponudbi, za katero nato povpraševalci licitirajo in dvigujejo ceno.
- **Med javno oziroma državno upravo in podjetji** (angl. *Government to Business*, v nadaljevanju G2B). Pri tej obliki elektronsko poslovanje in izmenjava podatkov poteka med podjetji in javno ali državno upravo ter obsega izvajanje javnih razpisov, napovedi za odmero davka na dodano vrednost (v nadaljevanju DDV) in oddajo obveznih obrazcev, vezanih na zdravstveno ter pokojninsko zavarovanje.
- **Med javno oziroma državno upravo in državljani** (angl. *Government to Consumer*, v nadaljevanju G2C). V tem primeru gre za sodelovanje med državljani in javno upravo, ki obsega elektronsko oddajo napovedi za dohodnino, izdajo potrdil ter izpiskov s strani upravnih enot.
- **Znotraj javne oziroma državne uprave** (angl. *Government to Government*, v nadaljevanju G2G). Gre za elektronsko poslovanje med enotami v okviru državne uprave z namensko razvitimi aplikacijami, ki omogočajo pridobivanje in posredovanje podatkov.

2.2 Prednosti in slabosti elektronskega poslovanja

Spremembe poslovanja prinašajo nove priložnosti, kot na primer sodelovanje na novih trgih, ki podjetju pred uvedbo elektronskega poslovanja niso bili dostopni (Gradišar et al., 2005, str. 159). Podobno meni tudi Skrt (2001), ki ugotavlja, da postajajo fizične razdalje z uporabo elektronskega poslovanja zanemarljive. Največje prednosti ob uporabi elektronskega poslovanja lahko pričakujejo majhna in srednje velika podjetja, ki lahko zaradi nizkih stroškov, ki jih omogoča uporaba internetnih tehnologij, v svoje poslovanje vključijo tudi velika podjetja in s svojo ponudbo konkurirajo na svetovnem trgu.

2.2.1 Prednosti elektronskega poslovanja

Obstajajo številne prednosti, ki jih prinaša uvedba e-poslovanja. Chaffey (2009, str. 29) ugotavlja, da uvedba elektronskega poslovanja podjetju prinaša prednosti, ki pozitivno vplivajo na ustvarjanje dodane vrednosti, in sicer jih deli na dve skupini:

- povečanje dohodkov, ki izhajajo iz povečanega dosega kupcev, njihove lojalnosti in ponavljajočih se nakupov obstoječih kupcev;

- prednosti ob uvedbi elektronskega poslovanja, ki se kažejo v nižjih stroških zaposlenih, transporta in materialnih stroških.

Kovačič (1998, str. 23–27) podrobneje opredeljuje prednosti, ki jih elektronsko poslovanje prinaša podjetjem in ostalim udeležencem tovrstne oblike poslovanja, in sicer so prednosti naslednje:

- **Nižji stroški nakupa.** Poslovne postopke, vezane na prodajo in nakup, spremlja množica dokumentacije, ki jo predstavljajo ponudba, naročilo, dobavnica, prevzemnica in račun. Podjetja poskušajo nakupne stroške znižati s tesnejšim sodelovanjem s svojimi ključnimi partnerji, saj lahko v tem primeru pričakujejo količinske popuste in tesnejše sodelovanje partnerjev. Elektronsko poslovanje podjetjem v procesu nakupa znižuje stroške, ki so vezani na izvedbo nakupnega procesa in kreiranje ter izmenjavo spremljajočih dokumentov.
- **Zmanjšanje obsega zalog.** Večji obseg zalog, ki so potrebne v proizvodnem procesu, za organizacijo pomeni večji strošek. Bolj kot je organizacija povezana s svojimi dobavitelji, manjše so lahko njene lastne zaloge surovin in materialov, ki jih potrebujejo v svojem proizvodnem procesu, hkrati pa manjši obseg zalog pripomore tudi k večji preglednosti nad zalogami ter učinkovitejši izrabi skladiščnih prostorov. Elektronsko poslovanje organizacijam omogoča zagotavljanje optimalne količine zalog, ki so jim na voljo ob pravem času in na pravem mestu.
- **Skrajšanje poslovnega cikla.** Nižje stroške, hitrejši odziv na zahteve trga in posledično tudi primerjalno prednost pred ostalimi proizvajalci si lahko organizacija zagotovi tako, da skrajša svoj poslovni cikel. Elektronsko poslovanje med organizacijo in njenimi ključnimi dobavitelji ter kupci lahko močno skrajša pošiljanje naročil, računov in ostale spremne dokumentacije.
- **Razvijanje učinkovitejše in uspešnejše pomoči ter povezovanje z odjemalci.** Organizacija lahko z elektronskim poslovanjem svojim odjemalcem omogoči podroben opis ponudbe in vpogled v stanje ter sledenje njihovih naročil in tako bistveno vpliva na njihovo zadovoljstvo ter uporabniško izkušnjo, ki so jo s tem deležni.
- **Znižanje stroškov prodaje in trženja ter ustvarjanje novih tržnih priložnosti.** Pri elektronskem poslovanju obseg prodaje ni več omejen s številom razpoložljivega prodajnega kadra, saj omogoča neposredno naročanje in hkrati minimizira oziroma popolnoma odpravlja čas, ki je bil potreben za fizično izpolnjevanje ter izmenjavo naročil. Prodajni kader lahko tako več časa posveti iskanju novih tržnih priložnosti oziroma skrbi za poprodajne aktivnosti, ki so namenjene zadovoljstvu kupca.

Tudi Perrott (2005, str. 76) izpostavlja prednosti, ki jih prinaša uvedba elektronskega poslovanja. Glede na njegovo delitev so prednosti razdeljene v naslednje štiri kategorije:

- **Dodana vrednost:** kategorija obsega prednosti, ki izhajajo iz povečanega zaupanja ter posledično tudi zadovoljstva kupcev, kar se odraža v zelo dobro podprtih pred- in poprodajnih storitvah.
- **Pritiski konkurence:** pritiski konkurence, ki je elektronsko poslovanje že uvedla, lahko povzročijo stagnacijo podjetja, ki tega ni naredilo. Naprednejša konkurenca v tem primeru pomeni spodbudo v smeri elektronskega poslovanja.
- **Stroškovne koristi:** kategorija obsega prednosti, ki izhajajo iz nižjih transakcijskih in operativnih stroškov, ki jih prinaša elektronsko poslovanje.
- **Tržne prednosti:** kategorija se nanaša na spodbude, ki izhajajo iz širitve trgov in večjega tržnega deleža.

Resnik (2010) deli prednosti elektronskega poslovanja na dve glavni skupini, in sicer na prednosti za končne uporabnike ter prednosti za podjetja. Prednosti za končne uporabnike so naslednje:

- storitve so prilagojene v večji meri;
- uporabnik ima glede dostopa do storitev in informacij na voljo več možnosti;
- uporabniki so vključeni v oblikovanje storitev in njihove izboljšave;
- stroški nakupa so nižji;
- uporabniki so vključeni v proces odločanja;
- za opravljanje storitev in pridobivanje informacij se porabi manj časa;
- storitve niso vezane na delovni čas podjetij.

Prednosti, ki jih elektronsko poslovanje prinaša podjetjem, so po mnenju Resnika (2010) naslednje:

- kljub višjim začetnim investicijskim stroškom so dolgoročni stroški poslovanja nižji;
- storitve za uporabnika so hitrejšje;
- omogoča boljšo kakovost storitev;
- preglednejše poslovanje;
- boljša razporeditev virov;
- manj napak in manj podvajanja nepotrebnega dela;
- višja kakovost storitev in izdelka;
- boljša in modernejša podoba podjetja v očeh končnih uporabnikov ter konkurentov.

2.2.2 Slabosti elektronskega poslovanja

Elektronsko poslovanje in njegovo uvedbo se vedno omenja v povezavi s prednostmi, ki jih bo prineslo, vendar se morajo podjetja pred njegovo uvedbo dobro zavedati tudi morebitnih nevarnosti in slabosti, ki se kažejo v skritih stroških, varnosti podatkov, marketinških napakah ter nedostopnosti ali nedelovanju informacijskih tehnologij (Dontigney, b.l.).

Elektronsko poslovanje prinaša tudi nevarnosti, ki so lahko strateške ali pa dejanske. Ena glavnih strateških nevarnosti je sprejemanje napačnih odločitev glede poslovnih naložb in vlaganj v elektronsko poslovanje. V vsakem gospodarstvu obstajajo tako podjetja, ki so z uvedbo elektronskega poslovanja pridobila konkurenčne prednosti, kot tudi tista, ki jih kljub vlaganjem in naporom niso uspela pridobiti. Vzroke slednjih lahko pripišemo pomanjkljivim načrtom uvedbe elektronskega poslovanja ali pa so bili uporabljeni načrti neprimerni za njihovo vrsto poslovne dejavnosti. Drugo skupino nevarnosti predstavljajo dejanska tveganja, ki so povezana z uporabo že uvedenega elektronskega poslovanja, in sicer se jih povezuje z nezadovoljstvom končnega uporabnika, ki je lahko posledica naslednjega: nedelovanje spletnih strani, kraja identitete in podatkov kupcev s strani hekerjev, težave z dobavo naročenega blaga in nedovoljeno oglaševanje (Chaffey, 2009, str. 35).

Resnik (2010) izpostavlja, da so slabosti elektronskega poslovanja v veliki meri vezane tudi na težave z uporabo in sprejemanjem informacijske tehnologije. Glavni vzroki za nesprejem predstavljajo:

- ekonomski dejavniki, ki predstavljajo dostopnost informacijske tehnologije;
- dejavniki povezovanja z učenjem, ki obsegajo spoznavanje novih tehnologij in računalniško pismenost;
- nazadostna razvitost infrastrukture in omrežja za zagotavljanje informacijske tehnologije;
- možnosti zlorabe informacijske tehnologije;
- osebni in kulturni dejavniki, ki obsegajo pomanjkanje osebnega stika ter nezaupanje v novosti.

Bogataj et al. (2007) ugotavljajo, da med majhnimi organizacijami uporaba elektronskega poslovanja narašča zelo počasi. Med dejavnike, ki uporabo ovirajo, vključujejo tehnične vzroke podjetij, kot so nepovezljivost, zasebnost in slaba omrežja. Med ovire za širšo uporabo elektronskega poslovanja pa uvrščajo notranji odpor za delitev informacij in novih načinov dela, vprašanje nadzora nad informacijami ter pomanjkanje znanja.

2.3 Elektronski račun

Elektronski račun je račun, ki ga izdajatelj izda prejemniku računa v elektronski obliki in se prek sistema posreduje do prejemnika (SEPA, E-račun, 2016).

S skupnim imenom e-račun se označujejo dokumenti – knjigovodske listine, ki so lahko izdane ali prejete in vsebujejo zapise o poslovnih dogodkih, na osnovi katerih se spreminjajo sredstva ali obveznosti do njihovih virov, prihodki ter odhodki. Obvezni elementi listine, ki so opredeljeni tako v Pravilniku o računovodstvu organizacije kot v Zakonu o davku na dodano vrednost, skrbijo za popolnost, resničnost in poštenost poslovnega dogodka, ki ga izkazujejo. Za e-račune se šteje dokumente oziroma listine, ki vsebujejo obvezne elemente, in sicer ne glede na to, kako se ta listina imenuje (Uprava Republike Slovenije za javna plačila, b.1.).

Za to, da se račun lahko upošteva kot e-račun, mora biti izdan in prejet v elektronski obliki zapisa, ki je lahko v obliki strukturiranih sporočil z dodano ustrezno vizualizacijo ali v drugih elektronskih oblikah. Pomembno je, ne glede na vrsto elektronske oblike, da je račun v elektronski obliki, ko je izdan in prejet (Elektronski računi, 2016).

Podjetja, ki pošiljajo in prejemajo velike količine računov, veliko pogosteje uporabljajo elektronsko poslovanje ter z njim povezane e-račune. Manjša podjetja in druge organizacije začenjajo razumeti, da e-računi v oskrbovalni verigi predstavljajo pomemben dejavnik, ki povečujejo učinkovitost celotnega poslovnega procesa (Pulli, 2005, str. 1).

Za podjetja je pomembno, da imajo ustrezno podporno tehnologijo, ki omogoča prevzem in izdajo e-računa, s tem pa tudi digitalizacijo procesov, vezanih na njih. Izdajanje in prejemanje e-računov podjetju omogoča višjo produktivnost, znižanje transakcijskih stroškov, izboljšanje hitrosti izvajanja procesov, natančnost in razpoložljivost potrebnih informacij ter boljše upravljanje finančnih tokov (Pulli, 2005, str. 8–25).

E-račun prinaša prednosti vsem udeležencem v procesu, ki se delijo na (SEPA, E-račun, 2016):

- **Prednosti za izdajatelje:** predstavljajo prihranek pri manipulativnih stroških in poenostavitev obstoječih procesov izdajanja računov v podjetjih.
- **Prednosti za prejemnike:** predstavljajo prejem računa neposredno v elektronsko banko, enostavnejše in hitreje plačevanje ter nižji stroški plačevanja.

Dodatne prednosti e-računa v primerjavi s klasičnim računom so naslednje (Uprava Republike Slovenije za javna plačila, b.1.):

- zaradi manjše porabe papirja je poslovanje podjetja bolj ekološko;

- podatki se pošiljajo po varni elektronski poti;
- avtomatizirana izmenjava računov pohitri poslovanje;
- boljši nadzor pri delu z računi in večja preglednost;
- avtomatski prevzem računov zmanjšuje število napak, saj prepisovanje v bazo ni več potrebno;
- omogočajo avtomatizirano delo računom (prejem, knjiženje v dokumentnem sistemu, knjiženje v računovodskem sistemu);
- zmanjša se ročno delo, poraba papirja in poštnih storitev, kar vpliva na nižje stroške poslovanja;
- omogočena je hramba računov v elektronski obliki.

V primeru, ko podjetje nima uvedene elektronske likvidacije e-računa, lahko njegova uvedba namesto zmanjšanja stroškov povzroči njihovo povečanje. Če podjetje nima urejenega celotnega procesa, vezanega na e-račune, mora vsak prejeti račun še natisniti, kar povzroča dodatne stroške.

Razloge za neuporabo e-računov lahko pripišemo tudi dejstvu, da podjetja nimajo sprejetih internih predpisov in uvedenih standardov, ki bi jim omogočali izmenjavo e-računov ter spremljajoče dokumentacije. Hkrati pa se mikro in majhna podjetja soočajo z pomanjkanjem sredstev, ki so potrebna za uvedbo e-računov (Cote, Vezina & Sabourin, 2005).

Skrtni (2006) ugotavlja, da so papirni računi po nekaj deset letih v arhivu popolnoma berljivi, česar pa ne moremo trditi za e-račune. Omejena obstojnost računalniških medijev in spremembe standardov zahtevajo nenehno skrb za vzdrževanje elektronskega arhiva, kar je povezano z dodatnimi stroški.

Pulli (2005, str. 55) navaja najpogostejše razloge, ki podjetja odvrnejo od tega, da bi prešla na elektronsko poslovanje in z njim povezane e-račune. Razlogi neuporabe e-računov so naslednji:

- kupci ali dobavitelji ne zahtevajo uporabe e-računov;
- prednosti, ki jih prinašajo e-računi, niso dovolj prepričljive;
- o e-računih je premalo znanih informacij;
- konkurenca v panogi ne uporablja e-računov;
- uvedba e-računov je predraga;
- pomanjkanje ustreznega kadra, ki bo na strani podjetja skrbel za uvedbo e-računov;
- integracija e-računov v obstoječi informacijski sistem je težavna;
- uvajanje e-računov v sistem traja predolgo.

2.4 Zakonodaja

Slovenija je bila med prvimi državami, ki je področje elektronskega poslovanja začela uvajati tudi na zakonodajnem področju, saj zakonodaja na tem področju velja že več kot deset let. Sprejeta zakonodaja ureja pravila na področju brezpapirnega poslovanja in zaščito uporabnikov na način, da predpisuje, kako zagotoviti prepoznavnost avtorja elektronskega dokumenta, kako zagotoviti nespremenljivost elektronskih dokumentov, kako dokumente hraniti in kako zagotavljati njihovo veljavnost v davčnih, sodnih in drugih postopkih (Slovenski Nacionalni forum za eRačun, 2016).

V nadaljevanju so predstavljeni zakoni, priporočila in pravilniki, ki se nanašajo na elektronsko poslovanje ter podpis, poslovanje z e-računi, izmenjavo e-računov s proračunskimi uporabniki in arhiviranje dokumentov.

2.4.1 Elektronsko poslovanje in podpis

Področje urejata Zakon o elektronskem poslovanju in elektronskem podpisu (v nadaljevanju ZEPEP) ter na njegovi podlagi izdana Uredba o pogojih za elektronsko poslovanje in elektronsko podpisovanje, ki elektronskemu podpisu priznavata enako veljavo, kot jo ima lastnoročni podpis pri dokumentih, ki so v fizični obliki. Področje je urejeno z akti, opisanimi v nadaljevanju.

Direktiva Evropskega parlamenta in Sveta 1999/93/ES o okviru Skupnosti za elektronski podpis. Na podlagi predlogov Ekonomsko-socialnega odbora in Odbora regij je bil pripravljen predlog za zagotavljanje varnosti ter zaupanja v elektronsko komuniciranje in s tem povezano oblikovanje evropskega okvira za digitalne podpise ter šifriranje. Na podlagi predlogov je bila sprejeta direktiva, katere namen je spodbujanje elektronskega poslovanja v Evropski uniji (v nadaljevanju EU). Direktiva oblikuje pravni okvir za elektronski podpis in storitve overjanja z namenom primerne delovanja notranjega trga EU (Ur.l. EU, št. OJ L 13).

Zakon o elektronskem poslovanju in elektronskem podpisu. Zakon je bil sprejet 23. junija 2000 in povzema določbe direktive EU o elektronskem podpisu ter vzorčnega zakona o elektronskem poslovanju Komisije Združenih narodov za mednarodno trgovinsko pravo (angl. *United Nations Commission on International Trade Law*). Pomembnejši členi so naslednji (Ur.l. RS, št. 98/2004-UPB1, 61/2006-ZEPT, 46/2014):

- v drugem poglavju, ki opredeljuje podatke v elektronski obliki in elektronsko sporočilo, je v 12. členu opisano zagotavljanje revizijske sledi ter možnosti za preverjanje avtentičnosti in celovitosti zapisov, podatkov ter elektronskih dokumentov;
- 15. člen tretjega poglavja definira elektronski podpis. Varen elektronski podpis, overjen s kvalificiranim digitalnim potrdilom, je izenačen z lastnoročnim podpisom.

Uredba o pogojih za elektronsko poslovanje in elektronsko podpisovanje. Uredba je bila izdana na podlagi zakona ZEPEP in določa (Ur.l. RS, št. 77/2000, 2/2001, 86/2006):

- merila, ki se uporabljajo za presojanje izpolnjevanja zahtev za delovanje overiteljev, ki izdajajo kvalificirana potrdila, in podrobnejšo vsebino notranjih pravil overiteljev. Podrobnejše tehnične pogoje za elektronsko podpisovanje in preverjanje elektronskih podpisov, časovno veljavnost kvalificiranih digitalnih potrdil, uporabo varnih časovnih žigov ter pogoje za elektronsko poslovanje v javni upravi;
- v 34. in 35. členu podrobneje opredeljuje časovni žig in določa dodatne zahteve za njegovo uporabo.

2.4.2 Poslovanje z e-računi

Tako kot račun v fizični obliki mora tudi e-račun vsebovati obvezne elemente knjigovodske listine, ki so opredeljeni v Zakonu o davku na dodano vrednost. Področje poslovanja z e-računi je urejeno z naslednjimi akti:

Zakon o davku na dodano vrednost. Zakon med drugim opredeljuje tudi poslovanje z e-računi, njegove obvezne podatke, celovitost vsebine in berljivosti računa ter definira pristnost izvora. Pomembnejši členi, vezani na e-račune, so naslednji (Ur.l. RS, št. 13/2011-UPB3, 18/2011, 78/2011, 38/2012, 83/2012, 86/2014, 90/2015):

- v 84. členu zakon izenačuje papirni in e-račun;
- 81. člen opredeljuje, da so računi vsi dokumenti na papirju ali v elektronski obliki, ki izpolnjujejo pogoje iz 80. člena o obveznostih izdajanja računov in 84. člena o papirnih ter e-računih;
- v 82. členu so opredeljeni obvezni podatki računa, ki veljajo tudi za e-račune;
- vodenje evidenc in obvezna hramba dokumentov je opredeljena v 137. členu.

Pravilnik o izvajanju ZDDV. V 152. členu pravilnika so opredeljeni pogoji hrambe elektronskih listin, ki jih je davčni zavezanec izdal, poslal ali prejel in za katere mora omogočiti takojšen ter neoviran dostop do podatkov o izvoru, prejemniku, času in kraju pošiljanja ali prejemu elektronske listine (Ur.l. RS, št. 141/2006, 52/2007, 120/2007, 21/2008, 123/2008, 105/2009, 27/2010, 104/2010, 110/2010, 82/2011, 106/2011, 108/2011, 102/2012, 54/2013, 85/2014, 95/2014, 39/2016).

Zakon o davčnem postopku. Zakon definira obveznost dokumentiranja za namene izvajanja zakonov o obdavčenju in hrambo dokumentacije. Na e-račune se navezujejo naslednji členi (Ur.l. RS, št. 13/2011-UPB1, 32/2012, 94/2012, 101/2013-ZDavNep, 111/2013, 25/2014-ZFU, 40/2014-ZIN-B, 90/2014, 91/2015):

- 31. člen definira obveznost dokumentiranja, ki velja tudi za e-račune;

- hramba dokumentacije v fizični ali elektronski obliki do poteka absolutnega zastaralnega roka pravice do izterjave davka je definirana v 32. členu;
- elektronska obdelava podatkov je v urejena v 38. členu z naslednjimi zahtevami:
 - oblika podatkov mora omogočati nadaljnje elektronsko poslovanje,
 - omogočen mora biti dostop in vpogled v baze podatkov, programsko ter strojno opremo, ki se uporablja pri vodenju elektronskih evidenc,
 - dostop do podatkov mora biti zagotovljen na elektronskih medijih,
 - zagotovljeno mora biti varovanje zaupnosti in celovitosti podatkov,
 - zavezanec mora zagotoviti avtentičnost izdanih dokumentov izdajatelja in
 - iz dokumentacije mora biti razviden opis elektronskega sistema za vodenje poslovnih knjig;
- elektronski prenos podatkov je definiran v 42. členu, ki dovoli njegovo uporabo tudi med subjekti, če ni z zakonom drugače določeno.

Slovenski računovodski standardi opredeljujejo knjigovodske listine in poslovne knjige. Na elektronsko poslovanje in e-račune se navezujejo naslednji standardi (Slovenski računovodski standardi 2006, 2007):

- standard 21.1 podrobneje opredeljuje knjigovodske listine, ki so lahko tudi v elektronski obliki;
- v standardu 21.14 je opredeljeno, da mora podjetje v svojem splošnem aktu opredeliti način hrambe knjigovodskih listin, ki so lahko prejete v papirni ali elektronski obliki;
- v primeru elektronske poslovne knjige se pooblaščen oseba ob njihovi zaključitvi podpiše z elektronskim podpisom, kar definira standard 9.17.

2.4.3 Izmenjava elektronskih računov s proračunskimi uporabniki

Odločilne spremembe pri uporabi e-računa in z njim povezanim elektronskim poslovanjem proračunskih uporabnikov je povzročil Zakon o spremembah in dopolnitvah zakona o opravljanju plačilnih storitev za proračunske uporabnike, ki od 1. januarja 2015 za vse proračunske uporabnike predpisuje obvezno uporabo e-računov in njihove spremljajoče dokumentacije.

UJP je v skladu s 26. členom Zakona o opravljanju plačilnih storitev za proračunske uporabnike postal enotna vstopna in izstopna točka za izmenjavo računov ter vse spremljajoče dokumentacije v elektronski obliki, ki jih izdajajo in prejema proračunski uporabniki. Slednji morajo e-račun prejemati in izdajati izključno prek UJP (Uprava Republike Slovenije za javna plačila, b.1.).

Kot edina vhodna in izhodna točke za e-račune proračunskih uporabnikov UJP opravlja naslednje naloge (Pilih, 2011, str. 13–16):

- zagotavlja postopke vključitve za izmenjavo e-računov, ki obsegajo oddajo vlog in prevzem registra prejemnikov e-računov;
- izdajateljem in prejemnikom e-računov prek portala Beta UJPnet omogoča testiranje;
- prek portala UJPnet in spletnega servisa B2B omogoča produkcijsko izmenjavo e-računov;
- na podlagi podpisane vloge izdajatelja e-računa vključi v evidenco izdajateljev e-računov;
- zagotavlja izmenjavo e-računov in druge spremne dokumentacije računov;
- za proračunske uporabnike vodi register prejemnikov e-računov;
- prejete e-račune posreduje prejemnikom – proračunskim uporabnikom;
- izdajateljem, ki so proračunski uporabniki, posreduje vse povratne informacije, vezane na izmenjavo računov;
- zavrne e-račun v primeru, če račun ali priloge niso skladne z veljavno predpisano obliko, s strukturo ovojnice ali z velikostjo elektronskega računa, ki je 2 MB.

Pravne in fizične osebe, ki so izdajatelji e-računov, lahko te pošljejo proračunskim uporabnikom na naslednje načine (Uprava Republike Slovenije za javna plačila, b.1.):

- prek ponudnikov elektronskih poti in bank, s katerimi ima UJP sklenjene pogodbe o izmenjavi e-računov;
- prek portala UJP eRačun, ki podpira pošiljanje e-računov za manjše izdajatelje, ki proračunskim uporabnikom posredujejo mesečno do pet oziroma letno do šestdeset e-računov;
- prek vmesnika B2B, pri katerem je izmenjava podatkov med UJPnet in informacijskim sistemom avtomatizirana.

Zakoni, pravilniki in direktive, vezani na uporabo e-računov pri proračunskih uporabnikih, so opisani v nadaljevanju.

Zakon o opravljanju plačilnih storitev za proračunske uporabnike. V 14. točki 4. člena zakon kot edino enotno vstopno in izstopno točko, prek katere proračunski uporabniki pošiljajo in prejemajo račune ter spremljajoče dokumente, definira UJP. V 26. členu zakona je podrobneje opisana izmenjava dokumentov, ki morajo biti izključno v elektronski obliki. Člen zavezuje tudi vse pravne in fizične osebe, ki proračunskim uporabnikom izdajajo račune za opravljeno dobavo blaga ali storitev, da jih izdajo v elektronski obliki in vložijo izključno prek spletnega programa UJPnet (Ur.l. RS, št. 59/2010, 111/2013).

Pravilnik o standardih in pogojih izmenjave elektronskih računov prek enotne vstopne in izstopne točke pri Upravi Republike Slovenije za javna plačila. V

pravilniku so določeni standardi, oblika in pogoji za izmenjavo e-računov. Pomembnejša člena sta naslednja (Ur.l. RS, št. 9/2011, 109/2012, 75/2014, 94/2014, 75/2015, 9/2016):

- v 3. členu je določeno, da morajo e-računi po vsebini in obliki ustrezati vsem pogojem ter zahtevam, določenim z zakonom, ki ureja davek na dodano vrednost in vsem predpisom, izdanim na njegovi podlagi. Predpisana oblika za e-račun je e-SLOG. Ko proračunski uporabnik izdaja račun fizični osebi mora poleg njega obvezno priložiti vizualizacijo računa, ki je lahko v PDF ali TIFF obliki;
- tretje poglavje opredeljuje administrativne pogoje izmenjave e-računov. V 5. členu so določeni obvezni elementi ovojnice, ki mora vsebovati podatke za usmerjanje, podatke o izvoru dokumenta, podatke za plačilo in podatke o prilogah e-računa.

Pogoji spletnega poslovanja med proračunskimi uporabniki in upravo Republike Slovenije za javna plačila. UJP kot enotna vstopna in izstopna točka s splošnimi pogoji določa pogoje posredovanja izdanih računov ter spremljajočih dokumentov v elektronski obliki, ki se lahko izmenjujejo prek UJPnet ali UJPnet B2B. Pomembnejši so naslednji členi (Uprava Republike Slovenije za javna plačila, 2012):

- v 5. členu so podrobneje definirani tehnični pogoji, ki so potrebni za dostop do UJPneta;
- 10. člen ureja elektronsko podpisovanje;
- 16. člen ureja administrativne postopke, ki morajo biti zagotovljeni pred posredovanjem in prejemanjem e-računov prek UJPneta ali UJPneta B2B.

Direktiva 2014/24/EU o javnem naročanju in razveljavitvi Direktive 2004/18/ES ureja izvajanje javnega naročanja v elektronski obliki s ciljem njegove poenostavitve, prilagodljivosti in transparentnosti. Elektronsko naročanje v EU postaja obvezno, zato direktiva predpisuje naslednje (Ur.l. EU, št. OJ L 94):

- razpisi in razpisna dokumentacija morajo biti do marca 2016 obvezno objavljeni v elektronski obliki;
- elektronska oddaja ponudb za centralno naročanje v javni upravi postaja obvezna do marca 2017;
- do septembra 2018 postaja obvezna elektronska oddaja ponudb za celotno javno upravo.

Direktiva 2014/55/EU o izdajanju elektronskih računov pri javnem naročanju. Z direktivo je urejeno področje elektronskega naročanja v javnem naročanju. V 6. členu opredeljuje elemente e-računa. V 7. členu javnim naročnikom narekuje prejem in obdelavo e-računov, ki morajo biti skladni z evropskim standardom za izdajanje e-računov (Ur.l. EU, št. OJ L 133).

Pojasnilo Ministrstva RS za finance, Direktorata za zakladništvo in Združenja bank Slovenije v povezavi z obveznim izdajanjem e-računov. S strani Združenja bank Slovenije je bila podana zahteva za upoštevanje izjave glede izdaje e-računov pri poslovanju s proračunskimi uporabniki, vendar je bila s pojasnilom zavrnjena, saj zakon ZOPSPU-A v 26. členu ne predvideva izjem (Direktorat za zakladništvo, 2014).

2.4.4 Arhiviranje elektronskih dokumentov

Po nastanku in uporabi je treba vso poslovno dokumentacijo ustrezno hraniti, dokler zanje ne potečejo zakonsko predpisani roki. S hrambo mora biti zagotovljeno ohranjanje njene vsebine, trajnosti, celovitosti in dostopnosti. Berčič (2004, str. 4) ugotavlja, da je trajno elektronsko arhiviranje, ki je nadgradnja pravnega priznanja veljavnosti elektronskih dokumentov, ključnega pomena za elektronsko poslovanje. V nadaljevanju so navedeni predpisi, povezani z arhiviranjem elektronskih dokumentov.

Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih. Pomembnejši členi zakona so naslednji (Ur.l. RS, št. 30/2006, 51/2014):

- v 2. členu zakon določa razliko med dokumentarnim in arhivskim gradivom. Pod dokumentarno gradivo spadajo vse vrste in oblike zapisov, ki so nastale ali bile prejete pri poslovanju. Z arhivskim gradivom pa se opredeljuje dokumentarno gradivo, ki predstavlja trajen pomen za zgodovino, druge znanosti in kulturo ali trajen pomen za pravni interes pravnih ter fizičnih oseb;
- s 3., 4., 5. in 6. členom je podrobneje opredeljena hramba dokumentarnega gradiva, s katero mora biti zagotovljena trajnost, nespremenljivost, integralnost, urejenost ter dokazljivost izvora gradiva;
- z 11. in 12. členom je določena oblika za dolgoročno hrambo gradiva. Če je za gradivo predpisano trajanje hrambe daljše od petih let je treba gradivo pretvoriti v digitalno obliko za dolgoročno hrambo;
- enakost v digitalni obliki varno hranjenega gradiva je enaka izvirnemu gradivu, kar je definirano v 31. členu;
- v 6. poglavju zakon ureja zagotavljanje ustrezne infrastrukture in storitev.

Uredba o varovanju dokumentarnega in arhivskega gradiva določa naslednje: ureja delovanje ter notranja pravila oseb, ki hranijo dokumentarno oziroma arhivsko gradivo, hrambo tega gradiva v fizični in digitalni obliki. Dodatno ureja tudi splošne pogoje, registracijo in akreditacijo opreme ter storitev za digitalno hrambo, odbiranje in izročanje arhivskega gradiva javnim arhivom, strokovno obdelavo ter vodenje evidenc arhivskega gradiva, varstvo filmskega in zasebnega arhivskega gradiva, uporabo arhivskega gradiva v arhivih ter delo arhivske komisije. Pomembnejši členi so naslednji (Ur.l. RS, št. 86/2006):

- v 11. členu je določen postopek zajemanja izvirnega dokumentarnega gradiva v digitalni obliki, ki mora omogočati:
 - evidentiranje in upravljanje vseh enot dokumentarnega gradiva ne glede na obliko,
 - pravilen zajem vsebine in primerno klasifikacijo tako zajetega dokumentarnega gradiva,
 - samodejno in ročno kontrolo pravilnosti zajema vsebinskih ter metapodatkov in hrambo ustrezne dokumentacije, s katero se zagotavlja ustreznost metod zajema,
 - varnost in nespremenljivost zajetega dokumentarnega gradiva po pravilnem zajemu ter
 - možnost zajema in popravke napak s strani pooblaščenih oseb ob upravičenih ter dovoljenih popravkih podatkov;
- varna dolgoročna hramba v digitalni obliki zajetega gradiva je opredeljena v 16. členu;
- zagotavljanje avtentičnosti in celovitosti hranjenega gradiva je definirano v 17. in 18. členu.

Enotne tehnološke zahteve. Arhiv RS je skladno s predpisi 91. člena zakona ZVDAGA sprejel enotne tehnološke zahteve, ki podrobneje opredeljujejo poslovne, organizacijske in tehnološke pogoje za izpolnjevanje ZVDAGA. Gre za povezovalni element med zakonskimi zahtevami, ki izhajajo iz temeljnih načel zagotavljanja varne hrambe in spreminjajočimi se potrebami prakse (Arhiv Republike Slovenije, 2013).

3 ANALIZA INFORMACIJSKE REŠITVE

3.1 Predstavitev Izbranega podjetja in situacije

Izbrano podjetje je javni zdravstveni zavod in opravlja bolnišnično ter ambulantno specialistično dejavnost na sekundarni in terciarni ravni. Podjetje je posredni proračunski uporabnik, in sicer se uvršča v skupino javnih zavodov ali drugih izvajalcev javnih služb.

Organizacijsko se podjetje deli na dve veliki skupini. Prvo skupino, ki opravlja zdravstveno dejavnost, predstavljajo klinike, službe in oddelki, medtem ko v drugo skupino spadajo upravno-tehnične službe. Med slednje spadajo kadrovsko-pravna služba, nabavna služba, finančno-računovodska služba, ekonomsko-analitska služba, služba za oskrbo in vzdrževanje ter služba za investicije. Te službe pri svojem delu uporabljajo tudi programsko rešitev Finančna aplikacija, in sicer je njenih uporabnikov približno dvesto petdeset. Največ uporabnikov aplikacije je v nabavni službi, ki jo sestavljajo oddelek nabave materiala in storitev, oddelek nabave opreme in oddelek za javna naročila ter v finančno-računovodski službi, ki jo sestavljata dva oddelka, in sicer finančni ter računovodski oddelek (Izbrano podjetje, 2014).

Izbrano podjetje je dolgoletni uporabnik Finančne aplikacije, razvite s strani Podjetja za računalniški inženiring. Finančna aplikacija predstavlja prilagodljivo programsko rešitev za področje računovodstva, financ, analiz in kontrolinga. Sistem je povezljiv, razširljiv, prilagodljiv in pomaga organizaciji k hitrejši ter ustreznejši adaptaciji na vsakodnevne spremembe, ki izhajajo iz notranjega in zunanjega okolja (Podjetje za računalniški inženiring, 2008).

Finančno aplikacijo sestavljajo posamezni moduli, ki so med seboj povezani na način, da se podatek v sistem vnese samo enkrat, podatki pa se med moduli prenašajo glede na delovni proces, ki se odvija v organizaciji. Modularna zasnova omogoča večjo prilagodljivost delovnega okolja, saj lahko uporabniki upravljajo tudi le s posameznim modulom, odvisno od delovnih nalog, ki jih morajo opraviti in za katere so zadolženi. Ti moduli so narejeni glede na posamezno funkcijsko področje, ki se odvija v organizaciji in so naslednji: prispeli dokumenti, finance, DDV, materialno poslovanje, fakturiranje, osnovna sredstva, potni nalogi, komerciala, plače in skupni objekti (Podjetje za računalniški inženiring, 2008).

Izbrano podjetje je posredni proračunski uporabnik, zato je moralo svoje poslovanje prilagoditi ZOPSPU-A, ki je s 1. januarjem 2015 nalagal obvezno uporabo prejetih in izdanih e-računov ter njihove spremljajoče dokumentacije. Podjetje v Finančni aplikaciji med drugim evidentira prejete in izdane račune, zato so se odločili za njeno nadgradnjo, ki bo omogočala delo z e-računi.

Izbrano podjetje na strani izdanih računov ni pričakovalo težav, povezanih z uvedbo zakonsko predpisanih e-računov. Za izdajo računov uporabljajo modul fakturiranje, zato je bilo treba Finančno aplikacijo dopolniti tako, da se bo ob potrditvi računa dodatno kreiral še e-račun in zapisal v aplikacijo, ki bo ta račun posredovala na UJP. Delovni proces uporabnika pri izdanih računih je z obvezno uporabo e-računov ostal praktično nespremenjen.

Večjo težavo so podjetju predstavljali prejeti računi, saj so za njih uporabljali dobro utečen delovni tok, vezan na ogromne količine fizično izmenjane dokumentacije, v kateri je udeleženih približno sto uporabnikov, ki mesečno obdelajo približno tri tisoč računov. Podatke, potrebne za knjigovodsko likvidaturo računov, se vnaša na fizične dokumente – račune, ki se jih nato vnese v Finančno aplikacijo, in sicer v modul prispeli dokumenti. Ko račun fizično prispe do naslednje osebe v delovnem toku dokumenta, ga ta spet poišče v modulu prispeli dokumenti in vnese podatke, za katere je zadolžena, tako v aplikacijo kot tudi dodatno na fizični račun ter račun preda naslednji osebi v delovnem toku. Pred samo analizo informacijske rešitve so bili opravljeni razgovori z zaposlenimi, ki so vključeni v likvidaturo računov ter vodjo informatike. Na podlagi zbranih dejstev je bil izdelan natančen popis obstoječega delovnega toka Izbranega podjetja za obdelavo prejetih računov, ki je naslednji:

- Račun prispe v finančno računovodsko službo:
 - račun se opremi z datumom prihoda in dofo. V modul prejeti dokumenti se vnesejo osnovni podatki (dobavitelj, dofa, vrednost, datum DDV, datum obdobja, vrsta stroška, šifra posla, namen, sklic, STM, konto, številko originalnega dokumenta itd.);
 - izpolni se podatke o davku na dodano vrednost;
 - račun se odpremi v ustrezni nabavni oddelek.

- Račun prispe v nabavno službo:
 - referentka na račun fizično zabeleži datum prihoda in referenta, ki bo izvedel likvidacijo ter nato te podatke vnese v modul prispeli dokumenti;
 - račun se izroči referentu, ki je izdal naročilnico;
 - če gre za material, referent poišče pripadajočo prejemnico in naročilnico ter opravi kontrolo vrednosti na računu;
 - če gre za storitev, referent pošlje račun v ustrezno službo kjer potrdijo, da je bila storitev v resnici izvedena skladno z naročilom oziroma pogodbo. Odgovorna oseba opravljeno storitev potrdi s podpisom na zadnji strani računa. Račun se vrne referentu;
 - referent vrednost na računu primerja z vrednostjo iz naročilnice oziroma pogodbe;
 - referent pravilnost računa potrdi s svojim podpisom na zadnji strani računa in ga preda v kontrolo vodji oddelka;
 - vodja oddelka s podpisom na zadnji strani računa potrdi pravilnost računa in ga preda referentki, ki je odgovorna za vodenje evidence prispelih ter oddanih računov;
 - referentka vpiše datum prenosa računa v oddelek finančno-računovodske službe in ga isti dan tja tudi fizično odnese.

- Račun ponovno prispe v finančno-računovodsko službo h knjigovodjem:
 - v oddelku plačilnega prometa se vpiše datum valute, spremeni se vrsta stroška (iz na primer 01 v 11, kar pomeni, da račun ni več v oddelku nabavne službe);
 - glede na oddelek (stroškovno mesto) se računi razporedijo v knjigovodstvo osnovnih sredstev, materialno knjigovodstvo in saldakonte, kjer v modulu prispeli dokumenti izpolnijo postavke računa ter izvedejo prenos dokumenta v modul finance;
 - potrjen račun se vrne v oddelek plačilnega prometa, kjer ostane dokler ni plačan, nato pa se vrne v knjigovodstvo.

Za potrebe dela z e-računi sta bili izdelani analizi za dve novi informacijski rešitvi, in sicer za aplikacijo eRačun, ki se bo uporabljala za prejem in pošiljanje e-računov, ter aplikacijo ePoslovanje, katere namen bo knjigovodska likvidatura prejetih e-računov na podlagi delovnega toka dokumenta, uporabljenega v Izbranem podjetju.

3.2 Metodologija razvoja v Podjetju za računalniški inženiring

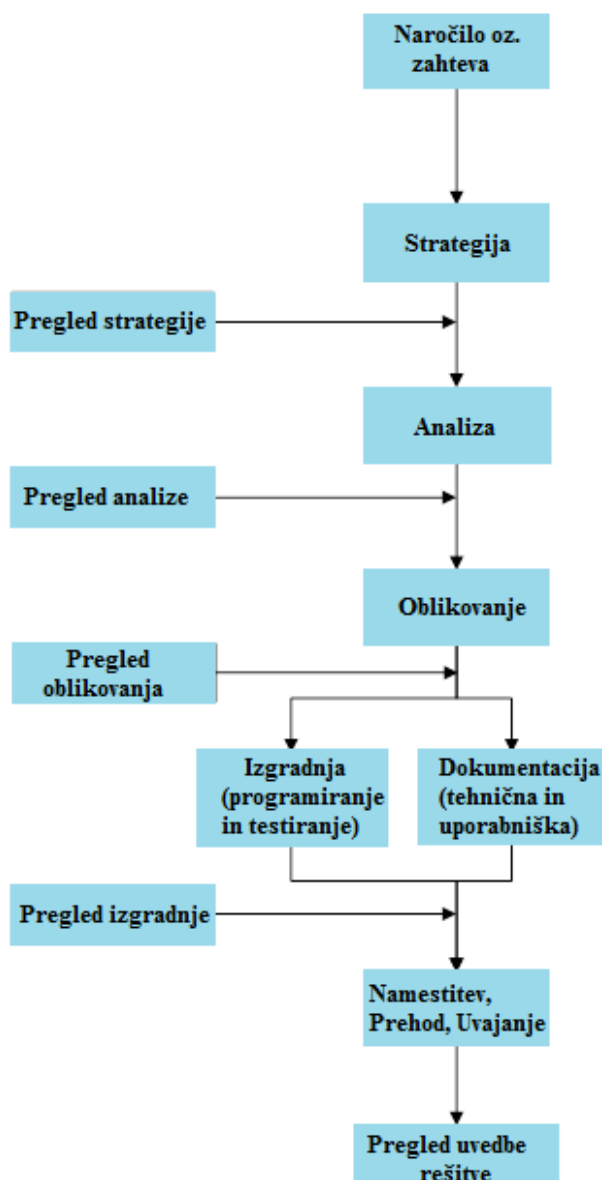
Metodologija razvoja aplikacij, ki je predpisana v Podjetju za računalniški inženiring, je sicer osnovana na metodologiji Oracle CASE, vendar nanjo ni omejena in se lahko po potrebi prilagodi zahtevam ter kompleksnosti posameznega projekta ali aplikacije. Za morebitno prilagoditev metodologije je pristojen projektant oziroma arhitekt na projektu, z odobritvijo vodje projekta. Bistvene razlike v pristopu, predvsem sprememba pri obveznih točkah v predpisani metodologiji, morajo biti preverjene in odobrene tudi s strani vodje kakovosti ter dokumentirane, bodisi v internem navodilu za člane projektnega tima, bodisi v vzpostavitev dokumentu projekta. Opustitev neobveznih točk posameznih faz ni treba posebej dokumentirati. Obseg dela v posamezni fazi je v veliki meri odvisen od tega, kako natančno so naročnikove zahteve že dokumentirane. Včasih je treba najprej izdelati celoten posnetek stanja, včasih pa so specifikacije zahtev naročnika lahko že tako natančne, da zadostujejo kriterijem faze analize (Podjetje za računalniški inženiring, 2014).

Shema osnovnega procesa razvoja izdelkov v Podjetju za računalniški inženiring je prikazana na Sliki 14. Poleg osnovnih postopkov in metod se lahko v postopku projekta izvajajo tudi dodatne aktivnosti, ki so odvisne od zahtev projekta ter dogovora z naročnikom, kot npr. izobraževanje končnih uporabnikov (Podjetje za računalniški inženiring, 2014).

Analiza je temeljni načrt za izgradnjo oziroma dopolnitev programske rešitve, zato mora, skladno z razpoložljivimi sredstvi projekta, dovolj natančno definirati potrebne postopke in prikaze te rešitve, da bi pri uporabi čim boljje ustrezala potrebam naročnika. Ustreznost, nedvoumnost in razumljivost dokumenta mora biti preverjena ter usklajena s strani naročnika in izvajalca, saj se le tako lahko izognemo nesporazumom pri izdelavi programske rešitve, ki lahko bistveno poslabšajo ustreznost rešitve, podaljšajo čas izvedbe in povečajo stroške njenega razvoja (Podjetje za računalniški inženiring, 2010).

Namen dokumenta analize je, da natančno opiše poslovne procese in funkcionalnosti, ki jih mora pokrivati aplikacija. Poslovni procesi morajo biti nedvoumno definirani in jasno opisani. Analiza nastane predvsem na osnovi specifikacij naročnika, na osnovi posnetka stanja ali zakonskih določil (Podjetje za računalniški inženiring, 2011).

Slika 14: Shema procesa razvoja v Podjetju za računalniški inženiring



Vir: Podjetje za računalniški inženiring, Standardni predpis št. 01, 2014.

Metodologija Oracle CASE, ki je predpisana v Podjetju za računalniški inženiring, predstavlja množico pravil, navodil in več tehnik navzkrižnega preverjanja, s katerimi se v posameznih kontrolnih točkah razvoja preveri zahtevane rezultate ter izpolnjevanje pogojev, ki omogočajo začetek naslednje faze v življenjskem ciklu razvoja. V osnovi metoda predstavlja klasični linearni oziroma slapovni model razvoja, pri katerem si posamezne faze sledijo zaporedno in med njimi ni prekrivanja. Eden izmed vzrokov za uporabo tega razvojnega modela v Podjetju za računalniški inženiring je sodelovanje in razvoj programskih rešitev za proračunske uporabnike, ki takšen razvojni model predpisujejo oziroma zahtevajo.

V primeru razvoja programskih rešitev za Izbrano podjetje je bila izvedena prilagoditev predpisane metodologije, saj klasični slapovni model v dani situaciji ni predstavljal optimalne izbire. Za razvoj je bil uporabljen agilnejši pristop, ki je temeljil na metodologiji poenotenega procesa (angl. *Unified Process*). Gre za objektno usmerjeno metodologijo, ki temelji na inkrementalnem in iterativnem razvoju, pri katerem se celoten projekt razdeli na več manjših projektov, ki se na koncu povežejo v celoto. Na prilagoditev agilnejši metodologiji je vplivalo več dejavnikov, med drugimi tudi zelo kratek razvojni čas in slabo definirane začetne zahteve naročnika – predvsem glede obstoječega delovnega toka likvidature prejetih računov ter posebnosti pri njem. Razvoj aplikacij je tako potekal na podlagi orodij Oracle CASE (Oracle Designer), s katerim so bili izdelani entitetni in podatkovni modeli, ki so bili nato uporabljeni v posameznih iteracijah. Za večje funkcionalnosti so bile izvedene posamezne iteracije, v katerih je bil uporabljen klasični slapovni razvojni model, kar je omogočalo postopen razvoj aplikacij, ki so bile kot takšne naročniku tudi predane v testno uporabo. Za kontrolne elemente posameznih iteracij so bile uporabljene zahteve in kazalniki, ki jih po posameznih fazah življenjskega cikla razvoja predpisuje metodologija Oracle CASE, kar je kljub uporabi agilnega pristopa omogočalo dober nadzor nad projektom in tveganji.

3.3 Analiza aplikacije eRačun

Za namene obdelave in procesiranja e-računov bo izdelana nova samostojna aplikacija za prevzem, izdajo, obdelavo, vizualizacijo in prenose e-računov, ki bo komunicirala z drugimi moduli Finančne aplikacije Podjetja za računalniški inženiring in UJP B2B-vmesnikom za izmenjavo podatkov.

Aplikacija bo namizna aplikacija, narejena na ogrodju Microsoft.NET Framework 4.0, uporabljala pa bo podatke Finančne aplikacije, ki so shranjeni v bazi Oracle 12c. Do podatkov bo aplikacija dostopala prek ogrodja Entity Framework, ki je Microsoftov objektno relacijski vmesnik za preslikavo med podatki v bazi in objektnimi modeli v aplikaciji.

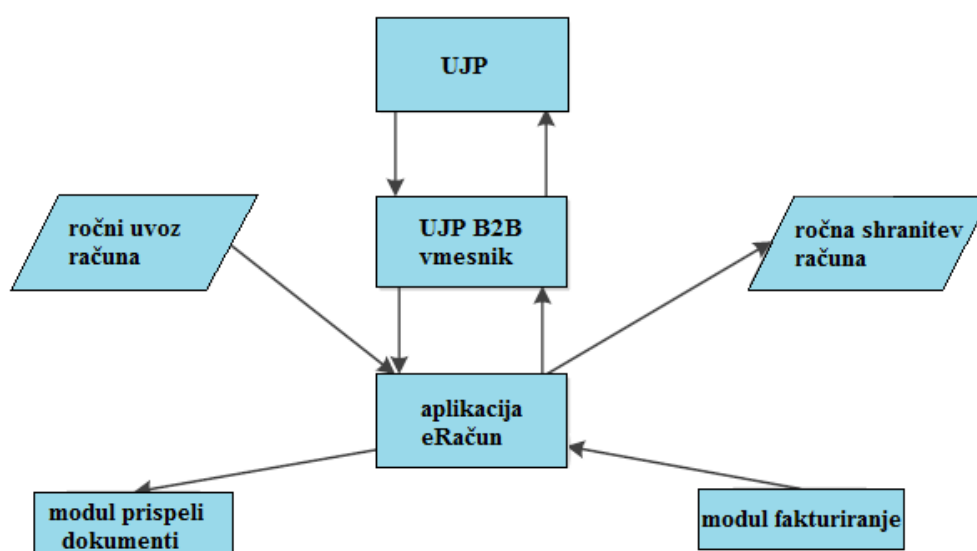
3.3.1 Shema povezav nove aplikacije

Okvirna shema povezav nove aplikacije eRačun je prikazana na Sliki 15 in obsega naslednje:

- UJP predstavlja enotno vstopno in izstopno točko za e-račune proračunskih uporabnikov;
- vmesnik B2B predstavlja povezavo z UJP, prek katere bo potekala izmenjava e-računov. Omogočal bo samodejno izmenjavo podatkov med UJPnetom in informacijskim sistemom proračunskega uporabnika;

- nova aplikacija eRačun bo prejete e-račune shranila v vmesne tabele, iz katerih bodo nato posredovani naprej v aplikacijo za obdelavo prispelih dokumentov, kjer bo izvedena njihova knjigovodska likvidatura. Nova aplikacija bo omogočala tudi prevzem izdanih računov iz vmesnih tabel Finančne aplikacije, natančneje modula fakturiranja. Nova aplikacija bo omogočala pregled, izpise, podpisovanje računov z digitalnim potrdilom in dodajanje prilog računom pred njihovim pošiljanjem;
- ročni uvoz računa in ročna shranitev računa bosta omogočala poslovanje brez vmesnika B2B, in sicer v primeru njegovega nedelovanja ali opsijske izmenjave e-računov z bankami.

Slika 15: Shema povezav nove aplikacije



Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije eRačun, 2015b.

3.3.2 Funkcionalnosti nove aplikacije eRačun

Uporaba aplikacije eRačun bo omogočala naslednje funkcionalnosti:

- Prevzem e-računov v ovojnici prek časovne zanke – B2B ali ročno, na željo uporabnika. Način uvoza bo viden prek dodatnega atributa, v katerem bosta lahko prikazani dve vrednosti: vrednost »B2B« za račune, ki so bili prevzeti prek vmesnika B2B ali vrednost »vhod« za račune, ki so bili ročno uvoženi v aplikacijo.
- Odpiranje ovojnice e-računov ter shranjevanje računa in prilog.
- Kreiranje ovojnic za izdane račune, v katero bo zajet račun in poljubne priponke oziroma spremljajoča dokumentacija. Ovojnica se bo kreirala za posamezni račun. BIC koda na ovojnici bo vezana na parameter (nastavitev) v šifrantih Finančne aplikacije.
- Omogočeno bo elektronsko podpisovanje računa z digitalnim potrdilom izdajatelja.

- Pregled prevzetih in izdanih računov prek namenskih preglednih mask, izpis PDF slike računa ali prikaz vizualiziranega računa (za e-račune brez PDF slike računa).
- Omogočen bo pregled in urejanje priponk, kar bo odvisno od izvora e-računa ter od trenutnega statusa, v katerem se bo e-račun nahajal. Izdanim računom se bo lahko pred kreiranjem ovojnice dodajalo poljubne priponke.
- Pošiljanje e-računov v aplikacijo za obdelavo prispelih dokumentov.
- Omogočeno bo vračanje računov iz aplikacije za obdelavo prispelih dokumentov (npr. napačni prevzem), ki bo spremenilo status e-računa tako, da bo ta pripravljen na ponovni prevzem.
- Beleženje revizijske sledi v tabelah za beleženje sprememb.

Za krmiljenje prehodov stanj, aktivnosti ali neaktivnosti posameznih funkcionalnosti maske bo uporabljen atribut status računa. Statusi e-računa bodo različni za vsako posamezno vrsto računa (prejeti ali izdani) in bodo vplivali na delo z njim.

Statusi za prejete račune bodo naslednji:

- **Prejet:** status bo e-račun dobil takoj ob uvozu oziroma prevzemu v aplikacijo.
- **Prevzet:** e-račun bo dobil ta status takoj, ko bo iz aplikacije eRačun poslan naprej v aplikacijo za obdelavo prispelih dokumentov in tam uspešno prevzet (shranjen).

Statusi za izdane račune bodo naslednji:

- **Pripravljen:** račun bo dobil ta status takoj, ko bo potrjen na maski za vnos izdanih računov v modulu Fakturiranje, ki je del Finančne aplikacije. Potrditev računa v modulu Fakturiranje pomeni hkrati tudi prenos računa v tabele nove aplikacije eRačun, kar bo veljajo le za poslovne partnerje, ki bodo imeli v šifrantu z ustreznim atributom označen prejem e-računa.
- **Podpisan:** status bo pomenil, da je bil izbrani račun podpisan z digitalnim certifikatom in je pripravljen na pošiljanje.
- **Poslan:** račun bo pridobil ta status takoj, ko bo uspešno posredovan na UJP ali banko.
- **Datoteka:** račun bo pridobil ta status, ko bo za njega ročno kreirana datoteka (samo pri uporabnikih standardne izmenjave s spletnimi bankami).
- **Zavrnil:** status bo označeval račune, ki bodo poslani, vendar kasneje nedostavljeni.
- **Izločen:** status se bo lahko določil računom, ki bodo nepravilni, napačni ali nepravilno posredovani v novo aplikacijo eRačun.

3.3.3 Opis mask, polj in gumbov

Izdelana bo nova maska za pregled in delo z e-računi. Videz maske, ki je prikazana na Sliki 16, bo podoben kot so ostale maske v Finančni aplikaciji, kar pomeni, da bo maska

vsebinsko razdeljena na tri dele. V zgornjem delu bodo vneseni parametri in iskalni filtri, osrednji del maske bo namenjen vrstičnemu pregledu računov glede na izbrane filtre, gumbi pa bodo postavljeni na dnu maske. Dodatno filtriranje računov bo možno še v osrednjem delu maske, in sicer naraščajoče ali padajoče po vseh prikazanih poljih ter s postavitvijo filtra samo po določenem polju – uporabnik bo v tem primeru vnesel iskano vrednost polja.

Slika 16: Maska za pregled e-računov

Partner	Št. računa	OE	Status	Št. prilog	Dat. računa	Dat. zapadlosti	Naslov računa	Znesek	Valuta	Opombe
	20140010		PREVZET	2	29.10.2014	06.11.2014	Račun številka: 20140010 z dne: 29.10.20...	85,40	EUR	test vmitve in
	122	0	PREVZET	3	09.09.2014	10.09.2014		12,20	EUR	
	123	0	PREJET	2	09.09.2014	10.09.2014		12,20	EUR	
	20140013		PREJET	2	29.10.2014	06.11.2014	Račun številka: 20140013 z dne: 29.10.20...	85,40	EUR	podvojena da
	20140011		PREJET	1	29.10.2014	06.11.2014	Račun številka: 20140011 z dne: 29.10.20...	85,40	EUR	podvojena da

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije eRačun, 2015b.

Zgornji del nove maske bo vseboval naslednja polja:

- Trenutni datum: gre za prikaz trenutnega (sistemskega) datuma.
- Prijavljen uporabnik: uporabniško ime trenutno prijavljenega uporabnika.
- Vrsta računov: spustni seznam, s katerim se bo lahko izbralo prikaz prejetih, izdanih ali vseh e-računov.
- Status računov: spustni seznam, s katerim se bo izbralo status e-računov, ki se bodo prikazali v osrednjem delu.
- Digitalno potrdilo: spustni seznam, s katerim se bo izbralo digitalno potrdilo, ki se bo uporabljajo pri podpisovanju izdanih e-računov.
- Geslo: prikazano bo geslo izbranega digitalnega potrdila.

V osrednjem delu maske bodo vrstično prikazani izbrani računi. Polja v tem delu maske bodo naslednja: polje za izbor (označitev s kljukico), partner (v primeru prejetega e-računa bo prikazan izdajatelj, v primeru izdanega e-računa bo prikazan naslovnik), številka računa, organizacijska enota, število prilog, datum računa, datum zapadlosti, naslov računa, znesek, valuta in opomba. Vrednosti v poljih številka računa in število prilog bosta

hiperpovezavi. Izbor številke računa bo pokazal PDF sliko računa, izbor vrednosti iz polja število prilog pa bo odprl novo masko za pregled in urejanje prilog.

V spodnjem delu maske bodo naslednji gumbi:

- **Gumb za osvežitev:** namenjen bo osvežitvi pregleda računov, ki se prikazujejo v osrednjem delu.
- **Podpiši:** gumb bo namenjen podpisu označenih e-računov z izbranim digitalnim potrdilom, ki bo izbran v zgornjem delu maske. Gumb bo aktiven za vse izdane račune s statusom pripravljen.
- **Označi za pošiljanje:** namenjen bo pošiljanju vseh izdanih e-računov, ki bodo v statusu podpisan.
- **Vrni:** uporaba gumba bo izbranemu prejetemu e-računu spremenila status iz prevzet nazaj na status prejet. Zaradi revizijske sledi bo v tem primeru obvezen vnos opombe.
- **Pošlji:** gumb bo namenjen takojšnjemu pošiljanju e-računov, ki so označeni za pošiljanje.
- **Prevzemi:** pri vključeni izmenjavi prek vmesnika B2B bodo na ročno zahtevo prevzeti vsi računi z UJPneta.
- **Shrani v datoteko:** gumb bo izbrani račun ali več izbranih računov shranil v datoteko in jo zapisal na izbrano področje, ki bo določeno v nastavitveni datoteki aplikacije. Pri shranitvi bo za vsak posamezni račun kreirana ovojnica, XML račun in vse priloge, ki so bile dodane e-računu.

3.3.4 Postopek dela s prejetimi računi

Uporabniku se bodo po uspešni prijavi v aplikacijo eRačun v osrednjem delu maske prikazali vsi prejeti in izdani e-računi, ki bodo urejeni padajoče po datumu dokumenta, kar pomeni, da se bodo najnovejši računi prikazovali na vrhu maske. Uporabnik bo lahko v zgornjem delu maske v polju vrsta računa določil samo prikaz prejetih računov.

Ob nastavljenem samodejnem časovniku za prevzem prek vmesnika B2B se bodo računi samodejno prevzemali v tabele nove aplikacije. Če uporabnik ne bo imel nastavljenega samodejnega časovnika za prevzem računov prek vmesnika B2B ali pa bo račune želel prevzeti neodvisno od časovnika, bo račune z UJPneta lahko ročno prevzel z gumbom **prevzemi**.

Z gumbom **vrni** bo uporabnik lahko spremenil status računu, ki je bil že prevzet v aplikacijo za delo s prejetimi dokumenti, nazaj v status prejet in s tem omogočil ponovni prevzem tega računa. Za uspešno shranitev statusa bo treba vnesti opombo, ki se bo nato prikazovala v osrednjem delu maske, v namenskem polju, in bo zagotavljala dodatno revizijsko sled.

3.3.5 Postopek dela z izdanimi računi

Uporabnik bo lahko v zgornjem delu maske v polju vrsta računa določil samo prikaz izdanih računov in te dodatno filtriral tudi v osrednjem delu maske. Urejenost računov bo padajoča, glede na datum dokumenta. Urejenost bo uporabnik lahko dodatno spremenil.

Uporabniki z aktivno povezavo na vmesnik B2B bodo imeli pri izdanih računih aktivne naslednje gumbе:

- Gumb **podpiši** bo aktiven pri vseh računih, ki bodo v statusu pripravljen. Uporabnik jim bo pred podpisom lahko dodal ali brisal priloge ter jih nato podpisal z digitalnim potrdilom, izbranim na vrhu maske. Po podpisu se bo računom status spremenil na podpisan.
- Ko bodo računi podpisani jih bo uporabnik z gumbom **označi za pošiljanje** pripravil za pošiljanje. Avtomatska obdelava B2B bo nato glede na nastavljeno časovno zanko poslala račun v UJPnet. Uporabnik bo lahko račune neodvisno od nastavljene časovne zanke tudi ročno poslal z gumbom **pošlji**. Po uspešnem pošiljanju se bo računom status spremenil na poslan.
- Gumb **shrani v datoteko** bo shranil izbrani račun ali več izbranih računov v datoteko na izbrano področje. Pri shranitvi se bo kreirala ovojnica, XML račun in vse priloge, ki so bile dodane e-računu. Pri shranjevanju v datoteko pri nastavljeni B2B-izmenjavi, se status računu ne bo spremenil in bo ostal poslan. Shranitev računov v datoteko bo uporabnik uporabil v primeru težav z delovanjem vmesnika B2B. Datoteko bo lahko nato ročno uvozil v UJPnet in izdane račune kljub nedelovanju vmesnika B2B uspešno oddal.
- Z gumbom **izloči** bo lahko uporabnik izbrani račun ali več teh izločil iz nadaljnjih obdelav, kar pomeni, da jih ne bo mogoče poslati naprej. Po izločitvi se bo računom spremenil status na izločen – sprememba statusa ne bo več mogoča. Za uspešno shranitev izločitve bo treba vnesti opombo, ki se bo nato prikazovala v osrednjem delu maske in bo služila dodatni revizijski sledi.

Uporabniki standardne izmenjave s spletnimi bankami bodo imeli aktivne druge gumbе:

- Gumb **podpiši** bo imel enako funkcionalnost kot pri načinu prek B2B-izmenjave.
- Računi s statusom podpisan, pripravljen ali datoteka bodo imeli aktiven gumb **shrani v datoteko**. Pri shranitvi v datoteko se bo kreirala ovojnica, XML račun in vse priloge – tako kot pri izmenjavi prek vmesnika B2B. Pri shranjevanju v datoteko pri standardni izmenjavi s spletno banko se bo status računu spremenil v datoteka.
- Gumb **izloči** bo imel isto funkcionalnost kot pri izmenjavi prek vmesnika B2B.

3.3.6 Urejanje prilog računov

Do maske za urejanje prilog se bo dostopalo tako, da se bo na izbranem računu kliknilo na vrednost polja število prilog. Odprla se bo nova maska, prikazana na Sliki 17, na kateri bodo prikazani izbrana številka računa in gumba za dodajanje ter prikaz e-računa. V osrednjem delu bodo prikazane vse uvožene ali ročno dodane priloge, ki se jih bo lahko tudi brisalo. Prilogo se bo izbrisalo tako, da se bo nanjo kliknilo z desnim miškinim gumbom ter nato izbralo ukaz za brisanje. Priloge prejetih e-računov, ki so bile shranjene v ovojnici, ne bo mogoče izbrisati. Prav tako ne bo mogoče izbrisati PDF slike izdanega računa, ki je bil poslan s strani modula fakturiranje, ki je del Finančne aplikacije.

Slika 17: Maska za pregled prilog

Prim?	Ime priloge	Opis priloge	Tip priloge	Datoteka
☒	eRačun		XML	
☐	...		PDF	

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije eRačun, 2015b.

Gumb **prikaz e-računa** bo prikazal PDF sliko računa oziroma njegovo vizualizacijo (če med prilogami ne bo PDF slike). Izbrano prilogo se bo lahko odprlo tudi s klikom na vrednost polja ime priloge.

Z gumbom **dodaj prilogo** se bo odprla nova maska za dodajanje prilog, ki je prikazana na Sliki 18. Uporabnik bo na maski lahko določil ime in opis priloge ter določil področje, s katerega bo prilogo želel dodati. Največja skupna velikost prilog posameznega računa bo omejena na 50 MB. Priloge na prejetih e-računih se bo lahko dodajalo samo v statusu prejet. Priloge na izdanih računih bo možno dodajati samo na računih, ki bodo v statusu pripravljen.

Slika 18: Maska za pregled prilog

The screenshot shows a web application window titled "eRačuni UJP.NET -- Priloge". At the top, there is a text field for "Številka računa" with the value "6910036-14". To the right of this field are two buttons: "Dodaj prilogo" and "Prikaz e-računa". Below these is a section titled "Dodajanje nove priloge". This section contains three text input fields: "Ime priloge", "Opis priloge", and "Izbrana datoteka". To the right of the "Opis priloge" field is a button with a folder icon. At the bottom of the "Dodajanje nove priloge" section are two buttons: "Shrani prilogo" and "Prekliči".

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije eRačun, 2015b.

3.4 Analiza aplikacije ePoslovanje

Aplikacija ePoslovanje bo predstavljala sodobno spletno rešitev za podporo upravljanja delovnega toka knjigovodske likvidature prejetih dokumentov. Rešitev bo prilagodljiva in bo z ustreznimi nastavitvami omogočala tudi podporo drugim delovnim tokovom v organizaciji. Aplikacija ePoslovanje bo imela lastnost samostojne rešitve kot tudi rešitve, ki bo integrirana oziroma povezana s strani naročnika izbranim poslovnim informacijskim sistemom. Ob implementaciji v poslovno okolje se bo aplikacijo ePoslovanje povezalo s poljubno informacijsko rešitvijo in poljubnim arhivskim sistemom naročnika.

Aplikacija bo spletna aplikacija, narejena v Microsoftovi tehnologiji ASP.NET in bo potekala na Microsoftovem IIS-spletnem strežniku (angl. *Internet Information Services*). Zgrajena bo na ogrodju .NET Framework 4.0 in bo uporabljala podatke Finančne aplikacije, shranjene v bazi Oracle 12c. Do podatkov bo dostopala prek NHibernate, ki je odprtokodni objektno-relacijski vmesnik za preslikavo med podatki v bazi in objektnimi modeli v aplikaciji.

V standardni različici bo aplikacija ePoslovanje povezana z informacijskima rešitvama Podjetja za računalniški inženiring, in sicer s Finančno aplikacijo ter aplikacijo eRačun. Analiza rešitve temelji na obstoječem delovnem toku dokumenta oziroma njegovi stopenjski obdelavi, ki je v Izbranem podjetju že uvedena, hkrati pa bo omogočala pregled nad vizualizirano sliko računa in vso spremljajočo dokumentacijo e-računa, ki jo uporabniki potrebujejo za uspešno knjigovodsko likvidaturo prejetega računa.

3.4.1 Funkcionalnosti nove aplikacije ePoslovanje

Nova aplikacija bo uporabnikom omogočala naslednje:

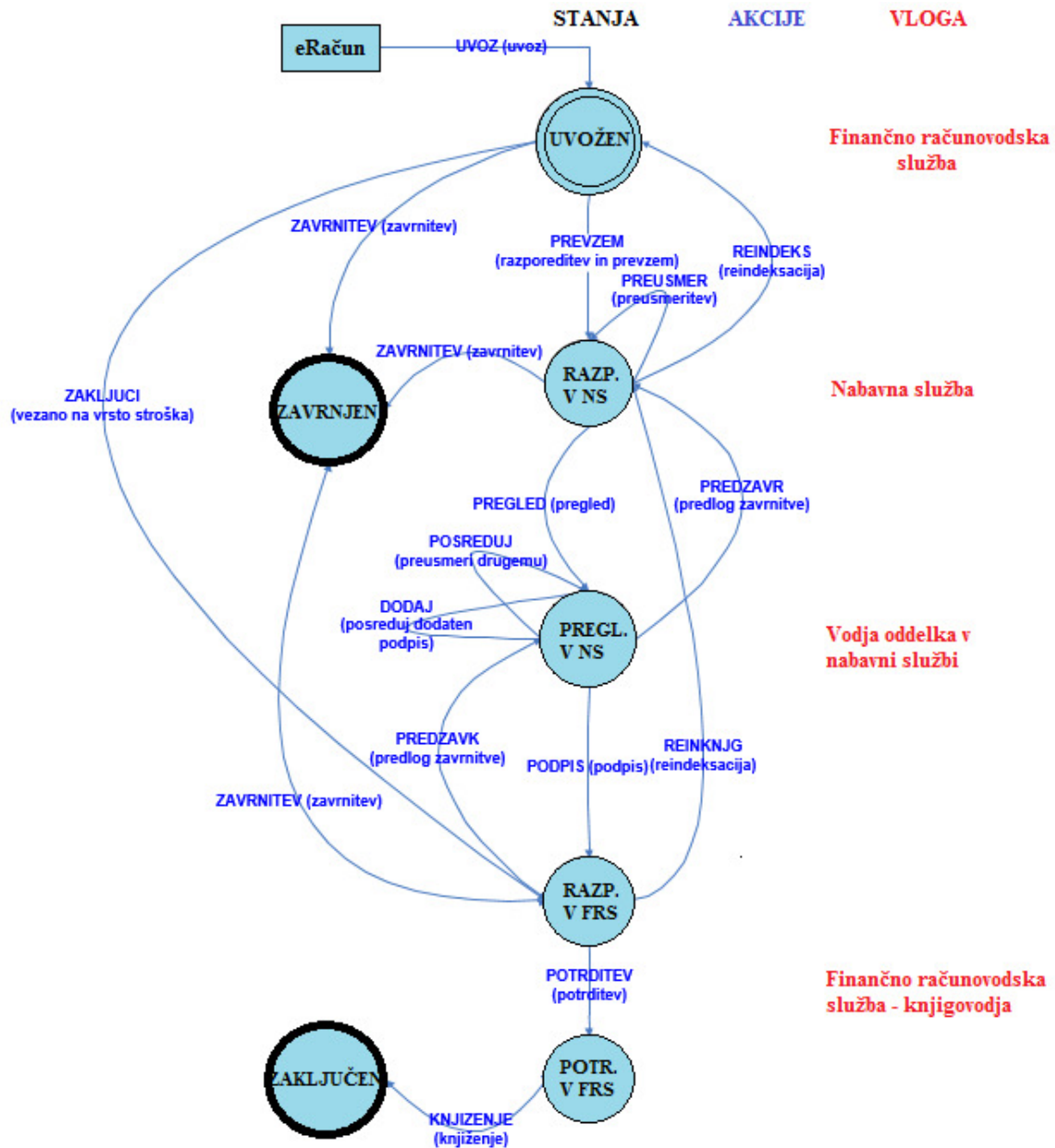
- Evidentiranje podatkov o prispelih dokumentih na podlagi skeniranih ali arhiviranih dokumentov. Standardna različica ePoslovanja bo povezana z aplikacijo eRačuni, iz katere bo prevzemala vse prejete e-račune.
- Samodejno polnjenje atributov dokumenta (v primeru e-računa bo to vsebina iz XML-datoteke računa).
- Samodejno in ročno razporejanje dokumentov med posamezne organizacijske enote ter uporabnike glede na nastavljeni delovni tok in njegova pravila.
- Obdelavo prejetih e-računov in ostalih dokumentov po različnih, nastavljivih delovnih tokovih.
- Administracijo vlog in pravic uporabnikov aplikacije.
- Prenos obdelanih in potrjenih e-računov ter ostalih dokumentov v izbrano informacijsko rešitev naročnika. Standardna različica ePoslovanja bo povezana s Finančno aplikacijo Podjetja za računalniški inženiring.
- Različne izpise iz aplikacije, kot npr. sezname prispelih dokumentov po različnih, uporabniško določenih filtrih glede na osnovne podatke dokumentov ali podatke v postavkah računa, dopise o zavrnitvi računov, izpise PDF slik e-računov in ostalih dokumentov.
- Nadomeščanje uporabnikov v primeru napovedanih in nenapovedanih odsotnosti z dela s strani pooblaščenega namestnika.
- Beleženje revizijske sledi v tabelah za beleženje sprememb.
- Obveščanje uporabnikov o čakajočih dokumentih na njihov elektronski naslov glede na nastavljen časovnik obveščanja.

3.4.2 Delovni tok dokumenta s prehodi stanj in vloge uporabnikov

Aplikacija ePoslovanje bo temeljila na delovnem toku knjigovodske likvidature prispelega računa, v katerem sodelujejo različni uporabniki, iz različnih oddelkov oziroma stroškovnih mest naročnika. Vsak uporabnik v procesu ima določeno vlogo in z njo povezane pravice, pooblastila ter zadolžitve, za katere je odgovoren v procesu likvidature oziroma v fazi, v kateri se bo račun trenutno nahajal.

Stanje dokumenta bo določalo, v katerem delu delovnega toka se bo dokument nahajal. Vsako stanje bo imelo nastavljene dovoljene prehode stanj (predhodno, naslednje stanje). Določene vloge uporabnikov bodo imele pravico obdelovati samo določena stanja dokumentov. Na Sliki 19 so prikazana stanja dokumenta in prehodi med stanji, vloge ter akcije, ki jih bo uporabnik lahko opravil za dokument v določenem stanju.

Slika 19: Delovni tok s prehodi stanj, vlogami in možnimi akcijami



Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije ePoslovanje, 2015a.

Na vlogo uporabnikov bo vezana tudi aktivnost ali neaktivnost določenih menijev, prikazov in gumbov. Nekatere vloge bodo vezane na sodelovanje uporabnika v delovnem toku likvidature, druge pa bodo namenjene le administratorju aplikacije. Uporabnik bo lahko imel eno ali več vlog.

Poleg vloge se bo uporabniku določilo tudi pravice za delo s posameznim oddelkom (stroškovnim mestom), več oddelki ali celotno organizacijo (v tem primeru bo imel pravice za delo z vsemi oddelki). Kombinacija vloge in pravice za delo z oddelkom bo omogočala, da bo uporabnik v aplikaciji ePoslovanje videl ter obdeloval samo račune, za katere bo

zadolžen. Aplikacija bo za uporabnika predstavljala virtualno pisarno, v kateri bo obdeloval samo svoje račune, in sicer le v delu procesa likvidature, za katerega bo določen.

Stanja dokumenta v aplikaciji ePoslovanje in akcije, dovoljene v posameznem stanju, bodo:

- **Uvožen.** Stanje bo označevalo vse račune, ki so bili uvoženi iz aplikacije eRačun. Možne akcije v tem stanju bodo:
 - razporeditev in prevzem,
 - zavrnitev,
 - zaključitev (vezano na določeno vrsto stroška).
- **Razporejen v NS.** Gre za drugo stanje v delovnem toku. Za račune v tem stanju bodo možne naslednje akcije:
 - pregled,
 - reindeksacija,
 - preusmeritev in
 - zavrnitev.
- **Pregledan v NS.** Gre za tretje stanje, v katerem bodo možne naslednje akcije:
 - podpis,
 - predlog zavrnitve,
 - preusmeritev in
 - posredovanje dodatnemu v podpis.
- **Razporejen v FRS.** V tem stanju bodo možne naslednje akcije:
 - potrditev,
 - reindeksacija,
 - predlog zavrnitve in
 - zavrnitev.
- **Potrjen v FRS.** Urejanje dokumenta v tem stanju ne bo več mogoče. Dovoljena akcija bo samo knjiženje, ki bo račune prenesla v Finančno aplikacijo.
- **Zaključen.** Gre za končno akcijo dokumenta. Dokumenti preidejo v to stanje po prenosu v Finančno aplikacijo podjetja.
- **Zavrnjen.** Drugo končno stanje. Označevalo bo račune, ki bodo zavrnjeni dobavitelju.

Določena vloga uporabnika bo imela pravice na vpogled in delo le z določenimi stanji dokumenta. Vloge uporabnikov bodo naslednje:

- finančno-računovodska služba (v nadaljevanju FRS),
- nabavna služba,
- vodja oddelka v nabavni službi,
- finančno računovodska služba – knjigovodja (v nadaljevanju knjigovodja),
- helpdesk,

- urejevalec nadomeščanj in
- administrator aplikacije.

Vloga FRS. Gre za prvo vlogo v procesu likvidature računa. Vloga vidi vse račune, ki so v stanju uvožen. Uporabniki s to vlogo bodo v aplikacijo prevzeli račune iz aplikacije eRačun, izvedli dopolnitev podatkov in račun glede na vsebino računa posredovali v nadaljnjo obdelavo ustreznemu oddelku nabavne službe. Možne akcije bodo naslednje:

- **Razporeditev in prevzem:** račun bo poslan v nadaljnjo obdelavo v nabavno službo in s tem preide v stanje razporejen v NS. Če se bo prek vrednosti v polju številka naročilnice lahko določilo uporabnika, ki je naročilo izdal (podatke se bo dobilo iz Finančne aplikacije), se bo račun posredovalo neposredno k njemu in ga bo lahko obdeloval le on. V nasprotnem primeru bodo račun lahko obdelovali vsi nabavni referenti posameznega oddelka.
- **Zavrnitev:** račun bo zavrnjen dobavitelju in s tem prešel v stanje zavrnjen.
- **Zaključitev:** v primeru določenih vrednosti v polju vrsta stroška se bo račun lahko posredovalo neposredno h knjigovodji. V tem primeru bo račun prešel v stanje razporejen v FRS.

Vloga nabavna služba. Gre za drugo vlogo v procesu likvidature. Vloga bo videla vse račune, ki so v stanju razporejen v NS. Uporabniki bodo dopolnili dodatno zahtevane podatke in glede na naročilnico, prevzemnico ter dobavnico izvedli knjiženje postavk (stroškov) računa. Možne akcije bodo naslednje:

- **Pregled:** Po dodajanju zahtevanih podatkov bodo uporabniki račun posredovali v podpis vodji svojega oddelka. Račun bo s tem prešel v stanje pregledan v NS.
- **Reindeksacija:** uporabnik vrne račun nazaj v FRS, kjer bodo popravili ali dopolnili podatke, za katere so zadolženi. S tem se bo spremenilo stanje dokumentu na uvožen.
- **Preusmeritev:** uporabnik jo bo uporabil, ko bo želel račune preusmeriti na točno določenega uporabnika z vlogo nabavna služba. Stanje dokumenta se z izvedbo te akcije ne bo spremenilo.
- **Zavrnitev:** račun se bo zavrnil dobavitelju in s tem prešel v stanje zavrnjen.

Vloga vodja oddelka v nabavni službi. Vloga predstavlja podpisnika dokumenta. Vloga bo videla vse račune, ki bodo v stanju pregledan v NS. Vodja oddelka bo imel na voljo naslednje akcije:

- **Podpis:** Če bo znesek računa v okvirih, za katere je vodja oddelka v nabavni službi pristojen (vezano na šifrant), bo račun podpisal in posredoval naprej h knjigovodji. Status računa se bo s tem spremenil na razporejen v FRS.

- **Predlog zavrnitve:** uporabnik bo vrnil račun nazaj v nabavno službo, kjer bodo popravili ali dopolnili podatke, za katere so zadolženi. S tem se spremeni stanje dokumentu na razporejen v NS.
- **Preusmeritev:** uporabnik bo s to akcijo posredoval račun drugemu uporabniku, z isto vlogo. Akcija ne bo spremenila stanja dokumenta.
- **Posredovanje dodatnemu v podpis:** če bo znesek računa zunaj okvirov podpisovanja, za katere bo vodja zadolžen, računa ne bo mogel podpisati. V tem primeru ga bo moral posredovati v dodatni podpis vodji oddelka z višjimi pooblastili. Akcija ne bo spreminjala stanja dokumenta.

Vloga knjigovodja. Gre za zadnjo vlogo v procesu likvidature. Vloga bo videla vse račune, ki bodo v stanju razporejen v FRS ali potrjen v FRS. Vloga bo prejemale račune vseh oddelkov. Uporabniki v tej vlogi bodo dopolniti podatke o davku na dodano vrednost. Možne akcije v tej vlogi bodo naslednje:

- **Potrditev:** uporabnik bo s to akcijo račun potrdil, kar pomeni, da je njegova likvidatura končana. Akcija bo spremenila stanje dokumenta na potrjen v FRS.
- **Reindeksacija:** uporabnik bo z njo vrnil račun nazaj v FRS, kjer bodo popravili ali dopolnili podatke, za katere so zadolženi. S tem se bo spremenilo stanje dokumenta na uvožen.
- **Predlog zavrnitve:** uporabnik bo z njo vrnil račun nazaj k vodji oddelka nabavne službe. Stanje računa se bo spremenilo na pregledan v NS.
- **Zavrnitev:** račun se bo zavrnil dobavitelju in s tem prešel v stanje zavrnjen.

Vloga helpdesk. Vloga ne bo sodelovala v procesu likvidature računa. Uporabniki s to vlogo bodo imeli pravico za delo s posebno masko, na kateri bo mogoče popravljati posamezne attribute računa, stanja in naslovnike računa. Na maski se bodo prikazali vsi računi, vseh oddelkov, ne glede na njihovo stanje.

Vloga urejevalec nadomeščanj: Vloga ne bo sodelovala v postopku likvidature računa. Uporabniki s to vlogo bodo imeli pravico do dela s posebno masko, na kateri se bo urejalo napovedane in nenapovedane odsotnosti uporabnikov ter določalo njihove namestnike.

Vloga administrator aplikacije: Vloga ne bo sodelovala v postopku likvidature računa. Uporabnik s to vlogo bo skrbel za vzdrževanje šifrantov aplikacije, nastavitve in določitev pravic za delo s posameznimi oddelki.

3.4.3 Opis uporabniškega vmesnika in menijev

Prijava v aplikacijo se bo opravila prek uporabnika operacijskega sistema. Za uspešno prijavo v aplikacijo bo moral biti ta s strani administratorja vnesen v šifrant uporabnikov.

Po uspešni prijavi se bo uporabniku prikazal **meni domov**, ki bo predstavljal pozdravno okno z informacijami o dokumentih, ki čakajo na obdelavo v posamezni vlogi uporabnika.

V osrednjem delu se bodo prikazale dodeljene vloge uporabnika s številom čakajočih računov posamezne vloge, kar je vidno na Sliki 20. Pod vlogo se bo izpisovalo dodatno opozorilo v rdeči barvi za tiste račune, ki jim je datum valute že potekel in obvestilo v oranžni barvi za tiste, ki jim bo datum valute potekel v kratkem. Število dni za opozorilo se bo lahko vneslo v nastavitveno datoteko aplikacije.

Slika 20: Meni domov

eposlovanje

Domov

Pozdravljeni, Štefan Zupančič. Trenutno so vaši aktivni dokumenti:

[Kot 'Finančno računovodska služba - knjigovodja' imate en dokument.](#)
Datum valute je že potekel enemu dokumentu!

[Kot 'Vodja oddelka v nabavni službi' imate en dokument.](#)

[Kot 'Nabavna služba' imate 2 dokumenta.](#)
Datum valute je že potekel enemu dokumentu!

Prikazani zapisi 1 do 2 izmed 2

Datum prejema	Stanje	DOFA	Številka dokumenta	Datum računa	Datum valute	Znesek	Val	Partner	Naziv partnerja	Vsebina
03.11.2014	Razporejen v NS	51424085	6910036-14	02.11.2014	17.11.2014	4,60	EUR	0		
26.11.2014	Razporejen v NS	-	51-14	26.11.2014	11.12.2014	9,89	EUR	0		

[Kot 'Finančno računovodska služba' imate 20 dokumentov.](#)
V kratkem bo potekel datum valute enemu dokumentu!
Datum valute je že potekel 3 dokumentom!

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije ePoslovanje, 2015a.

Po izboru vloge se bo odprl seznam vseh dokumentov, ki čakajo na obdelavo v tej vlogi. Po posameznih stolpcih se bodo prikazovali podatki o računu, ki se jih bo lahko urejalo naraščajoče ali padajoče. Uporabnik bo iz seznama izbral dokument, ki se bo odprl na novi pregledni maski za urejanje dokumenta.

Na levi strani aplikacije bodo dostopni meniji. Prikaz menijev bo odvisen od uporabnikovih pravic za delo z aplikacijo. Meniji bodo naslednji:

- domov,
- prevzem,
- prejeti računi,
- ostali dokumenti,
- izpisi,
- prenos,
- šifranti,
- administracija in
- administrativno urejanje.

Meni prevzem bo viden samo uporabniku z vlogo FRS oziroma uporabniku, ki bo imel dodeljeno pravico za prevzem dokumentov iz elektronskega arhiva oziroma iz aplikacije eRačun. Dokumenti se bodo iz aplikacije eRačun samodejno uvažali glede na časovni interval, ki bo nastavljen v nastavitveni datoteki ali ročno z namenskim gumbom za prevzem dokumentov.

Meni prejeti računi je prikazan na Sliki 21 in bo namenjen pregledovanju računov v obdelavi, ne glede na status, v katerem so. Izjema bodo zavrženi in zaključeni računi, ki v pregledovalniku ne bodo vidni, kar pomeni, da bodo na pregledni maski vidni le aktivni računi. Po izboru računa se bo ta odprl v maski za urejanje dokumenta. Popravljanje in urejanje računa, ki je dodeljen v obdelavo drugemu uporabniku, ne bo mogoče, ker bodo v tem primeru vsa polja maske zaklenjena. Po posameznih stolpcih se bodo prikazovali podatki o računu, ki se jih bo lahko urejalo naraščajoče ali padajoče.

Slika 21: Meni prejeti računi

Prejeti računi										
+ Iskalni pogoji										
Prikazani zapisi 1 do 24 izmed 24										
Datum prejema	Stanje	DOFA	Številka dokumenta	Datum računa	Znesek	Val	Datum valute	Partner	Naziv partnerja	Vsebina
19.10.2014	Uvožen	-	6910036-14	02.11.2014	4,60	EUR	17.11.2014	0		Račun številka: 6910036-14 z dne: 02.11....
03.11.2014	Razporejen v NS	51424085	6910036-14	02.11.2014	4,60	EUR	17.11.2014	0		Račun številka: 6910036-14 z dne: 02.11....
26.11.2014	Uvožen	-	51-14444	26.11.2014	9,89	EUR	30.11.2014	0		Račun številka: 6910050-14 z dne: 26.11....
20.11.2014	Uvožen	-	44-14	18.11.2014	4,60	EUR	03.12.2014	0		Račun številka: 6910044-14 z dne: 18.11....
28.11.2014	Uvožen	-	201400016	28.11.2014	85,40	EUR	06.12.2014	0		Račun številka: 201400016 z dne: 28.11.2....
01.12.2014	Uvožen	-	20140054	01.12.2014	85,40	EUR	09.12.2014	0		Račun številka: 20140054 z dne: 01.12.20....
01.12.2014	Uvožen	-	20140053	01.12.2014	85,40	EUR	09.12.2014	0		Račun številka: 20140053 z dne: 01.12.20....
01.12.2014	Uvožen	-	20140056	01.12.2014	85,40	EUR	09.12.2014	0		Račun številka: 20140056 z dne: 01.12.20....
01.12.2014	Uvožen	-	20140055	01.12.2014	85,40	EUR	09.12.2014	0		Račun številka: 20140055 z dne: 01.12.20....
01.12.2014	Uvožen	-	201400027	01.12.2014	1.056,00	EUR	09.12.2014	0		Račun številka: 201400027 z dne: 01.12.2....
26.11.2014	Uvožen	-	51-14	26.11.2014	9,89	EUR	11.12.2014	0201		Račun številka: 6910050-14 z dne: 26.11....
26.11.2014	Razporejen v NS	-	51-14	26.11.2014	9,89	EUR	11.12.2014	0		Račun številka: 6910050-14 z dne: 26.11....
26.11.2014	Pregledan v NS	51424087	51-14	26.11.2014	9,89	EUR	11.12.2014	0		Račun številka: 6910050-14 z dne: 26.11....
26.11.2014	Uvožen	-	6910050-14	26.11.2014	9,89	EUR	11.12.2014	0		Račun številka: 6910050-14 z dne: 26.11....
26.11.2014	Uvožen	-	51-143	26.11.2014	9,89	EUR	11.12.2014	0		Račun številka: 6910050-14 z dne: 26.11....
28.11.2014	Uvožen	-	6910055-14	28.11.2014	9,89	EUR	13.12.2014	0		Račun številka: 6910055-14 z dne: 28.11....
01.12.2014	Uvožen	-	6910061-14	01.12.2014	9,89	EUR	16.12.2014	0		Račun številka: 6910061-14 z dne: 01.12....
01.12.2014	Uvožen	-	6910063-14	01.12.2014	9,89	EUR	16.12.2014	0		Račun številka: 6910063-14 z dne: 01.12....
01.12.2014	Uvožen	-	6910060-14	01.12.2014	9,89	EUR	16.12.2014	0		Račun številka: 6910060-14 z dne: 01.12....
01.12.2014	Uvožen	-	6910059-14	01.12.2014	9,89	EUR	16.12.2014	0		Račun številka: 6910059-14 z dne: 01.12....
01.12.2014	Uvožen	-	6910064-14	01.12.2014	9,89	EUR	16.12.2014	0		Račun številka: 6910064-14 z dne: 01.12....
01.12.2014	Uvožen	-	6910062-14	01.12.2014	9,89	EUR	16.12.2014	0		Račun številka: 6910062-14 z dne: 01.12....

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije ePoslovanje, 2015a.

Meni izpisi bo namenjen pregledovanju in izpisovanju vseh računov (aktivnih in zaključenih). Po vstopu v meni se bo samodejno odprla forma z iskalnimi pogoji, kjer bo uporabnik omejil iskalne parametre, kar je vidno na Sliki 22. Prikaz rezultatov bo v osrednjem delu maske, kjer se bo posamezni račun prikazal v več vrsticah, in sicer v eni vrstici podatki o obveznosti, v drugi podatki od davku na dodano vrednost ter v ostalih podatki o postavkah računa (stroških). Po posameznih stolpcih se bodo prikazovali podatki o računu, ki se jih bo lahko urejalo naraščajoče ali padajoče. Če bo uporabnik kliknil na dokument, se bo ta odprl na novi maski za pregled in urejanje. Če bo imel uporabnik pravice za delo z računom (glede na vlogo in status računa), ga bo lahko urejal, v

nasprotnem primeru pa bodo vsa polja na maski zaklenjena. V spodnjem delu obrazca bo gumb **izvoz v Excel**, s katerim bo uporabnik seznam računov izvozil v Excel.

Slika 22: Meni izpisi

Izpisi Dokumentov

Iskalni pogoji

☐ Datum sprejema od		do datuma	
☐ Stanje	Uvožen		
☐ DOFA	51424084		
☐ Številka dokumenta	691		
☐ Datum dok. od		do datuma	
☐ Številka partnerja			
☐ Naziv partnerja			
☐ Datum valute od		do datuma	
☐ Stroškovno mesto od		do	
☐ Stroškovni nosilec			
☐ Konto			
☐ Protikonto			
☐ Pogodba od		do	
☐ Projekt od		do	
☐ Debet			
☐ Kredit			
☐ Datum obdobja od		do datuma	
☐ Datum ddv od		do datuma	
☐ Datum prenosa od		do datuma	
☐ Datum plačila od		do datuma	
☐ Vrsta posla			
☐ Vrsta stroška			
☐ Vrsta dokumenta			
☐ ID Dokumenta			
☐ Številka naročilnice		☐ Ali dobropis	
☐ Opomba			

Išči

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije ePoslovanje, 2015a.

Meni ostali dokumenti bo namenjen spremljanju ostalih dokumentov, ki se evidentirajo z drugim delovnim tokom in ne predstavljajo računov.

Meni prenos bo namenjen uporabnikom z vlogo knjigovodje. Ostali uporabniki tega menija ne bodo videli. Na meniju se bodo prikazovali računi s stanjem zaključen, ki jih bo lahko nato knjigovodja z namenskim gumbom prenesel v Finančno aplikacijo.

Meni šifranti bo namenjen pregledu šifrantov, ki so povzeti iz Finančne aplikacije in jih aplikacija ePoslovanje le prikazuje, ter nastavitvi šifrantov specifični aplikaciji ePoslovanje. Šifranti, ki jih bo aplikacija ePoslovanje samo prikazovala, bodo naslednji: organizacijske enote, poslovni partnerji, parametri PE, kontni plan, odstotek uveljavitve

DDV, vrste računov, vrste stroškov in vrste posla. Maske za pregled šifrantov bodo podobne kot so njihove maske v Finančni aplikaciji.

Šifranti specifični aplikaciji ePoslovanje bodo naslednji:

- **Stanja dokumenta:** v šifrantu bodo vnesena vsa stanja dokumenta (uvožen, razporejen v NS, pregledan v NS, razporejen v FRS, potrjen v FRS, zaključen in zavržen). Dodani bodo gumbi za urejanje naziva stanja in vnos novega stanja ter brisanje.
- **Nadomeščanja:** šifrant bo namenjen administratorju in uporabniku z vlogo urejevalec nadomeščanj. V šifrantu se bo lahko določilo odsotnega uporabnika (spustni seznam vseh uporabnikov), namestnika (spusti seznam vseh uporabnikov) ter datum od - do. Dodana bosta še gumba za shranjevanje, urejanje in brisanje zapisa. Če bo uporabnik nadomeščal odsotnega, bo s tem prevzel vse njegove vloge in pravice, ki jih ima odsotni uporabnik v aplikaciji. Zaradi revizijske sledi se bo v vse tabele sprememb evidentiralo, da so bile spremembe narejene s strani namestnika.

Meni administracija bo namenjen uporabniku s pravico administratorja, in sicer za nastavitve uporabnikov aplikacije, njihovih vlog, pravic za delo in ostalih potrebnih nastavitvev. Šifranti tega menija bodo naslednji:

- **Uporabniki:** v šifrant bo treba vnesti vse uporabnike aplikacije, in sicer ime, priimek in šifro uporabnika operacijskega sistema, ki ga oseba uporablja na svojem računalniku. Dodatno se bosta na nivoju uporabnika lahko vnesla elektronski naslov za obveščanje o zapadlih dokumentih in mejni znesek podpisovanja za podpisnike računov (vloga vodja oddelka v nabavni službi). Podatki bodo vrstično prikazani po stolpcih, ki se jih bo lahko urejalo naraščajoče ali padajoče. Dodani bodo gumbi za dodajanje, urejanje in brisanje zapisa.
- **Vloge uporabnikov:** v šifrantu se bo lahko za posameznega uporabnika določilo eno ali več vlog, ki jih bo imel v aplikaciji. Na posamezni vlogi se bo lahko določilo tudi to, ali ima vloga samo pravico vpogleda ali tudi urejanja. Podatki bodo vrstično prikazani po stolpcih, ki se jih bo lahko urejalo naraščajoče ali padajoče. Dodani bodo gumbi za dodajanje, urejanje in brisanje zapisa.
- **Podpisniki:** šifrant bo namenjen določanju podpisnikov posameznega uporabnika. Za vsakega uporabnika z vlogo FRS se bo definiralo enega ali več podpisnikov (uporabnik z vlogo vodja oddelka v nabavni službi). Podatki bodo vrstično prikazani po stolpcih, ki se jih bo lahko urejalo naraščajoče ali padajoče. Dodani bodo gumbi za dodajanje, urejanje in brisanje zapisa.
- **Nivo pravic za delo na OE:** v šifrantu se bo posameznemu uporabniku dodalo pravice za delo z enim ali več oddelki ali celotno organizacijo. Podatki bodo vrstično prikazani po stolpcih, ki se jih bo lahko urejalo naraščajoče ali padajoče. Dodani bodo gumbi za dodajanje, urejanje in brisanje zapisa.

- **Zahtevani atributi dokumentov:** v meniju bodo določeni zahtevani atributi (obvezna polja), ki bodo morali biti vneseni na dokumentu, da bo ta lahko prešel v naslednje stanje. Podatki bodo vrstično prikazani po stolpcih, ki se jih bo lahko urejalo naraščajoče ali padajoče. Dodani bodo gumbi za dodajanje, urejanje in brisanje zapisa.

Meni administrativno urejanje bo namenjen uporabniku z vlogo helpdesk ali administratorju aplikacije. Prek menija se bo lahko posameznemu dokumentu spreminjalo status, podpisnike in njegove naslovnike. Podatki bodo vrstično prikazani po stolpcih, ki se jih bo lahko urejalo naraščajoče ali padajoče. Dodani bodo gumbi za urejanje in shranitev zapisa.

3.4.4 Maska za delo z dokumenti

Maska bo predstavljala osnovno masko, prek katere bodo uporabniki vnašali, spreminjali in pregledovali e-račune. Do maske bo uporabnik dostopal tako, da bo odprl račun z menija domov (seznam računov po posameznih vlogah), menija prejeti računi ali menija izpisi. Polja za vnos bodo na maski zaklenjena ali odklenjena, glede na vlogo uporabnika, pravico uporabnika za delo z oddelkom in stanje, v katerem se bo dokument nahajal. Zaključene račune se bo npr. lahko samo pregledovalo, saj bodo vsa vnosna polja zaklenjena.

Maska bo razdeljena na dva dela, in sicer na del, ki bo namenjen vnosu podatkov ter del, ki bo namenjen pregledu PDF slike računa (ali njegove vizualizacije). Del, ki bo prikazoval sliko, se bo lahko poljubno širilo ali ožilo, slika pa se bo lahko tudi povečala. Prikazovanje slike bo uporabnikom omogočalo poenostavljeno likvidacijo e-računa. V zgornjem delu se bodo prikazovali zavihki po posameznih vsebinskih sklopih. Zavihki bodo naslednji:

- Dokument: namenjen bo vnosu podatkov o obveznostih.
- DDV: namenjen bo vnosu podatkov o davku na dodano vrednost.
- Postavke: na zavihku se bo evidentiralo podatke o postavkah (stroških) računa.
- Priloge: zavihke bo namenjen spremljanju in dodajanju prilog e-računa.
- Dodatno: zavihke za spremljanje dodatnih podatkov o računu, ki niso računovodske narave.
- Zgodovina: na zavihku se bo prikazovala zgodovina sprememb dokumenta.

Zavihek dokument je prikazan na Sliki 23 in bo namenjen vnosu podatkov z glave računa oziroma podatkov o obveznosti do dobavitelja. Posamezna polja se bodo na računu napolnila že z uvozom e-računa, in sicer iz njegove XML-datoteke. Vnosna polja bodo izvedena kot polja za vnos ali kot spustni šifranti za sezname. Na maski bodo vsa polja, ki jih za vodenje stroškovnega računovodstva in uspešno pripravo plačilnih nalogov potrebuje Finančna aplikacija, in sicer podatki o dobavitelju, sklicu, številki računa, kontu, datumih itd. Na dnu maske bodo gumbi za shranjevanje, preklic sprememb in pošiljanje dokumenta naslednjemu uporabniku v postopku obdelave dokumenta.

Slika 23: Zavihek dokument

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije ePoslovanje, 2015a.

Zavihek DDV bo namenjen spremljanju in vnosu podatkov, ki jih Finančna aplikacija potrebuje za poročanje o davku na dodano vrednost. V zgornjem delu maske, ki je prikazana na Sliki 24, se bodo prikazovali osnovni podatki računa (dofa, datum, znesek itd.), spodnji del pa bo predstavljal vnosna polja. Na dnu maske bodo gumbi za shranitev, preklic sprememb in pomožni izračun DDV, ki bo uporabniku glede na vneseno osnovo za izračun davka, stopnjo davka in vrsto posla (uvoz, dobava blaga v Skupnosti itd.) pomagal pri izračunu zneskov davčnih osnov ter odbitega in obračunanega davka na dodano vrednost.

Slika 24: Zavihek DDV

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije ePoslovanje, 2015a.

Zavihek postavke bo namenjen vnosu stroškov računa. V zgornjem delu maske, ki je prikazana na Sliki 25, se bodo prikazovali osnovni podatki računa (dofa, datum, znesek itd.), spodnji del pa bo predstavljal pregledna in vnosna polja. Na pregledni maski se bodo prikazovale že vnesene postavke računa, ki jih bo lahko uporabnik urejal – v tem primeru se bo pod pregledom vrstice odprl obrazec za urejanje polj. Po posameznih stolpcih, ki se jih bo lahko urejalo naraščajoče ali padajoče, se bodo prikazovali podatki, ki jih Finančna aplikacija potrebuje za spremljanje stroškovnega računovodstva (konto, znesek, stroškovno mesto, stroškovni nosilec itd.). Na masko bodo dodani gumbi za brisanje, shranitev, preklic sprememb in uvoz delilnika.

Slika 25: Zavihek postavke

Str. mesto	Konto	Debet	Kredit	Str. nos.	Protikonto	Opomba
001 - STM ZA OEZR 0	400001	1.094,42		002 - test	400001	Račun številka: 201430107 z dn...
0001 - Zakladnica	400001	0,58		001 - stanovanjski sklad	400001	Račun številka: 201430107 z dn...
0001 - Zakladnica	400001	10,00			400001	Račun številka: 201430107 z dn...
002 - KOMERCIALA	400001	1.000,00	10,00		400001	Račun številka: 201430107 z dn...

Skupaj debet: 2.105,00 kredit: 1.105,00 SALDO: -1.000,00

Konto: 400001 | Debet: 1094,42 | Str. mesto: 001 | Str. nosilec: 002 - test | Protikonto: 400001 | Analitika: AA | Ref. dok: 1111

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije ePoslovanje, 2015a.

Zavihek priloge bo namenjen spremljanju prilog e-račun in dodajanju novih prilog, ki jih ima uporabnik shranjene lokalno in so potrebne za uspešno knjigovodsko likvidacijo računa. V zgornjem delu maske, ki je prikazana na Sliki 26, se bodo prikazovali osnovni podatki računa (dofa, datum, znesek itd.). Priloge se bodo prikazovale vrstično. Uporabnik bo prilogo lahko odprl, natisnil ali jo shranil. Na dnu maske se bo dodalo gumbe za brisanje in dodajanje prilog. Brisanje prilog bo omejeno samo na priloge, ki so bile ročno dodane k računu.

Slika 26: Zavihek priloge

ID pril.	Tip priloge	Opis
57	APPLICATION/PDF	-
58	APPLICATION/XML	eRačun

Vir: Podjetje za računalniški inženiring, Uporabniški priročnik aplikacije ePoslovanje, 2015a.

SKLEP

Pri razvoju informacijskih sistemov in programskih rešitev je zelo pomembna izbira ustrezne metodologije razvoja, ki razvijalcem omogoča skrajšanje razvojnega časa ter lažje vodenje in obvladovanje projekta. Ustrezna izbira metodologije prinaša prednosti tudi naročniku, ki se kažejo v zadovoljstvu uporabnikov pri delu z implementirano rešitvijo.

Najbolj razširjena delitev metodologij slednje ločuje na dve veliki skupini, in sicer na tradicionalne, imenovane tudi težke, ter na agilne, ki se v literaturi označujejo tudi kot gibke ali lahke metodologije.

Tako tradicionalne kot tudi agilne metodologije prinašajo nekatere prednosti, vendar se je pri odločitvi glede njihove uporabe treba zavedati tudi omejitev in slabosti, ki jih posamezna metodologija prinaša. Obstajajo področja razvoja, za katera so nekatere metodologije primernejše od drugih in tudi takšna, za katere določene metodologije sploh niso primerne oziroma je njihova uporaba lahko celo kritična. Metodologije niso dobre in slabe ter posledično neuporabne, temveč so lahko le primerne oziroma neprimerne in posledično uspešne ali manj uspešne. Tudi izbira neprimerne metodologije namreč lahko privede do končnega rezultata, vendar bo za njega treba vložiti več sredstev in navora. V primeru razvoja programskih rešitev za Izbrano podjetje je bila uporabljena agilna metodologija, ki je temeljila na metodi poenotenega pristopa (angl. *Unified Process*), s prvinami tradicionalne metodologije Oracle CASE. Razlog za uporabo agilnejšega razvojnega pristopa je bil predvsem v slabo definiranih začetnih zahtevah in željah naročnika ter kratkem razvojnem času, ki je bil omejen na nekaj mesecev. Slabo definiranim začetnim zahtevam je botroval zelo kompleksen delovni tok likvidature prejetih računov, ki ga Izbrano podjetje uporablja. V njem sodeluje več kot sto uporabnikov iz različnih oddelkov – vsak s svojimi posebnostmi pri obdelavi. Naročnik je tako v začetni fazi definirale osnovni delovni tok računa in izpostavil dejstvo, da se bodo zahteve skozi izgradnjo tudi spreminjale. Metodologija poenotenega pristopa je omogočila, da se je rešitev razvijala po delih, ki so bili naročniku tudi predani v testiranje. Ugotovitve, pripombe in nove zahteve, ugotovljene v fazi testa, so bile implementirane v naslednjih iteracijah, ki so si sledile. Uporaba tradicionalnih teorij razvoja v dani situaciji ne bi predstavljala optimalne izbire, ker bi celoten proces razvoja trajal predolgo, hkrati pa razvita rešitev ne bi predstavljala optimalne rešitve, ker so se zahteve naročnika v postopku izgradnje večkrat dopolnile in spremenile.

Razvoj informacijskih tehnologij je prinesel tudi razvoj elektronskega poslovanja, ki v osnovi predstavlja poslovanje in iteracije med podjetji, kupci ter državo, ki se odvija na elektronskih medijih in prek elektronskih poti. Uvedba elektronskega poslovanja v podjetje zanj pomeni organizacijske spremembe, pa tudi prednosti, ki jih lahko od njega pričakuje. Enega ključnih elementov elektronskega poslovanja predstavlja e-račun, to je listina, ki jo izdajatelj računa izda svojemu dolžniku oziroma prejemniku računa v elektronski obliki in

enakovredno zamenjuje račun v papirni obliki. Ključne premike v smeri elektronskega poslovanja in prelomnico v slovenskem gospodarstvu je prinesla uveljavitev ZOPSPU-A, ki je s 1. januarjem 2015 proračunskim uporabnikom nalagal obvezen prejem in izdajo vseh računov v elektronski obliki.

Cilj magistrskega dela je bil priprava analize informacijske rešitve, ki bo Izbranemu podjetju omogočala prevzem in obdelavo e-računov skladno z ZOPSPU-A. V delu sta predstavljeni analizi za dve novi informacijski rešitvi, in sicer za aplikaciji eRačun ter ePoslovanje.

Aplikacija eRačun se bo uporabljala za prejem in pošiljanje e-računov, pri čemer se bo izmenjava z UJP izvajala samodejno, prek vmesnika UJP B2B in nastavljenega samodejnega časovnika, omogočeno pa bo tudi ročno kreiranje datoteke e-računov. Aplikacija bo predstavljala vmesni člen med obstoječo Finančno aplikacijo, v kateri Izbrano podjetje med drugim tudi kreira in vodi izdane račune, ter UJP kot enotno vstopno in izstopno točko za izmenjavo e-računov s proračunskimi uporabniki. Namen aplikacije ePoslovanje bo knjigovodska likvidatura prejetih e-računov na podlagi delovnega toka dokumenta, ki ga uporablja Izbrano podjetje. Aplikacijo bo zaznamoval intuitiven in prijazen uporabniški vmesnik, v katerem bo imel uporabnik možnost pregleda slike in ostale spremljajoče dokumentacije v vseh korakih obdelave dokumenta. Aplikacija bo predstavljala zmes med dokumentnim sistemom s podprtim tokom dokumenta in računovodsko aplikacijo, ker bo v svojem bistvu združevala prvine obeh. Kljub temu, da sta analizirani rešitvi ločeni aplikaciji, se bosta navezovali na obstoječo programsko rešitev, ki jo Izbrano podjetje že uporablja.

Kot analitik Podjetja za računalniški inženiring sem sodeloval na obeh projektih razvoja informacijskih rešitev za Izbrano podjetje. Vključen sem bil v faze strategije, analize izgradnje ter namestitve in uvajanja, v katerih sem aktivno sodeloval s projektanti in programerji ter vodjo projekta. Pri izgradnji je bila ključna uporaba agilne metodologije, s katerimi je bila rešitev razvita v več iteracijah ter dejstvo, da je bila aplikacija ePoslovanje razvita tako, da je posnemala obstoječi delovni tok prejetega računa, ki ga je naročnik že uporabljal, kljub temu, da ta ni optimalen in ponuja veliko priložnosti za izboljšave ter optimizacije. Uvedba aplikacije, ki bi močno spreminjala delovni tok obdelave računa in s tem tudi način dela v organizaciji, v tako kratkem času in ob velikem številu končnih uporabnikov ne bi bila smotrna oziroma izvedljiva. V prvih iteracijah se je razvila aplikacija, ki je podpirala osnovni delovni tok obdelave prejetega računa, posebnosti in specifične posameznih oddelkov pa so bile vgrajene kasneje, ker v prvotni fazi analize in načrtovanja s strani naročnika niso bile definirane. V takšnih situacijah je za uspeh ključna uporaba agilnih metodologij ali vsaj njihovih prvin, ki kljub spremembam zahtev omogočajo izgradnjo zadovoljive rešitve.

LITERATURA IN VIRI

1. Abrahamsson, P., Salo, O., & Ronkainen, J. (2002). *Agile software development methods. Review and analysis*. Espoo: University of Oulu: VTT Publications 478.
2. Alexandrou, M. (2011). Dynamic Systems Development Model (DSDM) Methodology. Najdeno 25. aprila 2016 na spletnem naslovu <http://infolific.com/technology/methodologies/dynamic-systems-development-model/>
3. Ambler, S. W. (2005). Feature Driven Development and Agile Modeling. Najdeno 25. aprila 2016 na spletnem naslovu <http://www.agilemodeling.com/essays/fdd.htm>
4. Arhiv Republike Slovenije. (2013). Enotne tehnološke zahteve, različica 2.1. Ljubljana: Ministrstvo za kulturo. Najdeno 31. maja na spletnem naslovu http://www.arhiv.gov.si/fileadmin/arhiv.gov.si/pageuploads/E-ARHIVI/ETZ_2_1/ETZ_-_II._del_razlicica_2.1_-_koncna.doc.pdf
5. Awad, M. A., (2005). *Comparison between Agile and Traditional Software Development Methodologies*. Perth: School of Computer Science and software Engineering, The University of Western Australia.
6. Bajec, M. (b.l.). Razvoj informacijskih sistemov. Bolonjski študijski program 1.stopnje. Študijsko gradivo, verzija 3.4. Najdeno 27. marca 2016 na spletnem naslovu <https://ucilnica.fri.uni-lj.si/pluginfile.php/792/course/section/1238/p1.pdf>
7. Bajec, M., & Krisper, M. (b.l.). Agilne metodologije razvoja informacijskih sistemov. Najdeno 20. marca 2016 na spletnem naslovu <http://bajecm.fri.uni-lj.si/CRP2001/Clanki/Agilne%20Metodologije%20Razvoja%20IS.pdf>
8. SEPA, E-račun. Najdeno 20. marca 2016 na spletnem naslovu <http://www.bankart.si/si/ponudba/simp/>
9. Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley Professional.
10. Berčič, B. (2004). Projekt e-SLOG – Pravna vprašanja elektronskega podpisa, elektronskega poslovanja in elektronskih arhivov. Najdeno 15. marca 2016 na spletnem naslovu http://www.gzs.si/e-poslovanje/dokumentacija/e-SLOG_Pravna_vprasanja_ePoslovanja_1.4.pdf
11. Bogataj, K., Pucihar, A., & Puc, M. (2007). *Izdajanje in prejemanje e-računov*. Ljubljana: JAPTI – Javna agencija Republike Slovenije za podjetništvo in tuje investicije.
12. Boehm, B. (2002). Get Ready for Agile Methods, with Care. *IEEE Computer*, 35(1), 64–69.
13. Chaffey, D. (2009). *E-business and e-commerce management: strategy, implementation and practice* (4th ed.). Harlow: Prentice Hall.
14. Clifton, M., & Dunlaph, J. (2003). What is DSDM. The Code Projects. Najdeno 22. aprila 2016 na spletnem naslovu <http://www.codeproject.com/Articles/5097/What-Is-DSDM>
15. Cockburn, A., & Highsmith, J. (2001). Agile Software Development, The People Factor. *IEEE Computer*, 34(11), 131–133.

16. Combe, C. (2006). *Introduction to e-business management and strategy*. Boston: Butterworth-Heinemann.
17. Cote, L., Vezina, M., & Sabourin, V. (2005). The strategic management process in e-business. Najdeno 10. maja na spletnem naslovu <http://iveybusinessjournal.com/publication/the-strategic-management-process-in-e-business/>
18. Direktiva 2014/24/EU Evropskega parlamenta in Sveta z dne 26. februarja o javnem naročanju in razveljavitvi Direktive 2004/18/ES. *Uradni list Evropske unije* št. OJ L 94.
19. Direktiva 2014/55/EU Evropskega parlamenta in Sveta z dne 16. aprila 2014 o izdajanju elektronskih računov pri javnem naročanju. *Uradni list Evropske unije* št. OJ L 133.
20. Direktiva Evropskega parlamenta in Sveta 1999/93/ES z dne 13. decembra 1999 o okviru Skupnosti za elektronski podpis. *Uradni list Evropske unije* št. OJ L 13.
21. Direktorat za zakladništvo. (2014). Pojasnilo Ministrstva za finance, Direktorata za zakladništvo, Združenje bank Slovenije v povezavi z obveznim izdajanjem e-računov. Najdeno 30. maja na spletnem naslovu <https://www.ujp.gov.si/DocDir/e-racuni/Pojasnilo%20MF%20Zdru%C5%BEenju%20bank%20Slovenije%20v%20povezavi%20z%20obveznim%20izdajanjem%20e-ra%C4%8Dunov.pdf>
22. Dontigney, E. (b.l.). Risk of E-business. Najdeno 28. aprila na spletnem naslovu <http://yourbusiness.azcentral.com/risks-ebusiness-1368.html>
23. Fowler, M. (2005). The New Methodology. Agile software development. Najdeno 19. marca 2016 na spletnem naslovu <http://www.martinfowler.com/articles/newMethodology.html>
24. Gradišar M., Jaklič, J., Damij, T., & Baloh, P. (2005). *Osnove poslovne informatike*. Ljubljana: Ekonomska fakulteta.
25. Gradišar M., Jaklič, J., & Turk, T. (2007). *Osnove poslovne informatike*. Ljubljana: Ekonomska fakulteta.
26. Highsmith J. (2002). *Agile Software Development Ecosystem*. Boston: Addison-Wesley
27. Hvala, D. (2011). Okretno programiranje. *Monitor Pro*, (pomlad 2011), 30–32.
28. *Software Development Methodologies*. Najdeno 20. aprila 2016 na spletnem naslovu <http://www.itinfo.am/eng/software-development-methodologies/>
29. Izbrano podjetje. (2014). *Organigram organizacije* (interno gradivo). Maribor: Izbrano podjetje.
30. Kovačič A. (1998). *Informatizacija poslovanja*. Ljubljana: Ekonomska fakulteta.
31. Krisper, M., Rupnik, R., Bajec, M., Rožanec, A., Zrnec, A., & Vavpotič, D., (2003). Enotna metodologija razvoja informacijskih sistemov. *Center Vlade RS za informatiko*. Najdeno 16. marca 2016 na spletnem naslovu <http://www2.gov.si/mju/emris.nsf>
32. Kettunen, P. (2009). Adopting keylessons from agile manufacturing to agile software product development. *Technovation*, 29(6–7), 408–422.
33. *Manifesto for Agile Software Development*. Najdeno 12. marca 2016 na spletnem naslovu <http://agilemanifesto.org/>

34. Pekka, A., Outi, S., Jussi, R., & Juhani, W. (2002). Agile software development methods. Najdeno 20. marca 2016 na spletnem naslovu <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>
35. Perrott, B. (2005). Towards a manager's model for e-business strategy decisions. *Journal Of General Management*, 30(4), 73–89.
36. Piliš, D. A. (2011) Enotna vstopna in izstopna točka za e-račune. Uprava Republike Slovenije za javna plačila. Najdeno 21. marca 2016 na spletnem naslovu http://www.ujp.gov.si/DocDir/Eracuni_Posveti16.6.11.pdf
37. Podjetje za računalniški inženiring. (2008). *Prodajna dokumentacija Finančne aplikacije* (interno gradivo). Ljubljana: Podjetje za računalniški inženiring.
38. Podjetje za računalniški inženiring. (2010). *OBR_analiza* (interno gradivo). Ljubljana: Podjetje za računalniški inženiring.
39. Podjetje za računalniški inženiring. (2011). *Opis produktne dokumentacije* (interno gradivo). Ljubljana: Podjetje za računalniški inženiring.
40. Podjetje za računalniški inženiring. (2014). *Standardni predpis št. 01* (interno gradivo). Ljubljana: Podjetje za računalniški inženiring.
41. Podjetje za računalniški inženiring. (2015a). *Uporabniški priročnik aplikacije ePoslovanje* (interno gradivo). Ljubljana: Podjetje za računalniški inženiring.
42. Podjetje za računalniški inženiring. (2015b). *Uporabniški priročnik aplikacije eRačun* (interno gradivo). Ljubljana: Podjetje za računalniški inženiring.
43. Pravilnik o izvajanju Zakona o davku na dodano vrednost. *Uradni list RS* št. 141/2006, 52/2007, 120/2007, 21/2008, 123/2008, 105/2009, 27/2010, 104/2010, 110/2010, 82/2011, 106/2011, 108/2011, 102/2012, 54/2013, 85/2014, 95/2014, 39/2016.
44. Pravilnik o standardih in pogojih izmenjave elektronskih računov prek enotne vstopne iz izstopne točke pri Upravi Republike Slovenije za javna plačila. *Uradni list RS* št. 9/2011, 109/2012, 75/2014, 94/2014, 75/2015, 9/2016.
45. Pulli, H. (2005). Factors Affecting the Adoption of Electronic Invoicing. Najdeno 22. aprila na spletnem naslovu <http://www.doria.fi/bitstream/handle/10024/29980/TMP.objres.107.pdf?sequence=1>
46. *Elektronski računi*. Najdeno 19. marca 2016 na spletnem naslovu http://www.racunovodja.com/clanki.asp?clanek=7026/Elektronski_racuni
47. Resnik, S. (2010, 12. avgust). Prednosti in slabosti e-poslovanja. Najdeno 19. marca 2016 na spletnem naslovu <http://mladipodjetnik.si/novice-in-dogodki/novice/prednosti-in-slabosti-e-poslovanja>
48. Rupnik, R. (b.1.). Osnove informacijskih sistemov. Najdeno 21. marca 2016 na spletnem naslovu http://www.google.si/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=4&cad=rja&uact=8&ved=0ahUKEwjOr4Shwb3LAhXIE5oKHWQICMsQFggvMAM&url=http%3A%2F%2Fwww.fri.uni-lj.si%2Ffile%2F69231%2Fois-predavanja_v2005_2006-v04.pdf&usq=AFQjCNGOarFgnYqcCP9-cx11aL7QW0TsCg&sig2=FeLuSBdAuZO_xaf0MRWFgg

49. Schwaber, K. (1996). *Controlled chaos: Livin on the Edge*. Najdeno 22. aprila 2016 na <http://static1.1.sqspcdn.com/static/f/447037/6485970/1270926057073/Living+on+the+Edge.pdf?token=0SWXPxt1RRNoqL%2B27dgI4mHEkA%3D>
50. Shore, J., & Warden, S., (2008). *The Art of Agile Development*. Sebastopol: O'Reilly Media.
51. Skrt, R. (2001, 14. marec). Medpodjetniško spletno poslovanje. Najdeno 18. marca 2016 na spletnem naslovu <http://www.nasvet.com/b2b/>
52. Skrt, R. (2002, 12. september). B2B poslovanje. Najdeno 18. marca 2016 na spletnem naslovu <http://www.nasvet.com/b2b-poslovanje/>
53. Skrt, R. (2006, 21. junij). Elektronski zajem in arhiviranje dokumentov. Najdeno 18. marca 2016 na spletnem naslovu <http://www.nasvet.com/elektronsko-arhiviranje/>
54. *Slovenski Nacionalni Forum za eRačun*. Najdeno 20. marca 2016 na spletnem naslovu <https://slovenskieracun.gzs.si>
55. *Slovenski računovodski standardi 2006*. (2007). Ljubljana: Slovenski inštitut za revizijo, Zveza računovodij, finančnikov in revizorjev Slovenije.
56. Uprava Republike Slovenije za javna plačila. (b.1.). eRačun – proračunski uporabniki. Najdeno 19. marca 2016 na spletnem naslovu <http://www.ujp.gov.si/dokumenti/dokument.asp?id=258>
57. Uprava Republike Slovenije za javna plačila. (2012). Pogoji spletnega poslovanja med proračunskimi uporabniki in upravo Republike Slovenije za javna plačila. Najdeno 30. maja na spletnem naslovu https://www.ujp.gov.si/DocDir/Delovna%20podrocja/Spletna%20aplikacija%20UJPnet/Pogoji%20spletnega%20poslovanja_UJPnet_objavljeno_27-03-2012.pdf
58. Uprava Republike Slovenije za javna plačila. (2014, 6. januar). Obvestilo. Najdeno 19. marca 2016 na spletnem naslovu http://www.ujp.gov.si/novice.asp?id_n=303
59. Uredba o pogojih za elektronsko poslovanje in elektronsko podpisovanje. *Uradni list RS* št. 77/2000, 2/2001, 86/2006.
60. Uredba o varovanju dokumentarnega in arhivskega gradiva. *Uradni list RS* št. 86/2006.
61. Williams, L., & Cockburn, A., (2003). Agile Software Development: It's about Feedback and Change. *IEEE Computer*, 36 (6), 39–43.
62. Yu, B. L., Wooi, K. L., Wai, Y. T., & Soo F. T., (2012). Software Development Life Cycle AGILE vs Tradicional Approaches. *IPCSIT*, 37(2012), 162–167.
63. Zakon o davčnem postopku. *Uradni list RS* št. 13/2011-UPB1, 32/2012, 94/2012, 101/2013-ZDavNepr, 111/2013, 25/2014-ZFU, 40/2014-ZIN-B, 90/2014, 91/2015.
64. Zakon o davku na dodano vrednost. *Uradni list RS* št. 13/2011-UPB3, 18/2011, 78/2011, 38/2012, 83/2012, 86/2014, 90/2015.
65. Zakon o elektronskem poslovanju in elektronskem podpisu. *Uradni list RS* št. 98/2004-UPB1, 61/2006-ZEPT, 46/2014.
66. Zakon o opravljanju plačilnih storitev za proračunske uporabnike. *Uradni list RS* št. 59/2010, 111/2013.

67. Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih. *Uradni list RS* št. 30/2006, 51/2014.
68. Zhang, W., Johnson, T., Seltzer, T., & Bichard, S. L. (2010). The revolution will be networked: The influence of social networking sites on political attitudes and behaviour. *Social Science Computer Review*, 28(1), 75–92.
69. Zornada, L., Veselič, D. & Ježovnik, A. (2002). Razvoj informacijskega sistema – od strateškega načrta do realizacije. *Management, kakovost, razvoj: zbornik 2. Strokovnega posveta Visoke šole za management v Kopru z mednarodno udeležbo, Bernardin, 16. – 17. november 2001* (str. 223–236). Koper: Visoka šola za management.