

**UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA**

MAGISTRSKO DELO

**RAZVOJ SEIZMOLOŠKEGA INFORMACIJSKEGA
SISTEMA**

Ljubljana, januar 2004

TAMARA JESENKO

IZJAVA

Študentka Tamara Jesenko izjavljam, da sem avtorica tega magistrskega dela, ki sem ga napisala pod mentorstvom prof. dr. Damij Taliba in skladno s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega dela na fakultetnih spletnih straneh.

V Ljubljani, 5.1.2004

Podpis: _____

KAZALO

1.	Uvod.....	1
1.1.	Cilji dela	2
1.2.	Metode dela	2
1.3.	Struktura dela	3
2.	Načrtovanje informacijskih sistemov.....	4
2.1.	Življenjski cikel informacijskega sistema	6
2.2.	Izbor metodologije	11
2.2.1.	Filozofija	11
2.2.2.	Model	15
2.2.3.	Tehnike in orodja	15
2.2.4.	Obseg.....	15
2.2.5.	Izhodi.....	15
2.2.6.	Izkušnje	15
2.2.7.	Produkt	16
3.	Prenova poslovnih procesov.....	17
3.1.	Planiranje.....	17
3.2.	Izhodišča za prenovo poslovnih procesov.....	17
3.3.	Prenova poslovnih procesov.....	18
3.4.	Postopek prenove	20
4.	Objektno orientiran pristop razvoja IS	24
4.1.	Značilnosti objektnega razvoja.....	24
4.2.	Koncepti objektnega pristopa.....	25
4.2.1.	Objekt	26
4.2.2.	Razred.....	26
4.2.3.	Atribut	27
4.2.4.	Operacije	27
4.2.5.	Ograjevanje	27
4.2.6.	Mnogoličnost.....	28
4.2.7.	Sporočilo	28
4.2.8.	Povezave med razredi.....	29
4.2.9.	Dedovanje.....	30
5.	Objektno orientirane metodologije.....	31
5.1.	Poenoten jezik za moduliranje UML	31
5.1.1.	Osnovni gradniki	32
5.1.1.1.	Diagram primerov uporabe	35
5.1.1.2.	Interakcijski diagram.....	37
5.1.1.3.	Diagram aktivnosti	38
5.1.1.4.	Diagram stanj	39
5.1.1.5.	Diagram objektov	40
5.1.1.6.	Diagram razredov	40

5.1.1.7.	Diagram komponent	40
5.1.1.8.	Diagram porazdelitve	41
5.1.2.	UML mehanizmi	41
5.1.3.	Arhitektura	42
5.1.4.	Poenoten proces.....	43
5.2.	Tabelarni razvoj aplikacij.....	47
5.2.1.	Definicija problema.....	47
5.2.2.	Analiza delovanja sistema.....	49
5.2.2.1.	Aktivnosti	49
5.2.2.2.	Naloge	50
5.2.2.3.	Delovni in poslovni procesi.....	51
5.2.3.	Prenova poslovnih procesov.....	51
5.2.3.1.	Analiza in prenova tabel.....	52
5.2.3.2.	Procesni model	52
5.2.4.	Razvoj objektnega modela	53
5.2.5.	Načrt sistema	54
5.2.6.	Implementacija sistema	54
6.	Razvoj seizmološkega informacijskega sistema	55
6.1.	Predstavitev Urada za seizmologijo	55
6.1.1.	Potresi.....	55
6.1.2.	Urad za Seizmologijo	56
6.1.3.	Opis problema	57
6.1.4.	Opredelitev osnovnih pojmov	60
6.2.	Definicija problema.....	62
6.3.	Analiza delovanja sistema.....	63
6.4.	Prenova poslovnih procesov.....	68
6.5.	Objektni model sistema.....	74
6.6.	Načrtovanje sistema	78
6.7.	Implementacija	83
7.	Sklep.....	84
8.	Literatura in viri	86
8.1.	Literatura	86
8.2.	Viri	89
Priloga		I

SEZNAM SLIK

Slika 1: Tradicionalni kaskadni model.....	7
Slika 2: Neupoštevanje potreb vodstva	9
Slika 3: Kriteriji za oceno metodologije	12
Slika 4: Atomska struktura organizacije	20
Slika 5: Vhodni elementi prenove procesov	21
Slika 6: Stopnje izboljševanja poslovnega procesa.....	23
Slika 7: Neposredna preslikava realnosti v objekte	24
Slika 8: Potres v Mehiki kot objekt, njegova stanja in obnašanje.....	26
Slika 9: Razred potresov	26
Slika 10: Ograjevanje	28
Slika 11: Tok sporočil med objekti	29
Slika 12: Primer generalizacije	30
Slika 13: Primer specializacije	30
Slika 14: UML gradniki	32
Slika 15: Primer predstavitve primera uporabe.....	36
Slika 16: Primer diagrama primerov uporabe	37
Slika 17: Primer diagrama zaporedij	38
Slika 18: Primer diagrama sodelovanja.....	38
Slika 19: Primer diagrama aktivnosti	39
Slika 20: Primer diagrama stanj	39
Slika 21: Primer objektnega diagrama	40
Slika 22: Primer diagrama razredov	40
Slika 23: Primer diagrama komponent (a) in diagrama porazdelitve (b).....	41
Slika 24: UML mehanizmi	41
Slika 25: '4+1 pogled' arhitekture.....	43
Slika 26: Faze in aktivnosti poenotenega procesa.....	44
Slika 27: Zaporedje aktivnosti iteracije.....	46
Slika 28: Porazdelitev najmočnejših potresov leta 2002 na Zemlji, njihova magnituda in globina sta podani z velikostjo in barvo krožca.	55
Slika 29: Primer zapisov potresa (Turčija, 1.5.2003 ob 00:27UTC) z označenimi vstopi skupine primarnih (P) in sekundarnih (S) valov.....	56
Slika 30: Organizacijska struktura Urada za seizmologijo.	57
Slika 31: Državna mreža opazovalnic.....	58
Slika 32: Primer izračunane lokacije potresa.	61
Slika 33: Procesni model (1/3)	73
Slika 34: Primer podatkov povezanih z dokumentom <i>Vprašalnik</i>	74
Slika 35: Funkcionalne odvisnosti med atributi dokumenta <i>Vprašalnik</i>	75
Slika 36: Asociacije med identifikacijskimi atributi	76
Slika 37: Integracija analize	77
Slika 38: Razredi in njihove medsebojne povezave.....	77

Slika 39: Začetni objektni model	79
Slika 40: Končni objektni model.....	80

SEZNAM TABEL

Tabela 1: Vrednosti atributov posameznega objekta razreda POTRES.....	27
Tabela 2: Vrste UML diagramov	35
Tabela 3: Primer tabele entitet	48
Tabela 4: Prikaz dela tabele aktivnosti.....	50
Tabela 5: Del tabele nalog.....	51
Tabela 6: Pomen podatkov o lokaciji potresa predstavljeni na sliki 32	61
Tabela 7: Tabela entitet	63
Tabela 8: Tabela aktivnosti - obstoječe stanje (1/2).....	65
Tabela 9: Tabela nalog - obstoječe stanje (1/3).....	66
Tabela 10: Tabela prikazuje povezanost aktivnosti z delovnimi in poslovnimi procesi.....	67
Tabela 11: Prenovljena tabela entitet	69
Tabela 12: Prenovljena tabela aktivnosti (1/2).....	70
Tabela 13: Prenovljena tabela delovnih in poslovnih procesov	71
Tabela 14: Prenovljena tabela nalog (1/3).....	72
Tabela 15: Tabela operacij	81
Tabela 16: Aplikacijski model	82

1. Uvod

Vsako leto zatrese Zemljo več stotisoč potresov, ki presegajo magnitudo 2,0 in katerih žarišča so razporejena predvsem na stikih večjih geotektonskih plošč. Potresno najdejavnejši območji sta obtihoceanski in sredozemsko-himalajski pas, ki vključuje tudi naše kraje. Večina potresov je šibkih in ne povzročajo gmotne škode ter ne zahtevajo človeških življenj. Dejavnost seizmologov že dolgo ni omejena le na štetje potresov. Za vsak potres se, če je to le mogoče, izračuna njegova lokacija in čas (Za tak izračun potrebujemo vstopne čase primarnih potresnih valov s treh opazovalnic.). Kadar je na voljo dovolj kvalitetnih podatkov, se določi tudi magnituda, ki je mera za sproščeno energijo ob potresu, in intenziteta, ki nam poda oceno učinkov potresa na ljudi, zgradbe in naravo. Težko bi rekli, da se potresna aktivnost prelomov, ki potekajo po ozemlju Slovenije, povečuje. Dejstvo je, da je tehnološki razvoj omogočil izdelavo vse boljših in natančnejših senzorjev, ki zaznajo tudi najšibkejše potrese, posledica česar je, da se je število zabeleženih potresov povečalo.

Po velikonočnem potresu leta 1998, ki je razgalil kar precej pomanjkljivosti v organiziranosti slovenske seizmološke službe, je Vlada Republike Slovenije odobrila projekt posodobitve slovenske državne mreže potresnih opazovalnic. Do ukinitve 27. aprila 2001 je seizmološko službo izvajala Uprava RS za geofiziko, nato so njene naloge prešle v pristojnost Agencije Republike Slovenije za okolje, urada za seizmologijo. Predvidena je postavitev petindvajsetih opazovalnic, ki bi enakomerno pokrivalo slovensko ozemlje, kar pomeni, da se bo število potresov, ki jih bodo opazovalnice zabeležile in jim bo moč določili njihove osnovne parametre (čas, lokacija, magnituda, intenziteta) temu primerno povečalo. Hkrati s tem se je pojavila tudi težnja po izgradnji informacijskega sistema, ki bi seizmologom olajšal delo in v katerem bi bili zbrani parametrični podatki o potresih.

V preteklosti so se podatki o potresih hranili predvsem v papirnati obliki (katalog potresov), nato so se, odvisno od uporabljenih programov, vpisovali v računalnik in se arhivirali na različnih koncih, v različnih medijih (diskete, CD-ji, osebni računalniki,...) in v različnih oblikah. Posledice takšnega delovanja so, da se kaj kmalu izgubijo pomembni podatki (kot na primer podatek o točnosti ure na merilniku, ki je zabeležil dogodek) in zato postanejo izmerjene vrednosti neuporabne. Dejstvo, da so podatki hranjeni na različnih krajih in v različnih oblikah, pa tudi otežkoča nadaljnje analize in raziskave, saj je potrebno precej časa le za to, da se jih zbere in preoblikuje v tak format, kot ga potrebujemo. Preveč je ročnega brskanja in zbiranja ustnih informacij, ki sodelujejo v procesu, ter podvajanja del, saj so ti podatki shranjeni v različnih lokalnih bazah.

Rešitev za obstoječe stanje je razvoj seizmološkega informacijskega sistema, ki bi v vsakem trenutku omogočal pridobivanje kvalitetnih podatkov o:

- delovanju potresnih opazovalnic,

- lokaciji in času potresa,
- magnitudi in intenziteti potresa,
- kvaliteti podatkov.

Omogočal bi hitrejšo izdelavo tedenskih in mesečnih biltenov, ki se pošiljajo v mednarodne seizmološke centre, ter lažjo pripravo mnenj o tem, ali je nek potres lahko povzročil določene poškodbe na objektih, saj bi bili vsi podatki zbrani na enem mestu in dosegljivi na enak način. Možna bi bila tudi sledljivost, na primer kako se je lokacija potresa spreminjala v odvisnosti od števila podatkov, ki so bili uporabljeni pri njenem izračunu.

1.1. Cilji dela

Cilji, ki jih želim doseči v okviru naloge in izhajajo iz njenega namena so:

- Predstaviti osnove razvoja informacijskih sistemov.
- Predstaviti osnovne gradnike objektno-orientiranega modeliranja in Poenoten jezik za modeliranje UML skupaj s Poenotenim procesom kot najbolj razširjeno objektno orientirano metodologijo.
- Podrobno predstaviti metodologijo Tabelarni razvoj aplikacij.
- Definirati in analizirati delovne in poslovne procese in izdelati logični načrt informacijskega sistema z uporabo metodologije Tabelarni razvoj aplikacij.
- Raziskati in predlagati morebitne spremembe delovnih procesov.

1.2. Metode dela

Magistrsko delo bo zastavljeno tako, da bom na praktičnem primeru načrtovanja seizmološkega informacijskega sistema, ki bo namenjen zbiranju, obdelavi, vrednotenju in arhiviranju podatkov o potresih in potresnih opazovalnicah, uporabila objektno orientirano metodologijo Tabelarni razvoj aplikacij. Delo bo osredotočeno predvsem na fazo analize, implementacija v njem ne bo zajeta. Še pred začetkom projekta želimo v Uradu za seizmologijo sami narediti analizo poslovanja in definirali za nas pomembne delovne in poslovne procese. Na podlagi dobljenih rezultatov se bomo lažje odločili, kaj natančno potrebujemo in ali se bomo zadeve lotili sami ali pa bomo iskali zunanje izvajalce.

Pri načrtovanju tega sistema je potrebno poznati tudi osnove seizmologije in zahteve mednarodnih seizmoloških centrov pri izmenjavi podatkov. Pri popisu aktivnosti, opisu nalog posameznega delovnega procesa in predlogu morebitnih sprememb bom uporabila praktične izkušnje, ki sem jih pridobila z aktivnim sodelovanjem v tem procesu in skrbjo za pripravo končnih biltenov in izmenjavo podatkov s tujimi seizmološkimi centri.

1.3. Struktura dela

Delo je razdeljeno na sedem poglavij. V *prvem*, v Uvodu, je navedena obravnavana problematika, zastavljeni cilji, namen dela in uporabljene metode dela. Podana je tudi struktura dela.

V *drugem poglavju* je na splošno opredeljeno načrtovanje informacijskih sistemov. Predstavljene so faze tradicionalnega življenjskega cikla sistema, omenjene nekatere pomanjkljivosti le-tega in kriteriji, ki jih moramo upoštevati pri izbiranju primerne metodologije.

V *tretjem poglavju* se dotaknemo področja planiranja in prenove poslovnih procesov.

Četrto poglavje je namenjeno opisu Objektno orientiranega pristopa. Podane so njegove značilnosti in koncepti, na katerih temelji objektno orientirano modeliranje.

V *petem poglavju* sta predstavljeni dve objektno orientirani metodologiji: Poenoten jezik za modeliranje UML skupaj s Poenotenim procesom in metodologija »Tabularni razvoj aplikacij« (TAD).

V *šestem poglavju* uporabimo metodologijo TAD na praktičnem primeru, to je razvoju seizmološkega informacijskega sistema. Na kratko je predstavljeno tudi delo urada, nekateri pojmi, ki se nanašajo na potrebe in njihova opazovanja, ter opis obravnavanega problema. Delo je osredotočeno na prvih pet faz metodologije, predvsem na fazo analize delovanja sistema, implementacija pa ni izvedena.

V *sedmem poglavju* so navedene sklepne ugotovitve.

Na koncu dela se nahaja še seznam uporabljene literature in virov, delo pa zaključuje priloga.

Osnovna hipoteza dela je, da je možno z uporabo metodologije TAD popisati cilje Urada za seizmologijo, na podlagi ciljev definirati in pregledno popisati poslovne in delovne procese seizmološkega informacijskega sistema, predlagati spremembe procesov ter pripraviti logični model sistema.

2. Načrtovanje informacijskih sistemov

V današnjem času si skoraj ne moremo predstavljati poslovanja organizacij brez uporabe informacijskih sistemov¹. Že dolgo ti ne služijo samo izvajanju transakcij, hranjenju in obdelavi podatkov, ampak lahko z njimi pridobimo veliko več. Omogočajo oblikovanje in pripravo informacij o poslovanju in za odločanje, pridobivanju konkurenčne prednosti ter vključujejo uporabo sodobnih informacijskih orodij, intraneta, elektronskega poslovanja in upravljanja znanja.

In kaj pravzaprav je *informacijski sistem*? Je sistem, v katerem se ustvarjajo, shranjujejo in pretakajo informacije. Ni le sistem računalnikov, ampak vključuje tudi uporabnike, ki jih pri svojem delu uporabljajo. Je zbirka ljudi, strojev in aktivnosti, ki omogoča obdelavo podatkov ter generiranje informacij na način, ki zagotavlja popolnitev informacijskih zahtev organizacije (T. Damij, 1994, str. 1).

Informacijski sistem je sistem, v katerem se zbirajo, shranjujejo, procesirajo in generirajo informacije, ki so pomembne za organizacijo in so dostopne vsem, ki jih želijo uporabiti. Je sistem, v katerega so poleg ljudi lahko vključeni tudi sistemi računalnikov (Buckingham v Avison, Fitzgerald, 1996, str. 12).

In kaj predstavlja termin *informacija*? Z gradnjo informacijskih sistemov želimo doseči določene cilje. Eden osnovnih je, da spremenimo podatke v informacije in znanje² (Turban, MacLean, Wetherbe, 1999, str. 45):

- *Podatek* je elementarni opis stvari, dogodka, aktivnosti, ki je zabeležen in shranjen, ni pa organiziran tako, da bi izražal določen pomen. Podatki so lahko številke, črke, slike, zvoki, ...
- *Informacija* ni zgolj podatek, ampak ima za uporabnika nek pomen in uporabno vrednost. Mora biti točna, pravočasna in uporabniku na voljo v trenutku, ko jo ta potrebuje, ter mu mora posredovati odgovore na njegova zastavljena vprašanja.
- *Znanje* je sestavljeno iz podatkov ali informacij, ki so organizirani tako, da izražajo razumevanje in izkušnje, pridobljene iz v preteklosti že rešenih problemov, ki jih lahko uporabimo na obstoječih problemih ali aktivnostih.

Informacijski sistemi vplivajo na izvajanje temeljne dejavnosti organizacije ter posredno usklajujejo njeno delovanje z omogočanjem boljšega načrtovanja in nadzorovanja ter večje povezanosti podsistemov. Informacijski sistem je lahko le podsistem ali del sistema, ki omogoča izvajanje in opravljanje temeljne dejavnosti organizacije in s tem doseganje njenih ciljev (Gradišar, Resinovič, 1996, str. 61-66).

¹ Sistem je množica objektov, ki vplivajo eden na drugega (Embley et al, 1992, str. 1).

² angleško: knowledge

Koristi informacijskih sistemov so (Gradišar, Resinovič, 1996, str. 308):

- neposredne: povečanje učinkovitosti, lažje in hitrejše delo, krajšanje odzivnega časa, zmanjšanje števila napak v podatkih, hitrejša komunikacija in
- posredne: boljše usklajevanje delovanja, boljši nadzor, kakovostnejši podatki za odločanje, možnost oblikovanja alternativ, možnost hitrega odziva v nepričakovanih situacijah.

Informacijske sisteme lahko razvrstimo na več načinov: glede na organizacijsko strukturo, funkcijsko področje, sistemsko arhitekturo ali podporo, ki jo nudi. Zaradi pomembnosti, ki jih imajo v moderni organizaciji, mora biti njihovo načrtovanje skrbno planirano.

V petdesetih in šestdesetih letih, v obdobju zametkov informacijskih sistemov, še ni bilo splošno formaliziranih postopkov razvoja informacijskih sistemov. Ponavadi so jih razvijali programerji, ki se niso poglobili v potrebe uporabnikov, ampak je njihovo delo temeljilo na tehnološkem vidiku. Lastnosti večine takratnih informacijskih sistemov so bile: slaba dokumentiranost, nezadostno upoštevanje potreb uporabnikov, veliki stroški, niso bili planirani na dolgi rok, težko jih je bilo vzdrževati, prenavljati,... Podjetja so bila odvisna od nekaj programerjev, ki so poznali strukturo sistemov. Zaradi slabe dokumentiranosti in ne uporabe poenotenih tehnik in praks je bilo težko uvajati novosti. Pogosto je bilo potrebno, tudi pri vpeljavi zelo majhnih sprememb na zahtevo uporabnikov, napisati uporabljene programe ponovno vse od začetka. Postopno so postajali sistemi kompleksnejši, omenjeni problemi pa so se stopnjevali. Kot odziv nanje se je pojavilo prepričanje o pomembnosti delitve postopka izgradnje informacijskega sistema. Povečala se je zavest, da je za izgradnjo dobrega informacijskega sistema pomembna tudi njegova analiza in skrbno načrtovanje³, kar je privedlo do pojavnosti formaliziranih metodologij namenjenih razvoju računalniških aplikacij (Avison, Fitzgerald, 1996, str. 18).

Metodologija je skupek postopkov, tehnik, orodij in dokumentacijskih pripomočkov, ki jih uporabljajo razvijalci sistema⁴ pri načrtovanju in implementiranju informacijskega sistema. Sestoji iz faz, te iz podfaz⁵, ki vodijo razvijalce sistema pri izbiranju primernih tehnik v vsaki fazi projekta. Pomagajo jim tudi pri načrtovanju, upravljanju, kontroliranju in vrednotenju projektov izgradnje informacijskih sistemov (Avison, Fitzgerald, 1996, str. 10).

Metodologije se med seboj razlikujejo v tehnikah, ki jih uporabljajo, in po vsebini posamezne faze. Lahko se razlikujejo tudi že v svojih osnovah. Nekatere poudarjajo človeški pogled na razvoj informacijskega sistema, druge pristopajo k temu na bolj znanstven način, tretje spet želijo razvoj čim bolj avtomatizirati. Katero bomo izbrali, je odvisno od naših zahtev in ciljev, ki naj bi se jih z uporabo določene metodologije doseglo (Avison, Fitzgerald, 1996, str. 11):

³ angleško: design

⁴ angleško: system developer

⁵ angleško: sub-phase

- Natančno beleženje zahtev informacijskega sistema: Metodologija naj bi bila v pomoč uporabnikom pri lažji določitvi lastnih zahtev, razvijalci sistema pa bi jih z njo preučili in analizirali.
- Sistematičen razvoj informacijskega sistema: Metodologija naj bi omogočala sistematičen razvoj in učinkovit nadzor napredka. Pri projektih velikega obsega morajo biti namreč faze izgradnje zelo dobro definirane in kontrolirane, če želimo, da bo izgradnja informacijskega sistema uspešna.
- Čim krajši čas izgradnje in nizki oz. sprejemljivi stroški: Metodologija naj bi omogočala razvoj informacijskega sistema v določenem časovnem obdobju in sprejemljivo višino vloženih finančnih sredstev.
- Dobro dokumentiran sistem z enostavnim vzdrževanjem: Metodologija naj bi omogočala razvoj sistema, ki bi bil dobro dokumentiran in enostaven za vzdrževanje. Take zahteve postavimo pri sistemih, ki se bodo v prihodnosti še spreminjali, saj želimo, da spremembe čim manj vplivajo na preostale dele sistema.
- Čim zgodnejša vključitev sprememb v procesu izgradnje sistema: Metodologija naj bi omogočala čim zgodnejšo vključitev sprememb, saj strošek sprememb narašča, ko se izgradnja informacijskega sistema približuje svoji implementaciji.
- Sistem je všeč ljudem, na katere vpliva: Metodologija naj bi omogočala razvoj sistema, ki bi bil všeč ljudem, na katere vpliva, saj bo tako več možnosti, da bo njegova uporaba uspešna in učinkovita.

Če želimo razviti dober informacijskih sistemov, se moramo dela lotiti sistematično. In kaj je dober informacijski sistem? Z vidika uporabnika je to sistem, ki daje kvalitetne in pravočasne rezultate, je zanesljiv, je enostaven za uporabo, je učinkovit,... Z vidika razvijalcev in vzdrževalcev sistema pa je to sistem, ki se ga da na enostaven način spremeniti ali razširiti, je razumljiv, sestavljen iz delov, ki se lahko ponovno uporabijo, se ga da enostavno testirati, primerjati z drugimi sistemi,...(Jacobson, 1996, str. 69).

2.1. Življenjski cikel informacijskega sistema

Kot posledica problemov, ki so bili omenjeni v prvem delu drugega poglavja, se je konec šestdesetih pojavila metodologija, imenovana **tradicionalni življenjski cikel informacijskega sistema**⁶, ki predstavlja izhodišče mnogim kasneje razvitim metodologijam. Služi kot ogrodje za planiranje, izvajanje in kontrolo razvoja projekta načrtovanja informacijskega sistema in je sestavljena iz naslednjih korakov (Avison, Fitzgerald, 1996, str. 20):

- Študije izvedljivosti⁷,

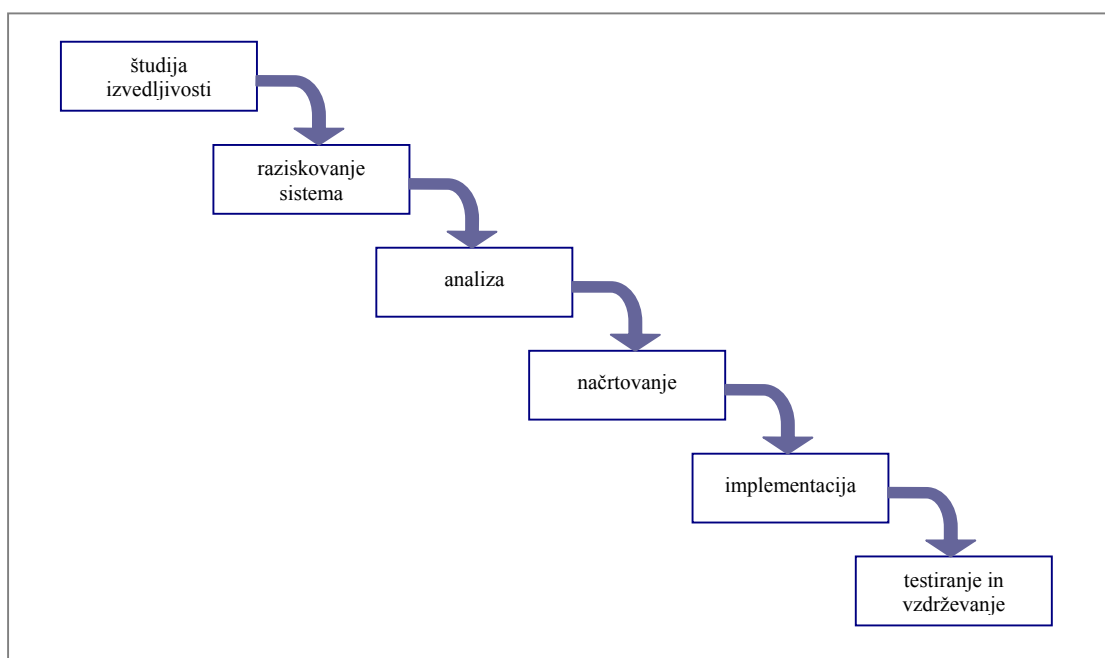
⁶ V literaturi je tradicionalen življenjski cikel različno poimenovan: npr. običajen življenjski cikel, tradicionalni življenjski cikel, kaskadni model, življenjski cikel razvoja sistema,... (Avison, Fitzgerald, 1996, str. 20).

⁷ angleško: feasibility study

- raziskovanja sistema⁸,
- analize,
- načrtovanja⁹,
- implementacije in
- testiranja in vzdrževanja.

Predstavljena je s kaskadnim modelom¹⁰, s katerim je ponazorjen tok procesa. Različni avtorji sicer različno definirajo in poimenujejo posamezne faze življenjskega cikla, bistvo in vrstni red pa ostajata ista.

Slika 1: Tradicionalni kaskadni model



Prirejeno po Turban, MacLean, Wetherbe, 1999, str. 605

V prvem koraku, **študiji izvedljivosti**, se pregleda obstoječi sistem, razišče nove zahteve in probleme, ki bi se lahko pojavili, pripravi možne alternativne rešitve ter opredeli stroške izvedbe in koristi posamezne rešitve. Za vsako možno rešitev je potrebno predvideti njene omejitve, predvsem tiste, ki so povezane s potrebnimi viri. Vse predlagane rešitve morajo biti izvedljive s tehnološkega, ekonomskega, socialnega in organizacijskega vidika. Izbere se najboljšo ter se poročilo o tem preda vodstvu, ki se na koncu odloči, če so priporočila analitika sprejemljiva. Če vodstvo projekt odobri, se preide na naslednji korak.

V drugem koraku, **raziskovanju sistema**, se pripravi funkcionalna specifikacija sistema, analizirajo se zahteve uporabnikov, opredelijo se tipi in količina podatkov, omejitve in

⁸ angleško: system investigation

⁹ angleško: design

¹⁰ angleško: waterfall model

problemi obstoječega sistema ter možne izjemne okoliščine¹¹. Informacije, zbrane v drugi fazi, morajo biti bolj izčrpne, kot tiste, ki so bile podane v fazi izvedljivosti. V ta namen se organizirajo intervjuji s predstavniki vodstva in operativnega kadra, pripravijo se različni vprašalniki, opazuje se način dela, se vzorči in pregleda rezultate ter drugo dokumentirano gradivo (Avison, Fitzgerald, 1996, str. 22):

- Z opazovanjem dela se pridobi vpogled v probleme, delovne pogoje, ozka grla in uporabljene metode dela.
- Intervjuvanje posameznikov ali skupin uporabnikov je najbolj uporabna tehnika pridobivanja in potrjevanja informacij. Hkrati se nam ponuja možnost, da preko razgovora z uporabniki odpravimo morebitni odpor pred spremembami.
- S pregledovanjem dokumentacije in rezultatov se lahko izluščijo obstoječi problemi, vendar se mora analitik zavedati, da je lahko dokumentacija že zastarela.
- Vzorčenje je uporabna tehnika, vendar ponavadi zahteva dodatna strokovna znanja (na primer s področja statistike).

Rezultati se predstavijo s pomočjo različnih diagramov, tabel in drugih dokumentov. Z diagrami se prikaže pretok dokumentov skozi oddelek, procese, organizacijsko strukturo posameznega oddelka ali cele organizacije, posamezne komponente sistema in interakcije med njimi. V dokumentacijo se lahko vključijo tudi zapiski, ki so bili narejeni med intervjuji.

Tretja faza, **analiza sistema**, temelji na rezultatih predhodne faze. Namenjena je izdelavi logičnega modela podatkov in procesov sistema, ki bo kasneje uporabljen pri načrtovanju sistema. Dognati je treba, zakaj obstajajo določeni problemi, če obstajajo alternativne metode, razumeti vzroke, zakaj so bile uporabljene nekatere metode, kakšna je predvidena rast podatkov (Avison, Fitzgerald, 1996, str. 23).

V četrtem koraku, v fazi **načrtovanja sistema**, se logični model prevede v načrt bodočega sistema. Dokumentira se vhodne in izhodne podatke sistema, njihovo strukturo, načine zajemanja podatkov in procese, ki pretvorijo podatke iz vhodnih v izhodne. Pri delu si pomagamo z uporabo različnih diagramov in tabelarnih predstavitev. Izdela se načrt tako strojne kot programske opreme in pripraviti načrte za implementacijo in testiranje sistema.

Faza **implementacije** je namenjena gradnji sistema. Če so bili pri načrtovanju predvideni računalniški programi, morajo biti sedaj ti kodirani in preverjeni. Nakupi in namesti¹² se potrebna strojna in programska oprema ter preveri, če ustreza zahtevam. Na koncu faze se izvede prehod iz starega na novi sistem. Pomemben element te faze je tudi izobraževanje osebja, ki uporablja sistem. Potrebno se je izogniti situacijam, da bi prišlo do napak zaradi

¹¹ angleško: exceptional condition

¹² angleško: install

neustreznega poznavanja uporabe sistema. Poskrbeti je treba tudi za varnost sistema in možnost restavriranja podatkov.

Fazi implementacije, ko sistem že deluje, sledi še **testiranje in vzdrževanje**. Zagotoviti je treba učinkovito in kontinuirano delovanje sistema. Sproti je treba odpravljati napake, ki se pojavljajo, in prilagajati sistem novim zahtevam. Pregledati je treba, če s sistemom dosegamo zastavljene cilje. V primeru, da sistem ne podpira več ustreznih poslovnih procesov, se njegov življenjski cikel zaključi in s *študijo izvedljivosti* prične življenjski cikel novega informacijskega sistema.

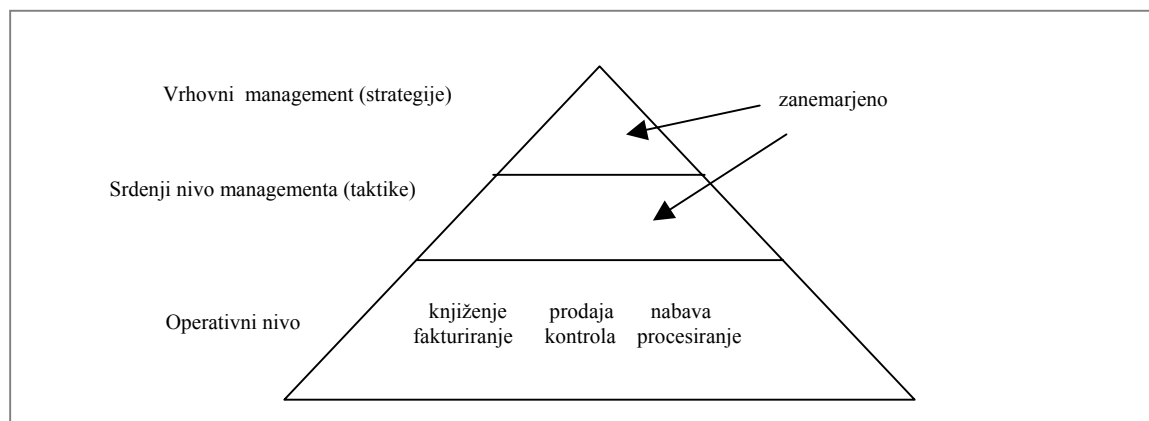
Pojav metodologije »tradicionalni življenjski cikel sistema« je pomenil velik napredek na področju razvoja informacijskih sistemov, saj ta vsebuje vse pričakovane sestavne dele metodologije (Avison, Fitzgerald, 1996, str. 29):

- Sestavljena je iz niza faz, ki se prične z raziskavo izvedljivosti projekta in konča s fazo pregleda in vzdrževanja. Vsaka faza je sestavljena iz podfaz in je dokumentirana.
- Vsaka naslednja faza se prične s formalno odobritvijo predhodne faze.
- Ocenijo se stroški in koristi posamezne rešitve.
- Vključena so različna orodja za kontrolo projekta.

Pojavile pa so se tudi številne kritike. V današnjem času, času hitrega spreminjanja in hude konkurence, je težko vnaprej napovedati vse informacijske potrebe, zahteve definirane na začetku življenjskega cikla se lahko, zaradi prilagajanj izzivom okolja in konkurence, tekom procesa bistveno spremenijo. Glavni očitki tradicionalnemu modelu življenjskega cikla informacijskega cikla sistema so (Avison, Fitzgerald, 1996, str. 30):

- Neupoštevanje potreb managerjev podjetij. Sistemi ne nudijo podpore vodstvu pri odločanju in ne vključujejo taktičnih in strateških ciljev.

Slika 2: Neupoštevanje potreb vodstva



Vir: Avison, Fitzgerald, 1996, str. 31

- Nezadovoljstvo končnih uporabnikov. Težko je napovedati vse zahteve sistema že v začetni fazi razvoja. Čas razvoja kompleksnejših informacijskih sistemov z dosledno uporabo tradicionalne metodologije se lahko meri v letih. Posledica tega je, da v današnjem času hitrega spreminjanja poslovnega okolja implementirani sistem ne zadovoljuje uporabnikovih trenutnih potreb in je dejansko že pri implementaciji zastaran.
- Dokumentacija ni namenjena uporabniku ampak tistim, ki so vključeni v analizo, izdelavo sistema in programiranje. Vzporedno s tehnično dokumentacijo se ne pripravlja tudi dokumentacija namenjena končnemu uporabniku. Pogosto so v dokumentaciji izpuščeni popravki, ki so bili opravljeni v zadnji fazi cikla, to je v fazi testiranja in vzdrževanja.
- Neambiciozno načrtovanje in nepopolnost sistemov. Pogosto se avtomatizira le delo, ki je bilo prej opravljeno ročno, in se ne vpelje sistemov, ki bi pomenili bolj radikalne spremembe in prinesli dejanske koristi. Z računalniki je možno hitro in hkrati obdelati¹³ ogromne količine podatkov ob predpostavki, da je proces dobro strukturiran, stabilen in rutinski. Vendar se ne moremo izogniti izjemnim okoliščinam¹⁴. Te morajo biti predvidene že pri načrtovanju sistema, a so pogostokrat zanemarjene ali pozabljene, kar vodi k neuspešnosti in nepopolnosti sistema. Brez upoštevanja izjemnih okoliščin, ki se lahko pojavijo, analiza in načrtovanje sistema ne moreta biti popolna.
- Težko vzdrževanje obstoječih sistemov. Zaradi zaostankov pri razvoju in same dolgotrajnosti razvoja se poskuša v že obstoječe sisteme na hitro vgraditi najnujnejše zahteve, ki so zato pogosto slabo ali nepopolno načrtovane. Rezultat tega je sistem, mešanica starega in novega, ki postane neobvladljiv in ga je nemogoče vzdrževati.

Glavna pomanjkljivost in očitek tradicionalnemu življenjskemu ciklu je premajhno upoštevanje potreb vodstva in posledično s tem strateških ciljev organizacije. V ta namen se je pojavilo *strateško načrtovanje informacijskih sistemov*, ki v procesu razvoja enakovredno vključuje strateške, taktične in operativne cilje ter potrebe organizacije, saj le na ta način lahko zagotovimo konkurenčne prednosti podjetja in izoblikujemo njegovo tekmovalno strategijo.

Strateški razvoj informacijskih sistemov se mora ujemati s strateškimi načrti podjetja, njegovi glavni cilji pa so (Tomažič, Novaković, 2000, str. 162):

- Izboljšanje komunikacije med predstavniki vodstva in informatiki.
- Načrtovanje toka informacij in procesov.
- Povezava razvoja informacijskih sistemov s poslovno strategijo podjetja.

¹³ angleško: to process

¹⁴ angleško: exceptional conditions

- Zmanjšanje stroškov in potrebnega časa za izvedbo aplikacije.
- Učinkovita razporeditev človeških virov.
- Priprava vseh potrebnih izhodišč in predlaganje optimalnega zaporedja nadaljnjih korakov pri planiranju in razvoju informacijskega sistema.

Kot smo rekli na začetku podpoglavja, je postavila metodologija, imenovana tradicionalni življenjski cikel informacijskega sistema, temelj za razvoj formaliziranih metodologij. Da bi premostili omenjene njene pomanjkljivosti, so bile razvite in se še razvijajo nove metodologije in različni modeli življenjskega cikla sistema. Naivno je pričakovati, da bo določena metodologija optimalna oziroma idealna v vseh pogledih in bo rešila vse probleme. Ponavadi je posamezna metodologija osredotočena na odpravo ene ali nekaj pomanjkljivosti, kar se navezuje tudi na namen, zaradi katerega je bila izdelana. Medtem ko je določen pristop primeren za neko organizacijo, za drugo, zaradi njene edinstvenosti, ni. Ne glede na to pa mora dobra metodologija vključevati nekaj temeljnih izhodišč:

- čim večja vključitev uporabnikov v celoten proces razvoja sistema,
- čim krajši časa razvoja in
- sprotno odpravljanje ugotovljenih napak, vključevanje na novo ugotovljenih zahtev ter, če je potrebno, vračanje v predhodne faze razvoja.

2.2. Izbor metodologije

Obstaja nekaj metodologij in poraja se vprašanje, katero izbrati? Glavna vprašanja, na katere moramo dobiti odgovore so: Kaj nam posamezna metodologija ponuja, kakšne bodo posledice njene uporabe in ali jo bomo znali izkoristiti? Čeprav so si lahko metodologije med seboj podobne in so razlike izpostavljene le zaradi njihovega trženja, moramo pred izborom raziskati, katera bo najbolj ustrezala našemu obravnavanemu problemu. Avison in Fitzgerald (1996) okvirno predlagata sedmih osnovnih elementov, kriterijev, pri tem, da nekateri izmed njih vsebujejo več pod-elementov (glej sliko 3 na str. 12).

2.2.1. Filozofija

Prvi, zelo pomemben pogled na metodologijo, je vprašanje filozofije, na kateri temelji. Omenjen kontekst filozofije se nanaša na načela oziroma niz načel, na katerih temelji metodologija. Izbira področja, ki ga pokriva metodologija, usmerjenost ljudi, podatkov, sistemov, možni predsodki in drugi vidiki temeljijo na filozofiji metodologije. Kot vodilo razumevanja filozofije so vpeljeni štirje faktorji: pristop, cilji, področje in uporaba.

Pristop: V našem primeru je izraz pristopa vpeljan kot način razmišljanja o problemu. Ponavadi se ta ne veže na določen subjekt, ampak obravnava različne probleme ne glede na njihove specifične vsebine. Ločimo dva pristopa k analizi sistema: znanstveni in sistemski.

Znanstveni pristop temelji na krčenju kompleksnosti in variabilnosti realnega sveta na manjše enote, ki jih lažje obravnavamo in razložimo. Rezultati se ocenijo na podlagi njihove ponovljivosti, pridobljeno znanje pa se dopolnjuje na podlagi sprejetja ali ovržbe predlaganih hipotez. Pri prevelikem drobljenju se lahko hitro pozabi, da celoten sistem ni le sestavljanka posameznih delčkov. To je vodilo k razvoju sistemskega pristopa, ki obravnava sistem kot celoto z vsemi lastnostmi in notranjimi povezavami med deli celote.

Slika 3: Kriteriji za oceno metodologije

1. Filozofija
 - Pristop¹⁵
 - Cilji
 - Področje
 - Uporaba
2. Model
3. Tehnike in orodja
4. Obseg¹⁶
5. Izhodi
6. Izkušnje
 - Izvor¹⁷
 - Profil uporabnikov
 - Udeleženci
7. Produkt

Vir: Avison, Fitzgerald, 1996, str. 446

Cilji: Rdečo nit filozofije metodologije predstavljajo tudi njeni cilji. V grobem lahko določimo tri osnovne cilje: boljši končni izdelek, boljši proces razvoja in standardiziran proces razvoja (Avison, Fitzgerald, 1996, str. 421).

- **Boljši končni izdelek.** Z uporabo določene metodologije želimo izboljšati končni izdelek oziroma želimo dober informacijski sistem. Tega cilja ne smemo zamenjevati s kakovost procesa razvoja. Težko je ocenjevati kakovost končnega izdelka, ki bi ga dobili z uporabo neke metodologije, saj je možno, da bi nam uporaba druge, prinesla podoben rezultat. Ocenjevanje rezultatov je tudi zelo odvisno od človeškega faktorja, saj se lahko mnenja o izdelku razlikujejo od človeka do človeka. Vseeno pa lahko nanizamo nekaj lastnosti, ki jih pričakujemo od dobrega informacijskega sistema:

¹⁵ V literaturi je faktor vpeljan kot paradigma, vzorec ali zgled. Angleško: paradigm

¹⁶ angleško: scope

¹⁷ angleško: background

- *Sprejemljivost in razpoložljivost*: Uporabniki so zadovoljni z informacijskim sistemom, saj izpolnjuje njihove zahteve. Dostop do informacijskega sistema je zagotovljen takrat in tam, kjer je potreben.
- *Povezanost*¹⁸ in *združljivost*¹⁹: Zagotovljena je povezanost posameznih komponent (podsystemov) v celoto in združljivost z ostalimi sistemi, ki so del organizacije. Vendar pa mora biti povezanost posameznih komponent *šibka*²⁰, tako da preoblikovanje posamezne komponente ne vpliva na delovanje celotnega sistema.
- *Dokumentiranost*: Pomembnost obstoja dobre dokumentacije smo že večkrat poudarili. Z njo si pomagamo pri komunikaciji med operaterji, uporabniki in razvijalci sistema in vodstvom.
- *Enostavnost učenja*: Učenje novih uporabnikov implementiranih rešitev je enostavno in intuitivno.
- *Ekonomičnost*: Stroški in koristi informacijskega sistema so v okviru predvidenih virov in omejitev.
- *Učinkovitost in zmogljivost*²¹: Delovanje informacijskega sistema na najboljši možen način dosega poslovne in organizacijske cilje ter maksimalno izkorišča razpoložljive vire za doseganje konkurenčne prednosti.
- *Hitrost razvoja*: Metodologija omogoča hiter razvoj informacijskega sistema glede na njegovo velikost in kompleksnost.
- *Prilagodljivost*²²: Informacijski sistem se na enostaven način preoblikuje, se mu doda ali odvzame komponente.
- *Funkcionalnost in realizacija*²³: Informacijski sistem zadovoljuje zahteve in prehod s starega na nov sistem je izvedljiv s tehnološkega, ekonomskega, socialnega in organizacijskega vidika.
- *Vzdrževanje in prenosnost*²⁴: Vzdrževanje informativnega sistema je enostavno, brez večjih problemov se ga da prilagoditi novim zahtevam ter ni vezan le na določen tip opreme ali kraj izvedbe.
- *Robustnost in varnost*: Informacijski sistem je stabilen, dopušča možne napake in neustrezno, napačno uporabo²⁵.
- *Enostavnost in možnost testiranja*: Nejasnosti in kompleksnost sistema so minimizirane, možno je njegovo testiranje, kar vodi k odpravi operativnih napak in uporabnikovega nezadovoljstva.

¹⁸ angleško: cohesiveness

¹⁹ angleško: compatibility

²⁰ angleško: low coupling

²¹ angleško: efficiency

²² angleško: flexibility

²³ angleško: implementability

²⁴ angleško: portability

²⁵ angleško: misuse

- *Pravočasnost in vidljivost*²⁶: Informacijski sistem deluje uspešno v vseh možnih pogojih in nam zagotavlja zahtevane informacije takrat, ko so potrebne. Dopolnjuje možnost spremljanja in sledenja izvedbi določenih akcij.

Našteli smo nekaj lastnosti, ki jih pričakujemo od dobrega informacijskega sistema. Idealno bi bilo, če bi lahko optimizirali vse, a žal to ponavadi ni izvedljivo. Pri izbiranju metodologije, namenjene razvoju informacijskega sistema, je pomembno, da se odločimo za tisto, ki bo poudarila lastnosti sistema, ki so pomembne za obravnavani problem.

- ***Boljši proces razvoja.*** Informacijski sistem želimo razviti s pomočjo metodologije, ki bo zagotavljala natančno kontrolo razvojnega procesa in izhodnih produktov posamezne faze procesa. S tem izboljšamo kontrolo projekta. V ta namen so bili sprejeti številni industrijski standardi kakovosti (na primer BS5750/ISO9001), ki zagotavljajo kakovost procesov, ne pa tudi kakovost končnih izdelkov. Past, v katero se lahko hitro ujamemo, je, da delo prilagajamo standardom, ne glede na to, če bo končni izdelek dober.
- ***Standariziran proces razvoja.*** Tretji cilj temelji na prepričanju in koristi, ki jih prinaša pristop, ki je splošen in enak za celotno organizacijo. Rezultati tega pristopa so: enostaven prehod z enega projekta na naslednjega, ponovno izobraževanje ni potrebno, v že izvedenih projektih nabrane izkušnje in znanje. Vse to vodi tudi k enostavnejšemu vzdrževanju sistema.

Za izvedbo projekta bi bilo najbolje, če bi izbrana metodologija zagotavljala izpolnitev vseh treh ciljev. Dober in standariziran proces razvoja bi nas varno vodil na poti do dobrega končnega izdelka, to je dobrega informacijskega sistema.

Področje: Tretji faktor, povezan s filozofijo metodologije, je področje, ki mu je metodologija namenjena. Da bi izbrali primerno metodologijo, je potrebno dobro poznati strukturo organizacije, jo analizirati od zgoraj navzdol²⁷ in izpostaviti strateške zahteve poslovanja. Metodologija mora omogočati, da bo načrtovani informacijski sistem izpolnjeval temeljne zahteve.

Uporaba: Zadnji faktor filozofije je povezan s primernostjo in uporabljivostjo metodologije. Nekatere metodologije so izdelane za reševanje točno določenih problemov, v določenem okolju, tipu ali velikosti organizacije, druge pa so namenjene splošni uporabi.

²⁶ angleško: visibility

²⁷ angleško: top-down analysis

2.2.2. Model

Drugi kriterij se nanaša na model oziroma na modele, ki jih posamezna metodologija uporablja. Modele lahko razdelimo v štiri skupine: opisne, analitične, shematične in simulacijske. Uporaba shematičnih modelov je pomembna predvsem zaradi boljše komunikacije med uporabniki in analitiki. Z modeli prikažemo ali opišemo bistvo problema in načrtovanja na abstrakten način. Modeli so pripomoček za boljše razumevanja sistema, ki ga razvijamo. Z modeli vizualiziramo obstoječi ali želeni sistem, določimo strukturo in obnašanje sistema ter dokumentiramo odločitve, ki smo jih med procesom razvoja sistema sprejeli (Booch et al., 1994, str.6).

2.2.3. Tehnike in orodja

Tretji kriterij je ocenitev tehnik in orodij, ki jih posamezna metodologija ponuja in uporablja. Tehnike, uporabljene v posamezni metodologiji so ponavadi delo njihovih avtorjev. Obstajajo tudi take metodologije, ki ne temeljijo na določeni tehniki, ampak priporočajo uporabo najboljših tehnik, ki so v določenem času na voljo. Nekatere metodologije ne predpisujejo uporabe določenih orodij, medtem ko druge tako uporabo priporočajo. Sem spadajo enostavna orodja za risanje kot tudi orodja, ki omogočajo prikaz celotnega procesa razvoja sistema z uporabo prototipov, projektnega managementa, generiranjem programske kode, ...

2.2.4. Obseg

Četrti kriterij, ki nam je v pomoč pri izboru ustrezne metodologije, je njen obseg. Nanaša se na število in vrsto faz življenjskega cikla sistema, ki jih posamezna metodologija pokriva. Vključuje tudi analizo poglobljenosti in uporabnosti posamezne faze življenjskega cikla. Problem, ki se pojavlja pri ocenjevanju obsega posamezne metodologije, je, da ni splošno predpisanih faz, ki bi bile v življenjskem ciklu sistema obvezne.

2.2.5. Izhodi

Pomembno je poznati izhodne produkte posamezne faze metodologije, kot tudi naravo končnega izdelka. Izhodi so od faze do različni. Lahko predstavljajo specifikacijo izvršene analize sistema ali pa delujoč implementiran sistem.

2.2.6. Izkušnje

Za razumevanje posamezne metodologije moramo poznati tudi njeno izvor. V grobem jih lahko razdelimo v dve skupini (Avison, Fitzgerald, 1996, str. 424):

- metodologije, katerih razvoj temelji na praksi, in
- metodologije, katerih razvoj temelji na teoriji.

V prvo skupino spadajo metodologije, ki so nastale v organizacijah oziroma podjetjih kot posledica reševanja določenega lastnega problema, nato pa so jih prilagodili še za trženje in ponudili tudi drugim. Značilnost takšnih metodologij je, da so splošno znane in so bile že večkrat uporabljene. Poznamo rezultate in izkušnje tistih, ki so jo že uporabili ter se lahko na podlagi tega odločimo, če ustreza našim zahtevam. Njihove pomanjkljivosti pa lahko izvirajo ravno iz te »ozke usmerjenosti«, torej da so bile razvite na podlagi reševanja določenega problema in niso bile prvotno namenjene reševanju splošnih problemov. Z vsako uporabo take metodologije se izlušči še kakšna pomanjkljivost, ki ni bila upoštevana. Obstoječe metodologije se na osnovi novih spoznanj dopolnjujejo in vanje se vgrajujejo nova orodja, pojavljajo se vedno nove in nove metodologije, te postajajo vedno obširnejše in, ne redko, težko obvladljive. V želji, da bi metodologija primerna za vse uporabnike, se zakrije bistvena točka, zaradi katere je bila sploh razvita.

V drugi skupini imamo metodologije, katerih korenine izvirajo iz teorije. Te so ponavadi redkeje uporabljene, manj znane in so rezultat raziskovalnih projektov izvedenih na univerzah ali raziskovalnih institucijah. Bolj je poudarjen raziskovalni in teoretični vidik, kot komercialna uporaba, čeprav obravnavajo realne primere, ki jih je mogoče testirati in preizkusiti.

Poznati moramo tudi profil uporabnikov (število in tip uporabnikov, njihove izkušnje) sistema in vedeti, koga je potrebno pri uporabi posamezne metodologije vključiti (udeleženci). Zanima nas, če lahko metodologijo izvedejo uporabniki sami ali pa je potrebno sodelovanje profesionalnih analitikov.

2.2.7. Produkt

Zadnji kriterij pri oceni metodologije je njen končni produkt. Zanima nas, kaj bomo za vložena sredstva dobili. Produkt je lahko sestavljen iz programov, napisane dokumentacije, odobrenega določenega števila ur izobraževanja, ustrezne podpore in svetovanja, ...

Predlaganim kriterijem lahko dodamo tudi druge, take, ki se nam zdijo za obravnavan problem pomembni in nam bodo pomagali pri pravilni odločitvi:

- hitrost razvoja informacijskega sistema,
- obseg predloženih specifikacij in dokumentacije,
- možnost prilagajanja posebnim okoliščinam, ...

Na koncu je potrebno še enkrat poudariti, da nobena metodologija, uporabljena pri izgradnji ali prenovi informacijskega sistema, sama po sebi ne more zagotoviti uspeha, lahko pa s svojimi priporočili, opisi posameznih korakov, ciljev in tveganj zviša kakovost končnega rezultata ter zmanjša tveganja.

3. Prenova poslovnih procesov

3.1. Planiranje

Planiranje je prva funkcija managementa. Je zamišljanje ciljev, rezultatov in poti za njihovo doseganje. Je odločanje, je usklajevanje ciljev, poti in seveda posameznikov ali pa delov podjetja (Rozman et al, 1993, str. 72). Planiranje je aktivnost, namenjena določitvi namena organizacije, analizi okolja, v katerem se nahaja, določitvi ciljev in odločanju o aktivnostih za doseganje teh ciljev. Planiranje je kontinuiran proces, plan pa je pisni dokument, rezultat procesa, v katerem je zapisano, kaj mora biti storjeno in na kakšen način. Plani in rezultati se nato uporabijo za pripravo novih planov in proces planiranja se nadaljuje.

Ločimo strateške in taktične plane. Želenim stanjem podjetja v prihodnosti pravimo *cilji*. Za doseganje teh ciljev si mora podjetje zamisliti poti, ki ga pripeljejo iz sedanosti v prihodnost. Tako kot je mogočih več ciljev, obstoji vrsta poti, ki vodijo do njih. Ključnim, dolgoročnejšim potem pravimo *strategije*, odločitve na taki poti pa so *taktike* (Rozman et al., 1993, str. 83 - 84). Predstavljati si moramo, kaj želimo doseči, se zavedati obstoječega stanja in določiti stvari, ki jih je potrebno storiti, da dosežemo želeno stanje. Nato je potrebno te spremembe, ki nas bodo vodile po zastavljeni poti, izvršiti. *Strateški plani* so okvirni, dolgoročni plani. Z njimi ne predvidevamo poslovanja organizacije, ampak jo pripravimo na nepredvidljivo prihodnost. *Taktični plani* so kratkoročni ali operativni plani. Z njimi določimo, kaj je potrebno v tekočem letu storiti, da bomo delovanje organizacije približali njenim dolgoročnim ciljem.

Proces planiranja je pomemben za uspeh vsake organizacije, ne glede na to, ali je njena dejavnost usmerjena v ustvarjanje dobička ali le v zagotavljanje določenih storitev. Planiranje prinaša prednosti. Pomaga pri prilagajanju na okolje, sprejemanju odločitev o pomembnih stvareh, izkoriščanju ponujenih priložnosti. S planiranjem se pojavlja manj napak pri odločitvah in povečuje uspešnost in doseganje zastavljenih ciljev organizacije.

3.2. Izhodišča za prenovo poslovnih procesov

Stati pri miru dejansko pomeni nazadovati (Watson, 1994, str. 38). Edini način, da preprečimo nazadovanje, je, da proces obrnemo v nasprotno smer, v napredek. V današnjem času se zaradi hitrega tehnološkega in ekonomskega razvoja soočamo s stalno spreminjajočimi se pogoji poslovanja, ki silijo podjetja k spremembam. Zavedati se moramo, da vse ne prinesejo pozitivnih rezultatov. Poznamo pet trendov spreminjanja poslovanja (Watson, 1994, str. 39):

- poslovanje ostane nespremenjeno (vendar le za krajše časovno obdobje),
- postopoma narašča ali
- postopoma upada,

- ima trenuten uspeh in hiter, skokovit porast ali
- katastrofalen učinek, ki nas prisili k umiku.

Vzdrževanje standarda, kontinuirane izboljšave in nenadne²⁸ izboljšave so trije trendi, ki lahko vodijo k pozitivnih spremembam. *Vzdrževanje standarda storitev* je pomemben dejavnik poslovanja, saj si potrošniki želijo vedno kakovostnih in zanesljivih storitev ter produktov. Na ta način pridobimo njihovo zvestobo in si zagotovimo minimalne pogoje za poslovno rast. Vendar se z doseženim nivojem ne smemo zadovoljiti. Da bi preprečili konkurenci posnemanje, moramo dodati tudi strategiji napredka. *Kontinuiran proces izboljšav* je usmerjen k izpopolnjevanju poslovnih in delovnih procesov, kakovosti produktov, storitev in podpore ter s tem k povečevanju zadovoljstva potrošnikov. Primeri takega procesa so na primer: hiter odziv na probleme, ki se pojavijo, in njihovo preprečevanje, stalen razvoj metod, ki pripomorejo k boljši kakovosti izdelkov in storitev, zmanjševanje časa proizvodnje,... Tretja izmed omenjenih pozitivnih sprememb poslovanja, *nagle izboljšave*, izvira iz zmožnosti organizacije, da vidi posel na drugačen način, s perspektive, ki jo njena konkurenca ni upoštevala. Vodi k razvoju novih produktov oziroma storitev, ki privabljajo potrošnike in povečujejo prednost pred konkurenco. Če želimo doseči prednost in jo tudi ohraniti, moramo upoštevati vse tri strategije in razviti ali prenoviti poslovne procese na tak način, da bomo maksimalno upoštevali tako pomembnost potrošnika kot tudi zmogljivosti in delo podjetja. S prenavljanjem poslovnih procesov²⁹ preprečimo povečevanje razlike med dejansko in možno učinkovitostjo izvajanja poslovnih procesov ter se prilagajamo spremembam okolja.

3.3. Prenova poslovnih procesov

Prenovo poslovnih procesov ali reinžiniring definira Michael Hammer, njen vodilni zagovornik, kot '*vnovičen temeljni premislek in korenito preoblikovanje poslovnih procesov, katerega cilj je doseči dramatične izboljšave ključnih kazalcev učinkovitosti, kot so kakovost izdelkov, storitev, hitrost, stroški*' (v Watson, 1994, str. 5). Uvede štiri ključne pojme prenove:

- Temeljni (angl. fundamental): Pri prenovi poslovnega procesa moramo vedeti, zakaj nekaj delamo in zakaj smo izbrali določen način izvedbe.
- Korenit (angl. radical): Poslovnih procesov ne želimo le površno popraviti, ampak jih raziskati do temeljev in na novo oblikovati.
- Dramatičen (angl. dramatic): Cilj prenove je preskok in ne le manjše, postopne izboljšave.
- Procesni (angl. process): Prenova ne sme biti usmerjena le k posameznim nalogam, ampak k procesom kot celoti.

²⁸ angleško: breakthrough

²⁹ angleško: Business Process Reengineering - BPR

Hammer je zagovornik korenitih sprememb in rešitev, ki prinašajo skokovite izboljšave, ter spodbuja nosilce prenove, da se je raje lotijo s popolnoma praznim listom papirja³⁰ kot s prenovo starih procesov in strukture. Vprašati se je potrebno, če je sedanje poslovanje ustrezno ali ni. V primeru slednjega odgovora je potrebno postaviti nove temelje, na katerih bomo zgradili novo filozofijo poslovanja.

Gregory H. Watson se sprašuje, kako lahko prenovimo procese, če ti niso bili nikoli definirani. Proces prenove oziroma uvajanja sprememb imenuje *inženiring poslovnih sistemov*³¹. Pri tem definira (Watson, 1994, str. 36):

- *Poslovni sistem* kot niz med seboj povezanih poslovnih procesov. *Poslovni procesi* so sestavljeni iz zaporedja aktivnosti in pripeljejo do izhodnega rezultata celotnega podjetja. Poslovni procesi so sestavljeni iz delovnih procesov, ki producirajo ali zagotavljajo specifične elemente produkta ali omogočajo podporni servis tem procesom. *Delovni procesi* so sestavljeni iz niza aktivnosti, nalog in korakov. Vsak delovni proces lahko karakteriziramo glede na kakovost izhodnih podatkov, dodajanje vrednosti produktom, porabo časa in sredstev pri izdelavi produkta ter zmožnost izdelave produkta.
- *Inženiring* kot umetnost razvoja in implementacije znanja v načrt produkta ali procesa.

Na podlagi definicije poslovnega sistema in inženiringa definira inženiring poslovnih sistemov kot metodo identificiranja, določanja in implementacije sprememb. Problemi se ne pojavljajo posamično in zato izboljšanje posameznega poslovnega procesa, če ga ne obravnavamo kot delčka celote, ponavadi ne prinese zadovoljive rešitve. Watson meni, da mora organizacija kot celota strmeti k prenovi in uresničitvi svojih strateških ciljev. Strukturo organizacije predstavi z atomi, s tremi med seboj povezanimi krogi, kjer vsak izmed atomov predstavlja določen nivo organizacije (glej sliko 4 na str. 20). Poudarjena je povezanost med nivoji, saj se lahko le z ustreznim sodelovanjem doseže poslovne uspehe.

Prvi nivo, *nivo podjetja*³², predstavlja vrhovni management podjetja. Vrhovni management mora organizirati in spodbujati spremembe v organizaciji. Odgovoren je za strateško planiranje, izdelano mora imeti jasno vizijo napredka. Vrhovni management imenuje prenoviteljsko skupino, sestavljeno iz strokovnjakov, ki dobro poznajo obstoječe procese, na osnovi katerih zasnujejo nove procese in nadzirajo prenovo. Strateški cilji, vizija organizacije, misija in cilji organizacije so vključeni v strateški plan in cilje organizacije, kar zagotavlja doseganje zastavljenih dolgoročnih ciljev. Vrhovni management nadzira uresničevanje strateških planov in ciljev na operativnem in poslovnem nivoju.

³⁰ angleško: clean sheet of paper

³¹ angleško: business systems engineering

³² angleško: enterprise level

Na drugem nivoju, *poslovnem nivoju*³³, se strateški plani in cilji prevedejo v poslovne cilje. Poslovni nivo je zadolžen za produkcijo izdelkov in izvedbo storitev. Tesno je povezan s potrošniki, saj mora razumeti njihove potrebe in pričakovanja ter vedeti, kakšen je njihov odziv na poslovanje konkurence. Tržni delež, zvestoba in zadovoljstvo potrošnikov ter odzivni čas na njegove potrebe so nekateri izmed ključnih kazalcev, ki se morajo upoštevati na poslovnem nivoju organizacije.

Slika 4: Atomska struktura organizacije



Vir: Watson, 1994, str. 47

Tretji nivo je povezan s procesi, *operativni nivo*³⁴. Poslovni cilji se prevedejo v plan akcij. Naloga operativnega nivoja je vzdrževanje pretoka produktov in storitev do potrošnika. Merljive količine, na katerih temeljijo izboljšave, so delež napak, čas procesnega kroga in doseganje standardov.

3.4. Postopek prenove

Podjetja se danes nahajajo v svetu kaosa in morajo zato biti prilagodljiva³⁵, konkurenčna in tekmovalna (Watson, 1994, str. 56). Glavni cilj vsakega podjetja je ustvarjanje vrednosti. Pravo vrednost določa potrošnik in ne vodstvo podjetja, zato mora podjetje, če hoče biti uspešno, analizirati svoje poslovanje in ugotoviti, na kakšen način bo najbolj zadovoljilo potrošnike. Poslovni procesi morajo biti, da bodo v tem kaotičnem svetu čim bolj služili organizaciji, prilagodljivi in taki, da se jih da na enostaven način preurediti. Cilji, ki jih želimo s prenovo doseči, so:

- Odprava prevelike razdrobljenosti procesov, ki ovira koordiniranje, zaradi česar obstaja večja nevarnost napak.
- Koraki v procesu naj se izvajajo v logičnem, naravnem zaporedju.

³³ angleško: business level

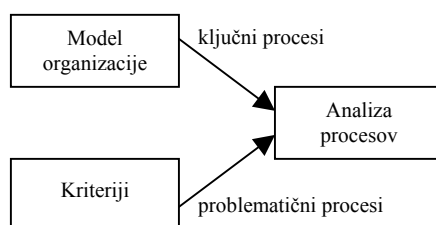
³⁴ angleško: operation level

³⁵ angleško: flexible

- Delo naj bo opravljeno tam, kjer je najbolj smiselno.
- Odprava nepotrebnega nadzora in preverjanj.
- Poenostavitev procesa, če je to mogoče, in s tem zmanjševanje pojavljanja napak.

Postopek prenove procesov se prične z **opredelitvijo obstoječih procesov** v organizaciji. Pred prenovo poslovnih procesov se je potrebno nanjo dobro pripraviti in organizirati. Vhodni element predstavlja model organizacije, saj je potrebno analizirati njeno poslovanje. Vsak nivo organizacije je pomemben za njeno delovanje in za vsakega je potrebno definirati, kam mora biti usmerjena njegova pozornost. Identificirajo se ključni poslovni procesi organizacije in njim podporni poslovni procesi. Opredele se njihov namen in medsebojne povezave.

Slika 5: Vhodni elementi prenove procesov



Prirejeno po Watson, 1994, str. 58

Sledi **izbor procesov, ki jih je potrebno preurediti**. Podlaga zanj so izbrani odločitveni kriteriji. Kriteriji se lahko nanašajo na končen produkt ali pa na sam proces. Watson (1994) predlaga pet kriterijev, ki predstavljajo merilo za konkurenčno prednost organizacije. To so *kakovost, stroški, čas, tehnologija in odzivnost*. Merila, ki jih uporabimo pri izdelavi vrstnega reda procesov, ki jih želimo preurediti, so (Hammer, Champy, 1993, str. 56-58): nefunkcionalnost procesa, njegova pomembnost in izvedljivost prenove. Nefunkcionalni procesi so procesi, ki zaradi simptomov, kot so prevelika razdrobljenost naravnega procesa, preveliko število kontrol, podvajanja del, prevelika kompleksnosti postopkov, ne delujejo optimalno. Nadalje upoštevamo, kateri procesi so za uporabnike najpomembnejši, in nazadnje še, če je njihova prenova izvedljiva oziroma, če je verjetno, da bo njihova prenova uspela in prinesla želene rezultate.

Preden se lotimo prenove procesov, jih moramo tudi razumeti do potankosti. **Analiza procesov** predstavlja osnovni proces, ki vodi k izboljšanju kvalitete, in mora biti temeljna aktivnost vsake organizacije. Je metodologija namenjena dokumentiranju, upravljanju in optimizaciji poslovnih in delovnih procesov. Delovni procesi predstavljajo elementarni nivo obravnave poslovanja in jedro, iz katerih so zgrajeni poslovni procesi in temelji celotnega poslovanja organizacije. Delovne procese lahko ocenimo glede na (Watson, 1994, str. 66):

- kakovost njegovega izhoda,
- dodajanje vrednosti izhodu,

- porabo časa, virov in sredstev,
- zahtevane spretnosti in znanja.

Če želimo razumeti delovanje delovnih procesov je potrebno, da so dobro dokumentirani. Identificirati moramo lastnike in uporabnike procesov, uporabnikove zahteve, omejitve, izhode in vhode procesov, določiti kriterije, na podlagi katerih bomo ocenili uspešnost prenove, in ključne dejavnike uspeha³⁶. Dobro poznavanje delovnih procesov omogoča boljše poznavanje poslovnih procesov, ki so sestavljeni iz teh delovnih procesov, in olajša sporazumevanje med udeleženci prenove poslovnega procesa, dokumentacija pa pripomore k razumevanju trenutnega stanja procesa in nam je v pomoč pri odkrivanju obstoječih problemov in predvidevanju mogočih sprememb. Če je cilj prenovljenega poslovnega procesa skrajšanje izvedbenega časa, potem moramo oceniti trajanje obstoječega procesa, saj v nasprotnem primeru primerjava ne bo mogoča.

S *prenovo procesov* želimo procese preoblikovati tako, da bodo odpravljene ugotovljene pomanjkljivosti. Več nalog združimo v eno in odpravimo nepotrebne naloge. Proces preuredimo tako, da bodo njegovi koraki potekali v naravnem vrstnem redu. Zmanjšamo število kontrol in ohranimo le tiste, ki so upravičene. Zapletene procese poenostavimo. Delo organiziramo tako, da zmanjšamo možnost neusklajenosti dogovorov. Cilj prenove je torej proces, ki bo v kratkem času in z nizkimi stroški v čim manj korakih zagotavljal ustrezno kakovost produkta ali storitev. Korenita prenova poslovnih procesov vpliva na način dela v celotni organizaciji, zato se lahko pojavi tudi odpor ali strah pred uvedbo sprememb znotraj same organizacije. Pomembno je, da se uvajanja sprememb lotimo na pravi način, ki bo poudaril pomembnost sprememb.

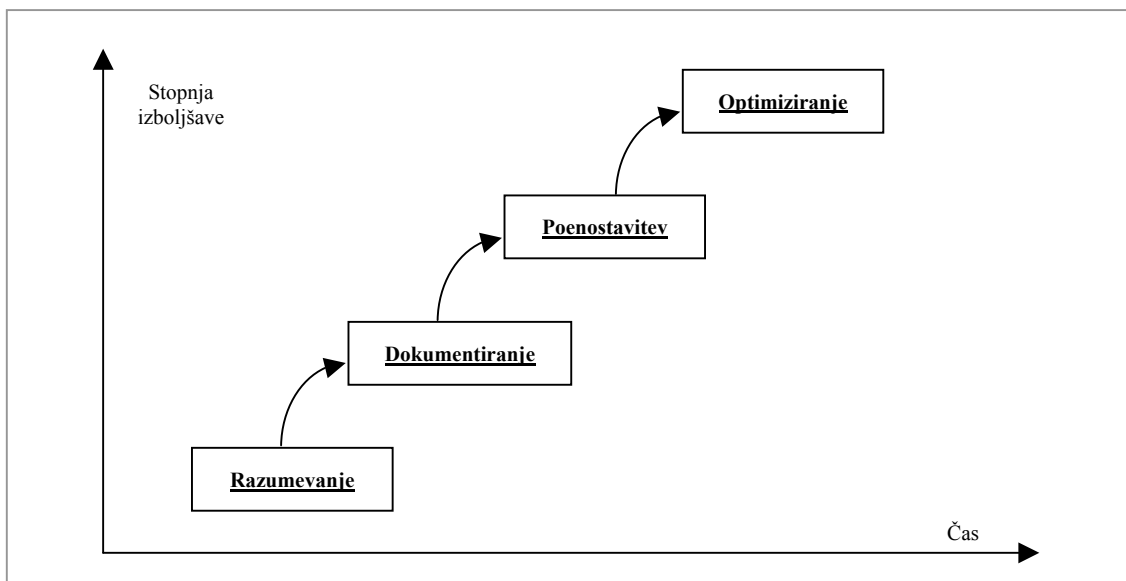
Watson (1994) predlaga proces postopnega izboljševanja poslovnih procesov, sestavljenega iz štirih korakov. Njegova ideja je preprečiti nastanek problemov in optimizirati proces namesto iskanja in popravljanja napak v procesih.

- V prvem koraku se osredotočimo na celotno organizacijo. Spremembe bodo izvedene le, če bodo ljudje na njih pripravljeni. Zato je potrebno dobro poznati strukturo zaposlenih in kakšno je njihovo medsebojno sodelovanje. Odpravimo morebitne medsebojne konflikte, ki bi zavirali potek procesa. Zgradimo dobre medsebojne odnose, organiziramo time, določimo skupne cilje, razdelimo delo, zadolžitve in odgovornosti, oblikujemo skupne norme in vrednote. Preučimo kako tok izdelave produktov in storitev potujejo znotraj organizacije. Ne moremo pričakovati kakovost in optimizacije procesov, če ljudje ne morejo delati skupaj. Organizacijo oblikujemo tako, da bo sposobna učinkoviteje izrabiti znanje zaposlenih. Hkrati povečamo pripravljenost ljudi, da sprejmejo spremembe. S tem, ko povečamo *razumevanje*, zakaj so spremembe potrebne, zmanjšamo strah in predsodke pred njimi.

³⁶ Ključni dejavniki uspeha so dejavniki, ki naredijo delovni proces uspešen s strani njegovega odjemalca.

- V drugem koraku izdelamo podrobno *dokumentacijo* vseh novih ali prenovljenih procesov. Obstoje dokumentacije pomeni, da se ljudje strinjajo, kako naj bi posamezen proces potekal. Pojavljajo se kritike, da podrobno dokumentiranje procesov in uvajanje standardov (na primer ISO 9000), pomeni stagnacijo procesov. Vendar temu ni tako. Ob dokumentiranju lahko odkrijemo marsikatero nepravilnost in nedoslednost in jo v tem koraku odpravimo.
- V tretjem koraku se lotimo *izboljševanja učinkovitosti*. Poenostavimo delovne procese. Odpravimo naloge, ki ne dodajajo vrednosti, optimiziramo naloge, ki trajajo predolgo, ter uskladimo naloge in aktivnosti, ki lahko potekajo sočasno. Cilj tretjega koraka je zmanjšanje časa trajanja procesa, povečanje vrednosti izhodov procesa in zmanjšanje stroškov.
- V zadnjem, četrtem koraku, nadaljujemo z *optimizacijo* procesov in odstranimo variabilnost izhodnih produktov. S statističnim spremljanjem rezultatov procesa poskušamo oblikovati proces, ki bo konsistentno izdeloval produkte, ki med seboj ne bodo odstopali.

Slika 6: Stopnje izboljševanja poslovnega procesa



Vir: Watson, 1994, str. 98

Šele ko zaključimo četrti korak lahko procese *avtomatiziramo*. Zavedati se moramo, da je med avtomatizacijo in prenovo procesov bistvena razlika. Z avtomatizacijo ne spremenimo temeljev procesov, ampak samo pospešimo in izboljšamo izvajanje posameznih nalog.

4. Objektno orientiran pristop razvoja IS

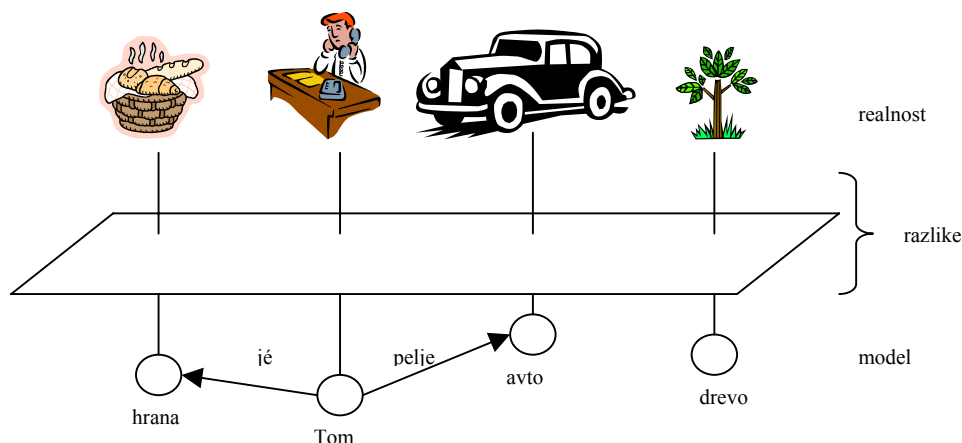
V drugem poglavju smo na splošno predstavili načrtovanje informacijskih sistemov in omenili metodologijo imenovano *tradicionalni življenjski cikel informacijskega sistema*, ki predstavlja izhodišče za vse kasnejše razvite metodologije načrtovanja informacijskih sistemov. V četrtem poglavju pa bo predstavljena ena izmed novejših tehnik načrtovanja, tako imenovan *objektno orientiran pristop razvoja informacijskih sistemov*.

4.1. Značilnosti objektnega razvoja

Objektno orientirana metodologija je ena izmed tehnik modeliranja, ki ima v zadnjem času veliko privržencev. Pred njenim pojavom so metodologije razvoja informacijskih sistemov dosledno ločevale procesni in podatkovni pogled, objektno orientiran pristop pa združuje modeliranje podatkov in procesov.

Ljudje dojemamo svet in svojo okolico kot zbirko objektov z njihovimi lastnostmi in obnašanjem (Kaj je objekt, bomo definirali kasneje.), zato bi želeli preslikati realnost v model na tak način, da bo preslikava čim bolj verodostojna in bodo razlike v predstavitvi čim manjše.

Slika 7: Neposredna preslikava realnosti v objekte



Prirejeno po Jacobson, 1996, str. 43

Bistvo objektnega modeliranja je torej, da se vsak objekt realnega sveta predstavi z objektom v modelu³⁷ (Grad, Jaklič, 1996, str. 115). Objektni pristop se je pojavil že v začetku sedemdesetih let, vzroki, da se pogostejše uporablja v zadnjem času, pa so naslednji (Avison, Fitzgerald, 1996, str. 70):

³⁷ Model je abstrakcija, predstavitev dela realnega sveta (Avison, Fitzgerald, 1996, str. 61).

- Popularnost grafičnih vmesnikov³⁸.
- Pogostejša uporaba programskega jezika C++.
- Osredotočenje poslovanja na zmanjšanje stroškov in potreba po večkratni uporabljivosti³⁹ programov.
- Razvoj tehnologije in večja uporaba računalnikov pri vsakdanjem delu.

Ugodnosti, ki jih prinašajo objektno orientirane metodologije, so (Khoshafian, Abnous, 1995, str. 3; Avison, Fitzgerald, 1996, str. 72, povzeto od Coad, Yourdon, 1991):

- Majhna razlika med modelom in realnim svetom.
- Večkratna uporabnost programskih komponent in enostavno razširjanje knjižnic⁴⁰ programskih modulov.
- Enostavno spreminjanje in razširjanje implementacij, ne da bi bilo potrebno ponoviti vsega od začetnih skic.
- Boljša komunikacija med analitikom in uporabnikom, saj je razumevanje pristopa na obeh straneh enako.
- Prožnost⁴¹ modela, ker lahko v modelu že predstavimo možne spremembe.
- Večja konsistentnost rezultatov, saj so vsi pogledi na problem modelirani na enak način.

Kljub vsem prednostim, ki jih ponuja objektno orientiran pristop, se moramo zavedati, da ima tudi nekaj pasti (Graham, 1994, str. 52). Zaradi časovne omejitve projekta, na primer, se lahko zgodi, da ni natančno in do konca implementirana ponovna uporabnost programskih komponent, popolna podpora večkratnemu dedovanju (glej str. 30) še ni na voljo, izdelati je potrebno knjižnice objektov in razširiti znanje o njih, pri uporabi dedovanja (glej str. 30) je potrebna natančna kontrola, če ne želimo, da ni bi bila ponovna uporabljivost komponent v nevarnosti, ne moremo se izogniti stroškom izobraževanja,...

4.2. Koncepti objektnega pristopa

Prednost objektno orientiranega pristopa lahko predstavimo z naslednjo formulo (Graham, 1994, str. 51):

ograjevanje+dedovanje+identiteta→večkratna uporabljivost+elastičnost+bogata semantika⁴²

³⁸ angleško: interface

³⁹ angleško: re-use

⁴⁰ angleško: libraries

⁴¹ angleško: resilient

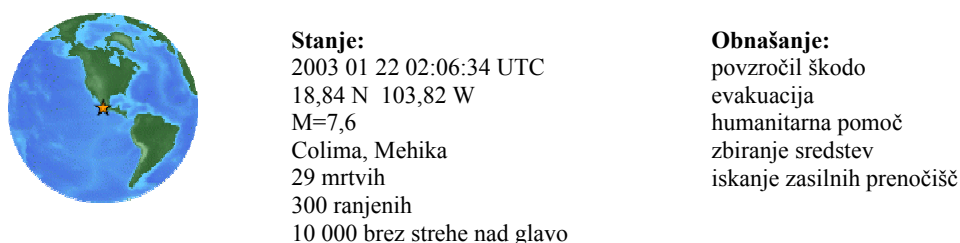
⁴² angleško: semantic richness

Ograjevanje, dedovanje in identiteta so nekateri izmed konceptov, na katerih temeljijo objektno orientirane metodologije, ki bodo predstavljene v nadaljevanju.

4.2.1. Objekt

Osnovni koncept, na katerem temeljijo objektna tehnologije, je, kot pove že ime samo, **objekt**. Predstavlja katerakoli stvar (entiteto) realnega sveta, o kateri hranimo podatke in njeno obnašanje (Grad, Jaklič, 1996, str. 117). Objekti so abstraktna predstavitev realnih stvari, kot na primer račun, avtomobil, telefon, oseba, žival, ki obstajajo v času in prostoru. Vsak vsebuje individualno informacijo (**identiteto**), po kateri se loči od ostalih objektov. Na primer vsak avto ima svojo registracijsko številko. Za vsak objekt so značilna stanja in obnašanje.

Slika 8: Potres v Mehiki kot objekt, njegova stanja in obnašanje

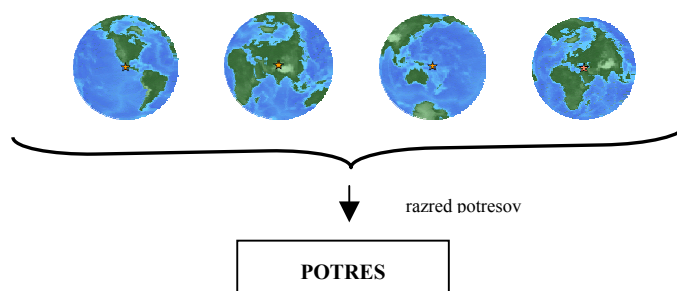


Vir: USGS, 2003

4.2.2. Razred

Objekti ponavadi niso izolirani. V sistemih imamo veliko število objektov, ki med seboj komunicirajo. Nekateri med njimi imajo podobne karakteristike (lastnosti in obnašanje), zato jih lahko na podlagi tega združimo v objektne tipe. Tiste, ki imajo tako enake operacije kot tudi enako notranjo strukturo, združimo v objektne razrede, čemur ponavadi rečemo na kratko kar **razred**. Vsak objekt nekega razreda je **primerek objekta** tega razreda (na kratko primerek razreda). V primeru, ki smo ga podali na sliki 8, so potresi v Mehiki, na Kitajskem, pri Salomonovem otočju in v Turčiji primerki razreda POTRES.

Slika 9: Razred potresov

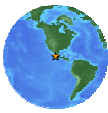
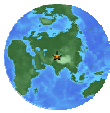
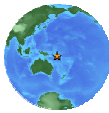


4.2.3. Atribut

Stanje objekta je določeno z vrednostmi atributov, ki ga opisujejo. Atribut je lastnost objektov v razredu, ki se lahko s časom spreminja. Vsak objekt ima enega ali več atributov, ki ga opisujejo in ločijo od ostalih objektov. Tem atributom pravimo tudi *kandidati za ključ*⁴³.

Atribute delimo na enostavne, referenčne in izpeljane atribute (Grad, Jaklič, 1996, str. 118). Vrednosti **enostavnih atributov** so literali, **referenčni** omogočajo predstavitev povezav med objekti, vrednosti **izpeljanih** pa lahko izračunamo iz vrednosti ostalih atributov. Ponavadi pri slednjih nimamo shranjenih njihovih vrednosti, ampak le postopek, ki se izvrši, ko vrednost atributa potrebujemo.

Tabela 1: Vrednosti atributov posameznega objekta razreda POTRES

Primerek Atribut				
Datum-Čas	2003 01 22 02:06:34 UTC	2003 02 24 02:03:41 UTC	2003 01 20 08:43:06 UTC	2003 05 01 00:27:04 UTC
Regija	Colima, Mehika	južni Sinjiang, Kitajska	Salomonovo otočje	Bingöl, vzhodna Turčija
Lon	18,84 N	39,61 N	10 40 S	39,01 N
Lat	103,82 W	77,24 E	160.75 E	40,51 E
Globina	24 km	11 km	33 km	10
Magnituda	7,6	6.4	7,3	6,4
Št. mrtvih	29	263	-	176

4.2.4. Operacije

Poleg svojih atributov vsebuje vsak razred tudi operacije, s katerimi je opisano obnašanje objektov v razredu (glej sliko 8 na str. 26) in ki se uporabljajo za branje ali manipulacijo podatkov o objektu. Vsak objekt mora vsebovati nekatere osnovne operacije, ki omogočajo delo z objekti: kreiranje novega objekta, dostop do objekta in brisanje objekta. (Grad, Jaklič, 1996, str. 120). Poleg tega lahko vsebuje še dodatne operacije, ki opisujejo njegovo obnašanje. Operacije razreda POTRES so lahko na primer Išči, Zbiraj, Evakuiraj. Z izrazom *metoda* označimo operacijo, zakodirano v programu.

4.2.5. Ograjevanje

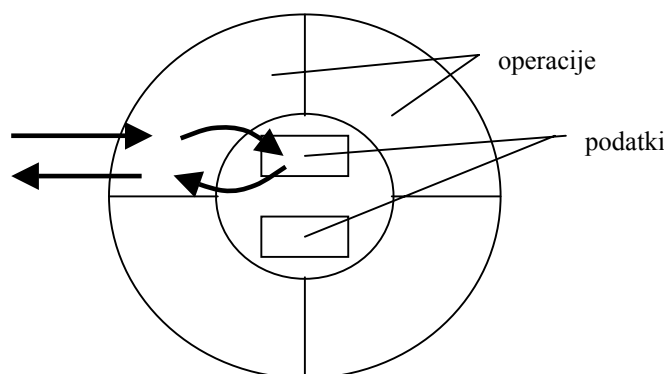
Operacije nekega objekta lahko neposredno dosegaajo samo podatke o objektu samem. To lastnost, značilno za objektno orientirano modeliranje, imenujemo **ograjevanje** ali

⁴³ Kandidat za ključ je minimalen set atributov razreda, ki enolično predstavljajo posamezen objekt razreda.

enkapsulacija. Ograjevanje je lastnost objekta, da lahko vidimo, dosežemo in spreminjamo njegove podatke in lastnosti le skozi njegove operacije (Grad, Jaklič, 1996, str. 119).

Ograjevanje, skrivanje strukture podatkov o objektu pred drugimi objekti in posledično s tem njegova zaščita omogoča večjo elastičnost pri spreminjanju obnašanja objektov, saj ni potrebno spreminjati programov, ki potrebujejo rezultate določenih operacij, če se te spremenijo.

Slika 10: Ograjevanje



Vir: Harmon, Taylor, 1993, str. 4

Omenili smo, da imenujemo operacije, kodirane v programu, metode. Ker so pri ograjevanju podatki o objektu dosegljivi le preko operacij, so posledično tudi metode skrite in zaščitene pred zunanjim vdorom.

4.2.6. Mnogoličnost

Pri objektno orientiranem pristopu je značilna še ena lastnost operacij. To je **mnogoličnost** ali **polimorfizem**. Mnogoličnost pomeni, da se objekti različnih razredov odzovejo na isto zahtevo različno (Grad, Jaklič, 1996, str. 121). Zahteva se interpretira na način, ki ga določa prejemnikov razred. Mnogoličnost omogoča, da lahko vpeljemo v sistem nove razrede, ne da bi bilo potrebno spremeniti obstoječe.

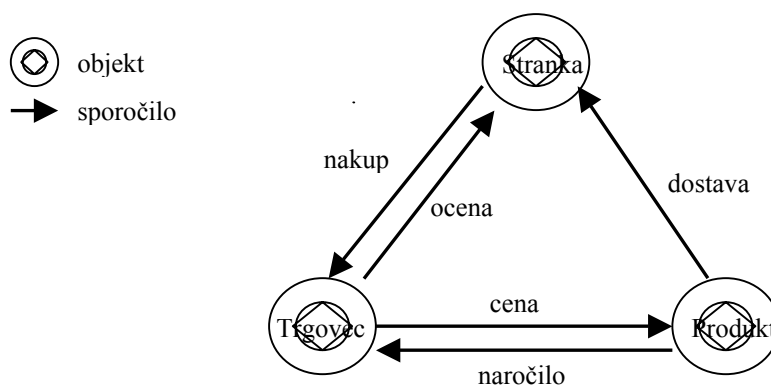
4.2.7. Sporočilo

Med seboj si objekti izmenjujejo sporočila in le preko njih komunicirajo med seboj. Objekt, ki pošlje sporočilo (pošiljatelj), pošlje sporočilo drugemu objektu (sprejemniku). S tem sproži zahtevo za sprožitev neke operacije. Običajno sporočila sestavljajo:

- ime prejemnika sporočila,
- ime operacije, ki naj se izvede, in
- parametri operacije, če ti obstajajo.

Po izvršeni operaciji sprejemnik vrne rezultat objektu, ki je sprožil akcijo. S sporočili brišemo, kreiramo in spreminjamo objekte, omogočajo tudi tok podatkov in kontrolo nad objekti, ne moramo pa poseči v njihovo notranjost in izvorno kodo (omejitev zaradi ograjevanja).

Slika 11: Tok sporočil med objekti



Vir: Harmon, Taylor, 1993, str. 3

4.2.8. Povezave med razredi

Osnova objektno orientiranih metodologij so objekti definirani z razredi. Da bi sistem deloval, moramo poznati tudi povezave med njimi. Objekti so lahko povezani med seboj na tri načine (Harmon, Taylor, 1993, str. 6):

- **Specializacija:** Tip povezave strukture »Je«⁴⁴. Podrobnejši opis je naveden na strani 30 v okviru opisa dedovanja.
- **Agregacija:** Agregacija je tip povezave strukture »Je-del«⁴⁵, s katero tvorimo objekte iz drugih objektov. Na primer Čas potresa lahko s procesom agregacije sestavimo iz Ure, Minute in Sekund. Tudi atributi in operacije nekega razreda so povezani s svojim razredom z relacijo strukture »Je-del« (glej sliko 38 na str. 77). Poleg atributov pa lahko vsak razred vsebuje tudi enega ali več razredov.
- **Sodelovanje**⁴⁶: S tipom povezave strukture »Je-povezan-z«⁴⁷ označimo interakcije med razredi. Stanje drugega razreda je odvisno od stanja razreda, s katerim je v interakciji. Vsak razred lahko sodeluje z enim ali več razredi. Povezavi določimo tudi kardinalnost ali števnost, s katero določimo, koliko primerkov enega razreda je povezano s primerki drugega (glej sliko 38 na str. 77).

⁴⁴ angleško: Is-a

⁴⁵ angleško: Is-part-of

⁴⁶ angleško : association. Sopenenka je asociacija.

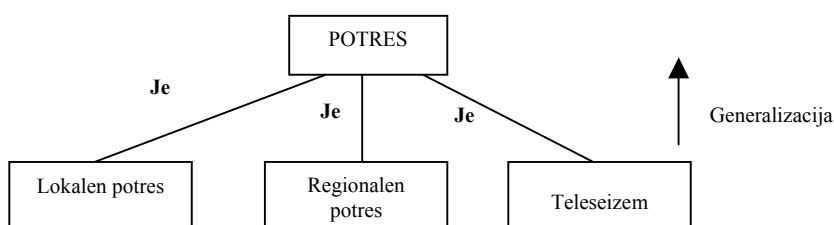
⁴⁷ angleško: Is-associated-with

4.2.9. Dedovanje

Dedovanje je še ena pomembna značilnost objektno orientiranih sistemov, ki omogoča večkratno uporabljivost podatkov in metod, vsebovanih v razredu. S tem se zmanjša tudi potrebna programska koda. Lahko bi rekli, da je zaradi tega dedovanje najbolj uporaben objektno orientiran koncept. Dedovanje poteka med razredi, ki imajo podobno podatkovno strukturo in operacije. Poznamo dve vrsti dedovanja: generalizacija in specializacija.

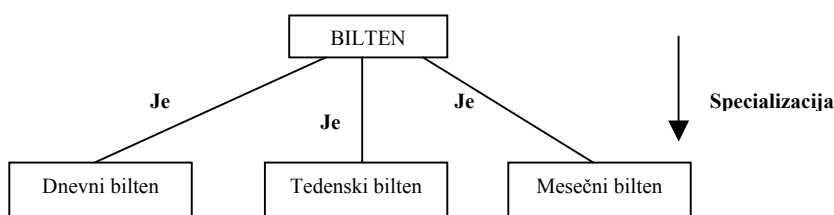
- **Generalizacija:** Kadar imata dva ali več razredov podobne atribute in operacije, lahko te združimo in prenesemo v bolj splošen razred, nadrazred⁴⁸. Povezava prvotnih razredov z nadrazredom ima strukturo »Je«. Lahko rečemo, da imajo razredi poleg svojih karakteristik tudi tiste, ki so jih podedovali od nadrazreda.

Slika 12: Primer generalizacije



- **Specializacija:** V primeru, ko hočemo upoštevati določene posebnosti obstoječih razredov, te razbijemo na podrazrede⁴⁹. Podrazred podeduje karakteristike razreda, s katerim je povezan, doda pa se mu lahko tudi lastne karakteristike.

Slika 13: Primer specializacije



Generalizacija in specializacija sta dva načina analize istih povezav. Generalizacija je analiza razredov, ki se izvaja od spodaj navzgor, medtem ko poteka specializacija od zgoraj navzdol.

Podrazred lahko podeduje karakteristike več razredov. V takem primeru govorimo o *večkratnem dedovanju*, medtem ko se dedovanje karakteristik samo enega razreda imenuje *enkratno dedovanje*.

⁴⁸ angleško: superclass

⁴⁹ angleško: subclass

5. Objektno orientirane metodologije

Peto poglavje je namenjeno predstavitvi dveh objektno orientiranih metodologij. V prvem razdelku poglavja bomo predstavili metodologijo, ki za preslikavo realnosti v model uporablja različne diagrame. Nato se bomo lotili še metodologije, ki predstavi to preslikavo z nizom tabel. Slednjo bomo v nadaljevanju uporabili tudi na praktičnem primeru pri gradnji seizmološkega informacijskega sistema.

5.1. Poenoten jezik za moduliranje UML

Poenoten jezik za modeliranje UML⁵⁰ (v nadaljevanju jezik UML) je jezik namenjen modeliranju. Je množica standardnih modelov, ki so nam v pomoč pri načrtovanju objektno orientiranih projektov, ne opisuje pa načina njegove implementacije. Leta 1995 so ga razvili G. Booch, J. Rumbaugh in I. Jacobson, kot posledico težnje po enotnem objektno usmerjenem jeziku za modeliranje. Skupaj so združili več objektno orientiranih metodologij. Najpomembnejše med njimi so OMT⁵¹, avtorja Jamesa Rumbaugh, OOSE⁵², avtorja Ivarja Jacobsona, in OOAD⁵³, avtorja Grady Boocha. Leta 1997 je bil jezik UML potrjen kot industrijski standard s strani skupine Object Management Group (OMG), ki sedaj nadzira njegov razvoj. V njej so združena vodilna podjetja s področja informacijske tehnologije. OMG skrbi za standarizacijo postopkov in specifikacij.

UML ni metodologija za razvoj informacijskih sistemov, ampak le univerzalni vizualni jezik modeliranja, s katerim lahko zgradimo različne modele (Arlow, Neustadt, 2002, str. 5) in ki nam omogoča vizualizacijo sistema, prikaz strukture in obnašanja sistema ter dokumentiranje naših odločitev. Ni vezan na določeno metodologijo ali življenjski cikel. Običajno ga povezujejo z metodologijo, ki so jo predlagali njegovi avtorji (Rumbaugh, Booch, Jacobson) in jo poimenovali Poenoten proces⁵⁴.

Z jezikom UML lahko vizualno predstavimo življenjski cikel sistema od raziskave zahtev in potreb do njegove implementacije, z njim lahko zasnujemo različne informacijske sisteme (izvajalni informacijski sistemi, sistemi za podporo odločanju,...), je nevtralen glede na izbor programskega jezika implementacije in platforme ter ni vezan, kot smo že omenili, na eno samo metodologijo. Od tod tudi njegovo ime, poenoten jezik za moduliranje.

UML strukturo sestavljajo: osnovni gradniki⁵⁵, UML mehanizmi in arhitektura sistema.

⁵⁰ angleško: Unified Modelling Language

⁵¹ angleško: Object Modeling Technique

⁵² angleško: Object-Oriented Software Engineering

⁵³ angleško: Object Oriented Analysis and Design

⁵⁴ angleško: unified process

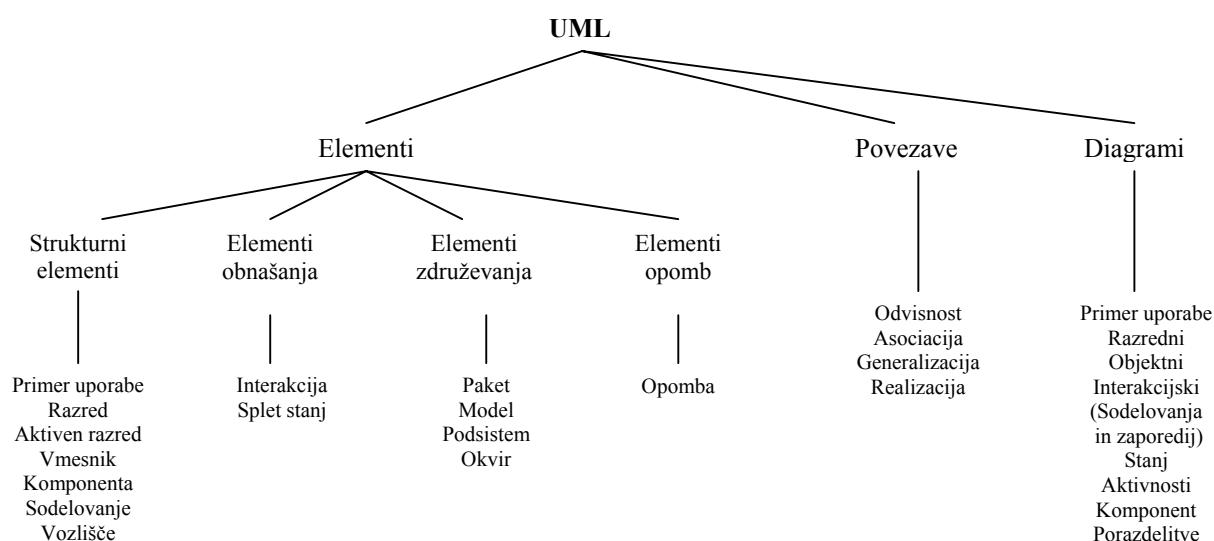
⁵⁵ angleško: building blocks

Opis sestavnih delov jezika UML je povzet po viru UML and the Unified Process: Practical Object-Oriented Analysis & Design (J. Arlow, I. Neustadt, 2002, str. 8 – 21) in The Unified Modeling Language User Guide (G. Booch, J. Rumbaugh, I. Jacobson, 1999, str. 17 - 32).

5.1.1. Osnovni gradniki

Osnovni gradniki, ki sestavljajo UML, so *elementi*⁵⁶, *povezave*⁵⁷ in *diagrami*.

Slika 14: UML gradniki



Vir: Jacobson et al., 1999, str. 422

UML *elemente* delimo na strukturne elemente, elemente obnašanja, združevanja in opomb. V nadaljevanju je podan kratek opis elementov.

- *Strukturni elementi*⁵⁸: V modelu so predstavljeni s samostalniki. Ponavadi so statični deli modela in predstavljajo fizične ali konceptualne elemente. Poznamo sedem osnovnih strukturnih elementov:
 - Primer uporabe je opis niza akcij, ki jih sistem izvede, da prinese viden rezultat določene vrednosti za določenega akterja. Uporablja se pri sestavljanju elementov obnašanja v model. Grafično je prikazan z elipso z imenom.
 - Razred je posplošen opis podobnih objektov, to je objektov, ki imajo enake attribute, operacije, metode, povezave in obnašanje. Grafično je prikazan s pravokotnikom, ki vsebuje ime, attribute in operacije.

⁵⁶ angleško: things

⁵⁷ angleško: relationships

⁵⁸ angleško: structural things

- Aktiven razred je razred, katerega objekti posedujejo enega ali več procesov, ki lahko sprožijo kontrolno aktivnost. Grafično je prikazan s pravokotnikom z bolj odebeljenimi robovi kot pri navadnem razredu in vsebuje ime, attribute in operacije.
 - Vmesnik⁵⁹ je zbirka operacij, ki določa storitve, ki jih ponuja razred, podsistem ali komponenta. Opisuje navzven vidno obnašanje objekta. Definira niz specifikacij operacij, ne pa tudi njihove implementacije. Grafično je prikazan s krogom z imenom vmesnika.
 - Komponenta je fizičen in nadomestljiv del sistema, ki poskrbi za realizacijo niza vmesnikov. Vsaka komponenta lahko vsebuje več razredov in skrbi za realizacijo različnih vmesnikov. Grafično je prikazana s pravokotnikom z dvema zankama.
 - Sodelovanjem je opredeljena interakcija in združba vlog in drugih elementov, ki s skupnim delom zagotavljajo kooperativno obnašanje. Grafično je sodelovanje prikazano s črtkano elipso, ki vsebuje ime sodelovanja.
 - Vozlišče je fizičen element, ki predstavlja računalniški vir. Ponavadi ima vsaj nekaj lastnega spomina in sposobnost procesiranja. Primer vozlišča je osebni računalnik. Grafično je prikazan kot kocka, ki ji pripišemo lastno ime.
- *Elementi obnašanja*⁶⁰: Predstavljajo dinamični del UML modela, so glagoli modela in opisujejo obnašanje skozi čas in prostor. Povezani so s strukturnimi elementi, predvsem z razredi, s sodelovanjem in z objekti. Poznamo dva osnovna elementa obnašanja:
 - Interakcija je obnašanje, ki obsega niz sporočil, izmenjanih med objekti v določenem kontekstu, katerih cilj je doseganje določenega namena. Interakcija vsebuje tudi druge elemente, med drugim sporočila, akcije (sprožene s sporočili) in povezave med objekti. Grafično je prikazana z neprekinjeno puščico, ki vsebuje ime operacije.
 - Splet stanj⁶¹ je obnašanje, ki opredeljuje niz stanj objekta, ki se izmenjujejo tekom njegove življenjske dobe kot odgovor na dogodke. Vključuje tudi druge elemente, kot so stanja, prehodi iz stanja v stanje, dogodki, ki sprožijo prehode, in aktivnosti, ki predstavljajo odgovore na prehode. Grafično je prikazan kot pravokotnik z zaobljenimi robovi, ki ponavadi vključuje njegovo ime in, če so možna, podstanja.
 - *Elementi združevanja*⁶²: Predstavljajo organizacijski del UML modela. So škatle⁶³, na katere lahko model razstavimo. Osnovni element združevanja je paket.

⁵⁹ angleško: interface

⁶⁰ angleško: behavioral things

⁶¹ angleško: state machine

⁶² angleško: grouping things

⁶³ angleško: boxes

- Paket⁶⁴ predstavlja mehanizem združevanja elementov v grupe. V njih lahko združujemo strukturne elemente in elemente obnašanja, pa tudi druge elemente združevanja. Paketi, v nasprotju s komponentami, obstajajo le v času izvajanja. Grafično so prikazani kot mapa.
- Okvirji⁶⁵, modeli in pod sistemi so variacije paketov.
- *Elementi opomb*⁶⁶: Predstavljajo del UML modela, ki vsebuje pojasnila. So pripombe, ki jih lahko vključimo v model, da bolje opišemo njegove elemente. Osnoven element opomb imenujemo opomba.
- Opomba⁶⁷ je zapisek, ki ga pripnemo enemu ali nizu elementov. Grafično je prikazana kot na enem robu zavihan list, na katerem je vpisana opomba.

S *povezavami* prikažemo medsebojno povezanost elementov v modelu. V UML jih razdelimo na asociacije, odvisnosti, generalizacije in realizacije.

- Asociacija je strukturna povezava, s katero je opisan niz povezav⁶⁸ med objekti. Agregacija je poseben primer asociacije, ki je obširneje opisana v četrtem poglavju (glej str. 29). Grafično je prikazana s polno črto, ki ji lahko na koncih pripišemo različne dodatke, kot sta na primer števnost in ime vloge.
- Odvisnost je povezava med dvema elementoma, kjer sprememba neodvisnega elementa vpliva na spremembo odvisnega. Grafično je prikazana s črtkano črto, ki kaže proti neodvisnemu elementu. «Uporablja⁶⁹» je najbolj pogosta odvisnost med razredi in pomeni, da na primer operacija odvisnega razreda potrebuje za svojo izvedbo parametre neodvisnega.
- Generalizacije so povezave tipa specializacija/generalizacija, s pomočjo katerih objekti dedujejo strukturo in obnašanje. Podrobneje je generalizacija opisana v četrtem poglavju (glej str. 30). Grafično je prikazana s puščico z votlo konico, ki je usmerjena k predniku.
- Realizacije so semantične povezave med elementi, kjer eden opredeljuje funkcije, ki jih drugi izvede. Ponavadi so to povezave med vmesniki in razredi ali komponentami oziroma med primeri uporabe in sodelovanji, ki jih uresničujejo. Grafično so prikazane s črtkano puščico z votlo konico.

Z jezikom UML modeliramo realen svet, kot sistem objektov, ki so med seboj v interakciji. Prikažemo lahko:

⁶⁴ angleško: packages

⁶⁵ angleško: framework

⁶⁶ angleško: annotational things

⁶⁷ angleško: note

⁶⁸ angleško: link

⁶⁹ angleško: uses

- *statično strukturo* (Opis objektov, ki so pomembni za model in relacije med njimi.) ali
- *dinamično strukturo*, ki nam poda življenjski cikel teh objektov in način, kako med seboj sodelujejo, da je zagotovljena zahtevana funkcionalnost sistema.

Model predstavlja skladišče⁷⁰ vseh elementov in povezav, ki smo jih kreirali in nam pomagajo prikazati obnašanje načrtovanega sistema. **Diagrami** nam služijo za prikaz različnih pogledov na obravnavani sistem. V UML imamo na voljo devet različnih diagramov. Razdelimo jih lahko v dve skupini glede na to, ali z njim prikažemo statično ali dinamično strukturo sistema.

Tabela 2: Vrste UML diagramov

DIAGRAMI		
Statični model (struktura sistema)	Dinamični model (obnašanje sistema)	
diagram razredov	diagram primerov uporabe ⁷¹	
diagram komponent	diagram zaporedja	} interakcijski diagram
diagram porazdelitve ⁷²	diagram sodelovanja	
	diagram aktivnosti	
	diagram stanj	
	diagram objektov	

Vir: Arlow, Neustadt, 2002, str. 11

Uporabnik sam izbere, katere diagrame bo glede na svoje potrebe uporabil, kljub temu pa se štirje uporabljajo skoraj vedno. Ti diagrami so: diagram primerov uporabe, interakcijski diagram, razredni diagram in aktivnostni diagram. Čeprav ni predpisanega pravila o njihovem vrstnem redu, začnemo ponavadi z diagramom primerov uporabe, s katerim definiramo namen sistema.

5.1.1.1. Diagram primerov uporabe

Diagrami primerov uporabe so namenjeni enostavnemu prikazu zahtev uporabnikov. So grafični prikaz delovanja obravnavanega sistema z vidika akterja. Osnovni elementi diagrama so akter, primer uporabe in poslovna pravila (povezave med akterji uporabe). Pri njegovi izdelavi si pomagamo z intervjuji, ki jih opravimo z uporabniki na vseh nivojih organizacije. Akterji so osebe ali drugi sistemi, ki sodelujejo z obravnavanim sistemom.

⁷⁰ angleško: repository

⁷¹ angleško: use case diagram

⁷² angleško: deployment diagram

Za vsak primer uporabe, ki opisuje funkcionalnost sistema, kreiramo dokument z naslednjimi podatki:

- ime primera uporabe,
- kratek opis namena,
- primarni akter,
- sekundarni akter,
- začetna točka (točka, kjer se primer sproži),
- končna točka (točka konca primera uporabe),
- merljiv rezultat,
- tok dogodkov,
- alternativni tok,
- razširjen primer uporabe,
- neustaljen izhod.

Po potrebi lahko nabor podatkov razširimo in prilagodimo lastnim potrebam.

Slika 15: Primer predstavitve primera uporabe

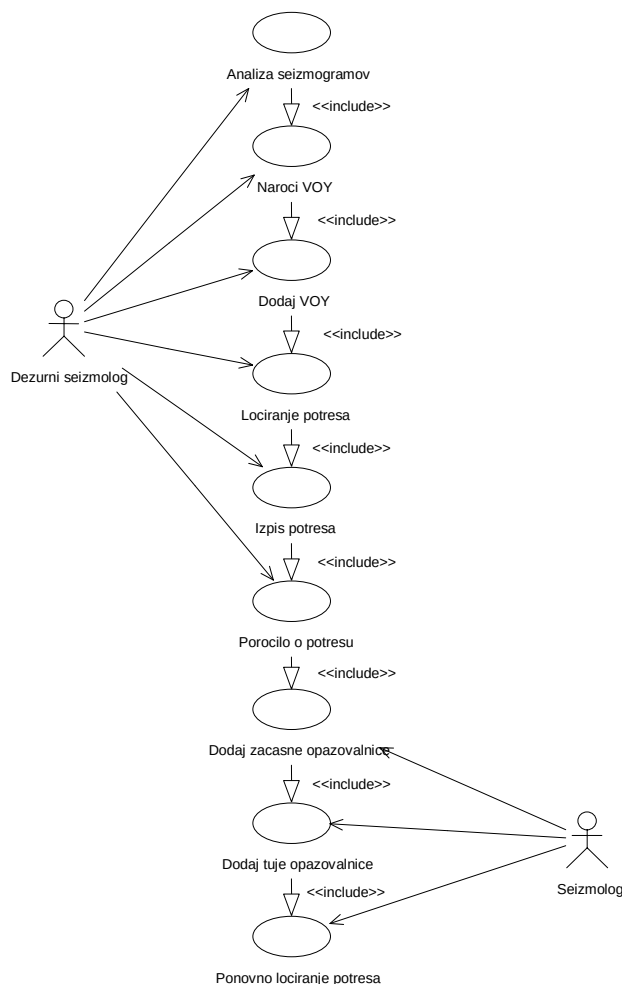
Primer uporabe: Analiza seizmogramov	
Opis	Dežurni seizmolog naredi analizo seizmogramov.
Primarni akter	Dežurni seizmolog
Sekundarni akter	CORS
Začetna točka	Trenutek prejema obvestila s CORS-a o potresu.
Končna točka	Končana analiza
Merljiv rezultat	Analiza kreirana
Tok dogodkov	Ko dobi dežurni seizmolog obvestilo o potresu, sproži zahtevo za analizo seizmogramov. Na njih očita podatke o postaji, fazi, vstopnem času valov, smeri vstopa, amplitudi in dT.
Alternativni tok	Če postaje niso zabeležile ničesar, konča z delom in obvesti CORS, da potresa ni bilo.
Razširjeni primer uporabe	-
Neustaljeni izhod	-

Ko izdelamo za vsak primer uporabe svoj dokument, jih prikažemo še z diagramom (glej sliko 16 na str. 37). Diagram primerov uporabe je dinamičen pogled na relacije med različnimi primeri uporabe. Vsak primer uporabe, predstavljen z elipso, je povezan z vsaj enim akterjem, ki ga predstavlja močic⁷³. Ko povlečemo vse povezave med akterji in primeri uporabe,

⁷³ angleško: a stick figure

dodamo še ustrezne povezave med posameznimi primeri uporabe (če te obstajajo). Uporabimo stereotipa *vključuje*⁷⁴ in *razširja*⁷⁵, kjer prvi pomeni, da prvi primer uporabe vključuje drugega, drugi pa, da je prvi primer uporabe variacija drugega primera uporabe (V dokumentu je to zapisano pod »Razširjeni primer uporabe«).

Slika 16: Primer diagrama primerov uporabe



5.1.1.2. Interakcijski diagram

Interakcijski diagrami grafično prikazujejo zaporedje komunikacij vsakega primera uporabe. Poznamo dve vrsti interakcijskih diagramov: diagram zaporedij⁷⁶ in diagram sodelovanja⁷⁷. V diagramu zaporedij je prikazano časovno zaporedje komunikacij, medtem ko diagram sodelovanja daje večji poudarek povezavi med elementi.

⁷⁴ angleško: include

⁷⁵ angleško: extend

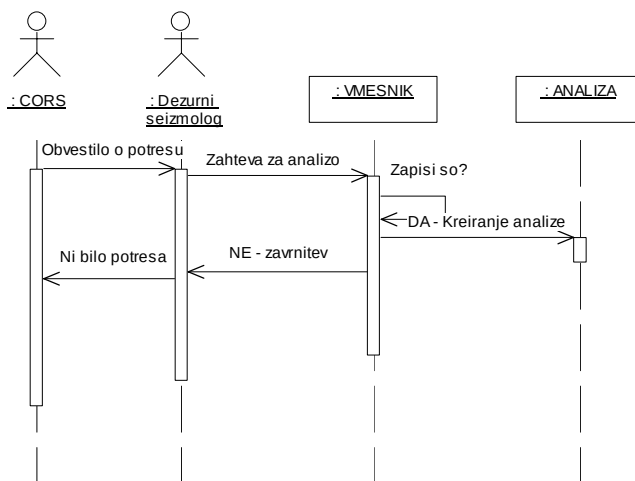
⁷⁶ angleško: sequence diagram

⁷⁷ angleško: collaboration diagram

5.1.1.2.1. Diagram zaporedij

Z diagramom zaporedij prikažemo časovno zaporedje izmenjave sporočil med objekti sistema in zunanji akterji. Sestavljen je iz akterjev, objektov, zvez in sporočil. Objekti so označeni s pravokotniki. Objekti in akterji si izmenjujejo sporočila, ki so ponazorjena s puščicami (smer puščice je smer toka sporočila). Navpične črte predstavljajo življenjske črte objekta, pravokotniki na njih pa obdobje, ko je objekt aktiven.

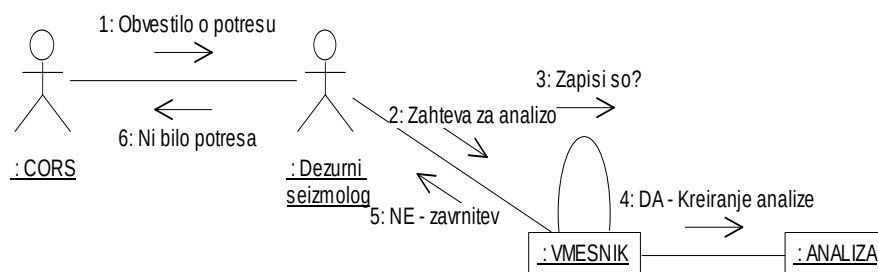
Slika 17: Primer diagrama zaporedij



5.1.1.2.2. Diagram sodelovanja

Tako kot diagram zaporedij, je tudi diagram sodelovanja namenjen prikazu dinamike sodelovanja med objekti, le da bolj poudarja organizacijo objektov, ki med seboj sodelujejo. Vrstni red sporočil je označen z zaporedno številko, ki je pripisana sporočilu.

Slika 18: Primer diagrama sodelovanja

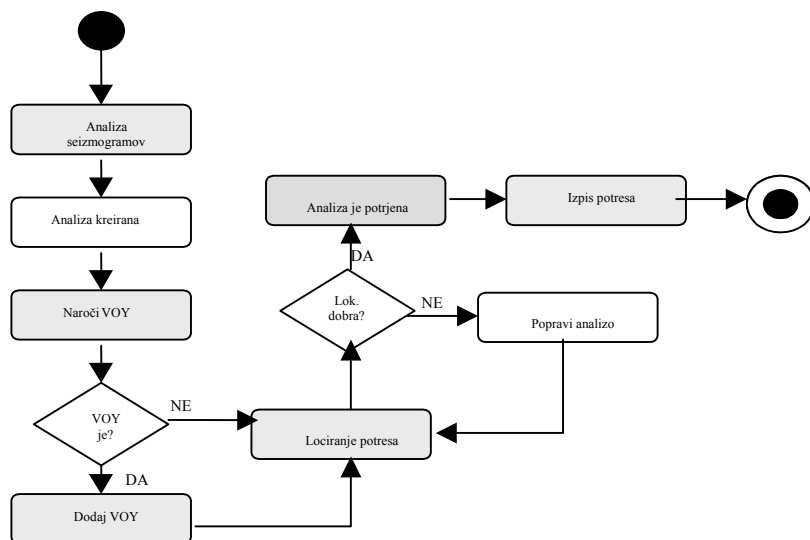


5.1.1.3. Diagram aktivnosti

Diagram aktivnosti je grafičen prikaz zaporedja aktivnosti in predstavlja dobro osnovo za fazo implementacije. Informacije črpa iz interakcijskih diagramov in jih predstavi bolj natančno.

Vsebuje aktivnosti in akcije, ki se izvajajo. Vključuje tudi točke Odločitve⁷⁸, označene z romбом, v katerih tok aktivnosti nadaljuje pot v smeri, ki je odvisna od izhoda Odločitve.

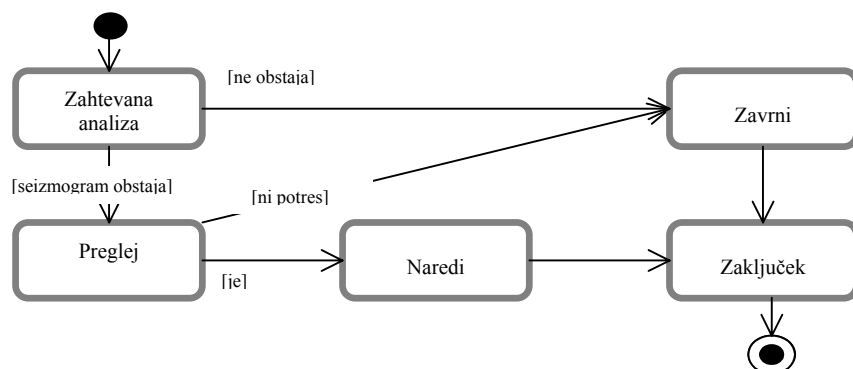
Slika 19: Primer diagrama aktivnosti



5.1.1.4. Diagram stanj

Diagram stanj se uporablja za predstavitev spreminjanja stanj sistema, ki jih povzročajo različni dogodki. Je eden izmed šestih diagramov UML modeliranja, ki predstavlja dinamičen pogled na sistem. Sestavljen je iz stanj, prehodov, dogodkov in aktivnosti. Uporaben je pri modeliranju obnašanja vmesnika, razreda ali komponente. Diagram aktivnosti je poseben primer diagrama stanj, kjer so prikazana vsa stanja aktivnosti in so prehodi sproženi z zaključitvijo določene aktivnosti.

Slika 20: Primer diagrama stanj

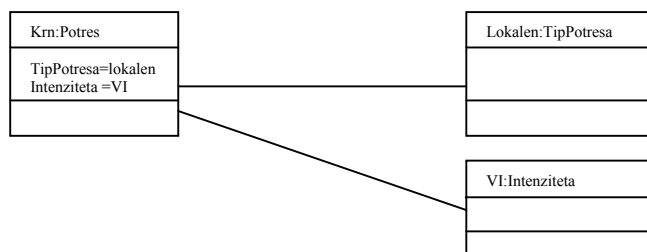


⁷⁸ angleško: decision point

5.1.1.5. Diagram objektov

Objektni diagram prikazuje niz objektov in povezave med njimi. Prikazuje strukturo podatkov in trenutni posnetek⁷⁹ sistema v nekem trenutku delovanja. V primerjavi z razrednim diagramom prikazuje objektni diagram povezave med konkretnimi primerki razredov.

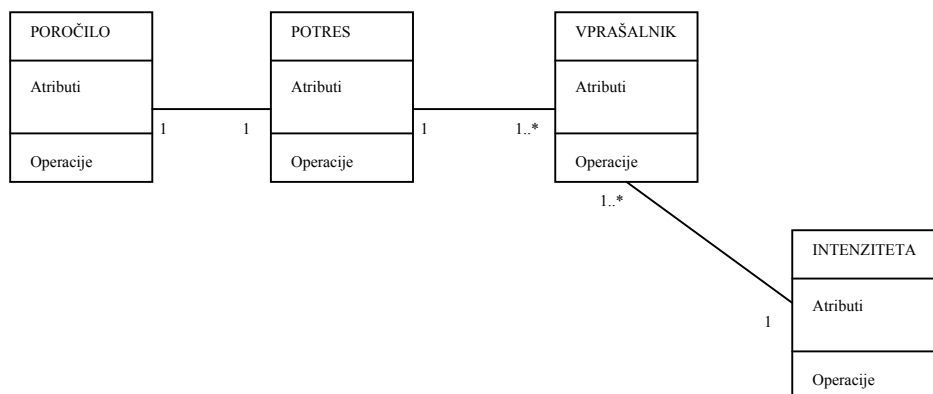
Slika 21: Primer objektnega diagrama



5.1.1.6. Diagram razredov

Razredni diagram prikazuje statičen model razredov. Prikazuje razrede, vmesnike, sodelovanja in njihove povezave. Vsak razred, na vrhu označen z imenom, je v diagramu razredov predstavljen s pravokotnikom. Navedeni so tudi njegovi atributi in operacije. Ti so lahko javni, prijateljski ali zasebni. Javne, označene z znakom »+«, lahko uporabljajo vsi razredi, zasebne, označene z znakom »-«, uporablja le razred sam, prijateljske, označene z znakom »#«, pa le določen podsistem razredov. Povezave med razredi so lahko agregacije, asociacije ali generalizacije. Povezavam pripišemo tudi števnost (glej str. 29) in smer.

Slika 22: Primer diagrama razredov



5.1.1.7. Diagram komponent

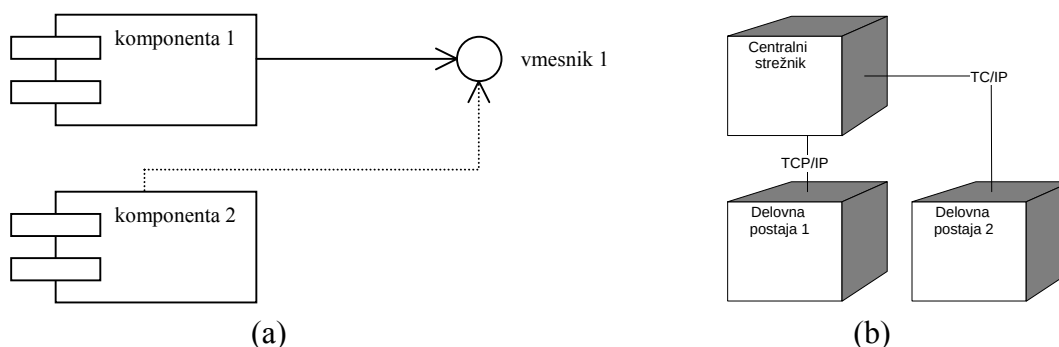
Diagram komponent prikazuje organizacijo komponent in odvisnosti med njimi. Ker komponente ponavadi vsebujejo enega ali več razredov, vmesnikov ali sodelovanj, je povezan z razrednim diagramom. Z njim prikažemo statično strukturo sistema (glej sliko 23(a) na naslednji strani).

⁷⁹ angleško: snapshot

5.1.1.8. Diagram porazdelitve

Diagram porazdelitve prikazuje strojno in programsko opremo sistema. Je statičen pogled na arhitekturo, to je na fizično strukturo sistema. Sestavljen je iz vozlišč in odvisnostmi ter asociacijami med njimi. Prikazuje lahko tudi namestitev komponent v vozlišča in vsebuje pakete in podsisteme.

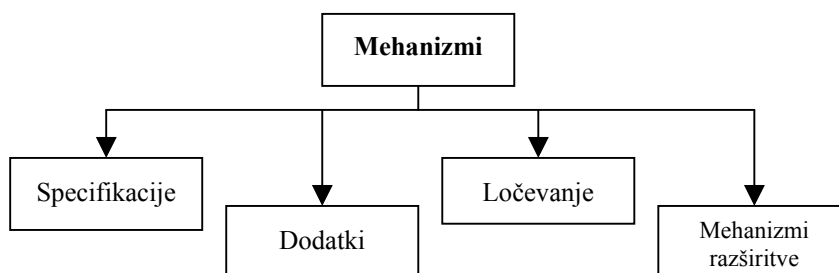
Slika 23: Primer diagrama komponent (a) in diagrama porazdelitve (b)



5.1.2. UML mehanizmi

V UML obstajajo štirje mehanizmi, ki zagotavljajo njegovo doslednost in opisujejo štiri pristope k objektnemu modeliranju, ki se pojavljajo v različnih kontekstih skozi celoten UML.

Slika 24: UML mehanizmi



Vir: Arlow, Neustadt, 2002, str. 12

UML modeli imajo vsaj dve dimenziji. Z uporabo diagramov in ikon vizualiziramo model (grafična dimenzija). Drugo dimenzijo, tekstovno, sestavljajo opisi različnih elementov modeliranja. *Specifikacije*⁸⁰ so torej tekstovni opisi semantike določenega elementa in dodajajo pomen modelu.

⁸⁰ angleško: specifications

Prikladnost jezika UML je, da je vsak element predstavljen z enostavnim simbolom, ki mu lahko, če želimo na diagramu prikazati več informacij, pripišemo številne *dodatke*⁸¹. Ker je diagram le pogled na model, se ponavadi omejimo le na dodatke, ki povečajo njegovo jasnost in berljivost.

V UML je uporabljen tudi mehanizem *ločevanja*⁸². Pri objektno orientiranem modeliranju razdelimo pogled na realen svet na več načinov. Ločimo med razredom in primerkom ter vmesnikom in implementacijo. *Razred* je abstrakten zapis določenega tipa, medtem ko *primerke* predstavlja točno določeno stvar (primer avto, moj avto). Druga delitev temelji na principu ločevanja med tistim, kar nekaj naredi (*vmesnik*), in načinom, kako se to izvede (*implementacija*). Gumb na televiziji je primer enostavnega vmesnika do kompleksnega notranjega mehanizma.

Da bi bil UML čim bolj univerzalen jezik za modeliranje, so vanj vključeni tudi trije *mehanizmi razširitve*⁸³:

- Omejitve⁸⁴ razširjajo semantiko elementov z vpeljevanjem novih pravil⁸⁵.
- Stereotipi⁸⁶ omogočajo vpeljavo novega elementa modeliranja, ki temelji na že obstoječem.
- Z uporabo dodatnih vrednosti⁸⁷ lahko elementom modela pripišemo nove lastnosti, takšne, ki jih jezik UML ne predvideva. Nekatere predstavljajo le dodatno informacijo o elementu in so z njim povezane preko stereotipov.

5.1.3. Arhitektura

Vizualizacija, opredelitev, konstrukcija in dokumentacija sistema zahteva dobro poznavanje le-tega iz različnih perspektiv. Arhitektura sistema zahteva niz pomembnih odločitev o:

- organiziranosti programske opreme sistema,
- izboru strukturnih elementov in vmesnikov, iz katerih je sistem sestavljen, ter njihovem obnašanju,
- kompoziciji strukturnih elementov in elementov obnašanja v podsisteme in
- stilu arhitekture, ki oblikuje to organizacijo: statični in dinamični elementi, vmesniki, sodelovanje med njimi in kompozicija v celoto.

⁸¹ angleško: adornments

⁸² angleško: common divisions

⁸³ angleško: extensibility mechanisms

⁸⁴ angleško: constraints

⁸⁵ angleško: rules

⁸⁶ angleško: stereotypes

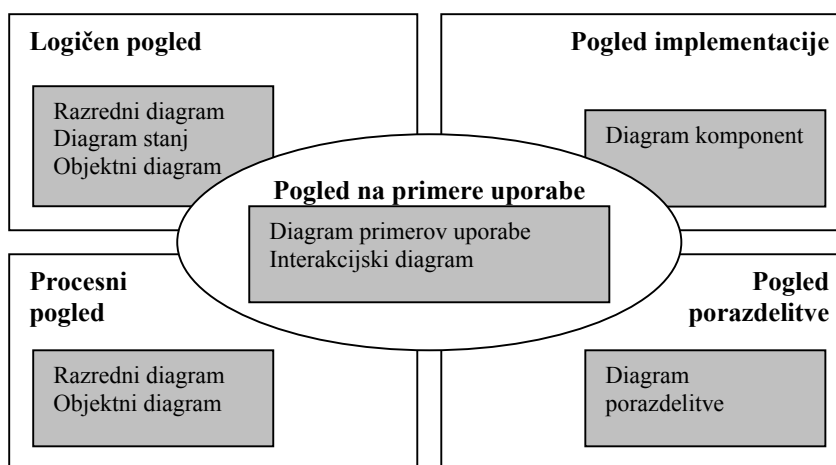
⁸⁷ angleško: tagged values

Arhitektura programske opreme skrbi tudi za uporabnost, funkcionalnost, izvedljivost, ekonomičnost, ponovno uporabnost in estetiko.

Da bi popolnoma zaobjeli kompleksen sistem, definira UML štiri različne poglede nanj (logičen, procesni, implementacijski pogled in pogled razvrstitve), ki so skupaj združeni s petim, to je s pogledom na primer uporabe.

Z *logičnem pogledom* predstavimo objekte in razrede, ki sestavljajo sistem. Z njim predstavimo funkcionalne zahteve sistema oziroma storitve, ki naj bi jih ta nudil končnim uporabnikom. *Procesni pogled* zaobjema procese in korake, ki sinhronizirajo celoten mehanizem. Pogled je usmerjen na izvedbo, obseg in zmožnosti sistema. S *pogledom implementacije* predstavimo komponente, ki sestavljajo sistem. Usmerjen je na konfiguracijo celotnega sistema. *Pogled razvrstitve* nam prikaže topologijo vozlišč in opiše distribucijo ter namestitev delov, ki sestavljajo fizično zgradbo sistema. Vsi štirje omenjeni pogledi izhajajo iz *pogleda na primere uporabe*, ki predstavlja zbirko vseh osnovnih zahtev, zbranih v nizu primerov uporabe, in predstavlja osnovo ostalim pogledom. Primeri uporabe opisujejo obnašanje sistema s strani končnih uporabnikov in analitikov.

Slika 25: '4+1 pogled' arhitekture



Vir: Arlow, Neustadt, 2002, str. 19

5.1.4. Poenoten proces

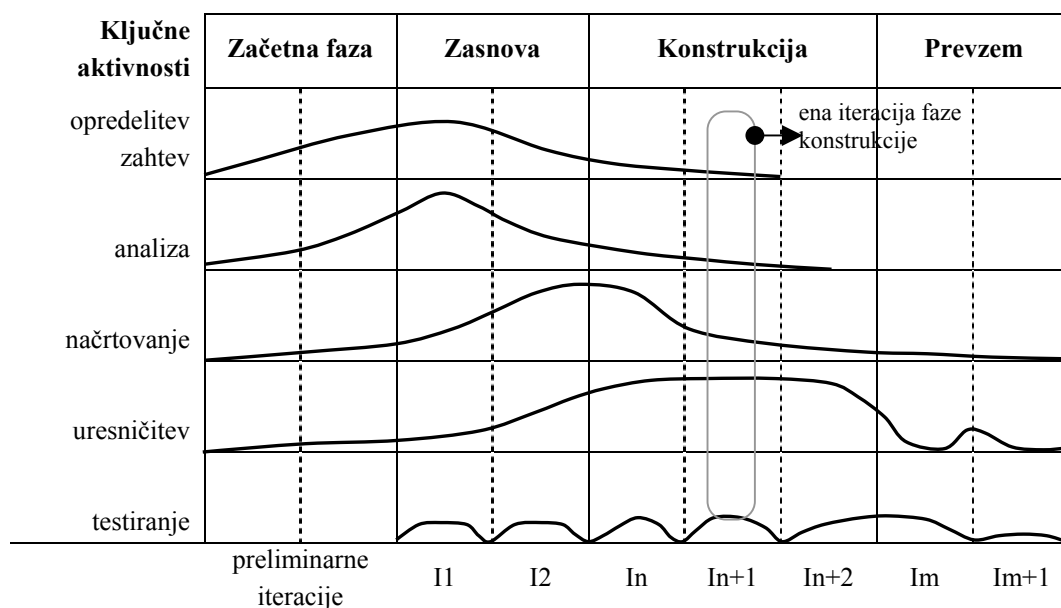
Poenoten proces razvoja programske opreme⁸⁸ ali na kratko Poenoten proces so izdelali avtorji jezika UML. Skupaj s jezikom UML tvorita metodologijo. Medtem ko predstavlja UML vizualni del obravnavanega problema, je s poenotenim procesom, kot pove že ime samo, zaobjet procesni del. Poenoten proces je iterativen in inkrementalen proces, kar pomeni, da lahko razdelimo projekte na manjše 'mini projekte', ki jih lažje obvladamo in

⁸⁸ angleško: the unified software development process

uspešno zaključimo. Vsak od teh 'mini projektov' predstavlja iteracijo ali ponovitev. Rezultat vsake ponovitve je nov prirastek⁸⁹ izdelkov, ki opisujejo delovanje sistema in omogočajo prehod v naslednjo ponovitev. Časovno je proces razvoja razdeljen na štiri faze razvoja (glej sliko na naslednji strani). Vsaka faza se zaključi z mejnikom⁹⁰, to je točko, ko so izdelani predpisani modeli ali dokumenti posamezne faze. Mejnik je namenjen kritičnemu pregledu dosedanjega dela in sprejetju pomembnih odločitev preden nadaljujemo z naslednjo fazo. Nadzorujemo lahko napredek in ocenimo, koliko časa je potrebno za zaključitev posamezne faze, kar lahko s pridom uporabimo pri bodočih projektih.

V **začetni fazi**⁹¹ se opredeli namen in cilj projekta oziroma razvoja načrtovanega sistema. Funkcionalnosti, ki jih zahtevamo od načrtovanega sistema, prikažemo s primeri uporabe. Podrobneje opredelimo le najvažnejše in identificiramo akterje, ki bodo s sistemom sodelovali. Ocenimo obsežnost dela, stopnjo tveganja ter stroške projekta, opredelimo potrebne vire in pripravimo začetni načrt projekta. Na koncu faze se preučijo *cilji življenjskega cikla*⁹² projekta, na podlagi katerih se odločimo za nadaljevanje ali zavrnitev projekta.

Slika 26: Faze in aktivnosti poenotenega procesa



Vir: Arlow, Neustadt, 2002, str. 32

Cilj druge faze, faze **zasnove**⁹³, je podrobna analiza področja problema. Določimo osnovno arhitekturo sistema, izdelamo plan projekta in ponovno ocenimo tveganje. Odpravimo tveganja, ki bi se pojavila zaradi nerazumevanja med razvijalcem in uporabnikom, zaradi

⁸⁹ sopomenka: inkrement

⁹⁰ angleško: milestone

⁹¹ angleško: inception

⁹² angleško: life cycle objectives

⁹³ angleško: elaboration

neustrezne tehnološke opreme za rešitev ali neustrezne usposobljenosti za izvedbo projekta. Osnovna arhitektura sistema temelji na dobrem poznavanju sistema kot celote, zato določimo v fazi zasnove vse primere uporabe in jih večino podrobneje obdelamo ter analiziramo. Na koncu faze se na podlagi opredeljenih ciljev sistema, izbrani arhitekturi sistema in oceni tveganj odločimo, ali bomo nadaljevali proces v naslednjo fazo.

V fazi **konstrukcije**⁹⁴ se sistem iterativno in inkrementalno dograjuje. Osnova za delo so primeri uporabe, zato mora biti v tej fazi podrobneje opredeljen in analiziran še preostanek primerov uporabe. Postopoma sistemu dodajamo njegovo funkcionalnost, ga testiramo in integriramo dokler ne dosežemo ustreznega nivoja kakovosti sistema. Pomemben del faze predstavlja testiranje sistema, pri katerem sodelujejo tudi njegovi končni uporabniki. Sproti lahko preverjajo funkcionalnost sistema ter predlagajo možne izboljšave in popravke. Izdelava se tudi tehnična in uporabniška dokumentacija. Na koncu faze ocenimo, če je izdelana programska oprema pripravljena za prevzem.

Zadnja faza Poenotenega procesa je faza **prevzema**⁹⁵. Izdelek namestimo v uporabnikovo okolje. Izvedemo še zadnja testiranja, meritve zmogljivosti sistema in izobrazimo uporabnike za njegovo vsakodnevno uporabo. Po predaji izdelka in zagonu v uporabniškem okolju se lahko pojavijo potrebe po dodatnih popravkih. Če so te manjše, jih opravimo v fazi prevzema. Ob koncu faze ocenimo, če so doseženi cilji življenjskega cikla izdelanega sistema. V nasprotnem primeru pričnemo z novim ciklom razvoja. Izkušnje, ki smo jih tekom razvoja sistema pridobili, uporabimo pri bodočih projektih.

Poenoten proces je, kot smo omenili, iterativen, zato se posamezna faza procesa lahko deli naprej na več ponovitev ali iteracij. Vsaka iteracija predstavlja 'mini projekt' zase, znotraj katerega izvedemo zaporedje petih aktivnosti (glej sliko 27 na str. 46):

- opredelitev zahtev⁹⁶,
- analiza,
- načrtovanje,
- implementacija in
- testiranje.

Namen prve aktivnosti je **opredelitev zahtev** bodočega sistema. Napravi se seznam vseh idej, ki jih predlagajo uporabniki sistema, analitiki in njegovi načrtovalci. To je lista možnih zahtev, ki jih kasneje, odvisno od izbora, vključimo v model sistema in prikažemo s primeri uporabe. Definirati je potrebno namen načrtovanega sistema in na podlagi tega izbrati ustrezne zahteve. Zahteve delimo na *funkcionalne*, ki podajajo, kaj bo sistem delal, in

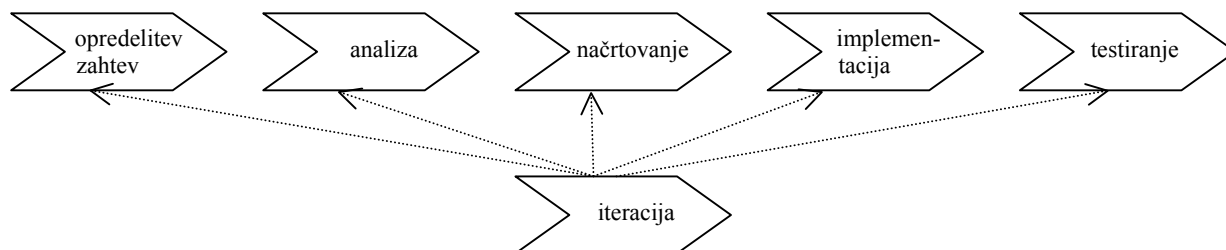
⁹⁴ angleško: construction

⁹⁵ angleško: transition

⁹⁶ angleško: requirements

nefunkcionalne zahteve, s katerimi navedemo omejitve, na primer odzivnost sistema. Večino zahtev opredelimo v začetni fazi in fazi zasnove (glej sliko 26 na str. 44), v fazi prehoda pa le, če je prišlo do spremembe zahtev.

Slika 27: Zaporedje aktivnosti iteracije



Vir: Jacobson et al., 1999, str. 99

Druga aktivnost, **analiza**, je namenjena izdelavi modela, ki vsebuje bistvene zahteve in karakteristike želenega sistema. Aktivnost opredelitev zahtev in analize potekata skoraj sočasno, saj je priporočljivo sproti analizirati pojavljajoče se zahteve in s tem sproti odpravljati pomanjkljivosti. Osredotočena je na opredelitev razredov, njihovo analizo in na dejansko realizacijo primerov uporabe. Opredelijo se razredi, njihovo obnašanje, povezave med njimi, vmesniki in paketi, ki zagotavljajo potrebno funkcionalnost sistema. Model analize je predstavljen z razrednim in interakcijskim diagramom. Ne vsebuje podrobnosti uresničitve.

Medtem ko je analiza osredotočena na izdelavo logičnega modela, s katerim je predstavljena funkcionalnost sistema, se aktivnost **načrtovanja** loteva problema, kako bo ta funkcionalnost implementirana. Načrtovanje postane prevladujoča aktivnost modeliranja ob koncu faze zasnove in v prvi polovici faze konstrukcije. Model analize prevedemo v model načrta, v model sistema, ki vsebuje podrobnosti, potrebne za implementacijo. Model načrta sestavljajo načrti podsistemov, razredov, vmesniki, realizacije primerov uporabe in zasnova⁹⁷ diagrama razvrstitve. Opredelimo vse razrede sistema in jim določimo attribute, vidljivost⁹⁸, če je potrebno, prednastavljeno vrednost⁹⁹ ter operacije prevedemo v metode. Opredelimo povezave med razredi, jim določimo števnost in smer. Z diagrami interakcij, stanj in aktivnosti prikažemo dinamično obnašanje sistema.

Aktivnost **implementacije** se prične proti koncu faze zasnove in je prevladujoča v fazi konstrukcije. Namen implementacije je prevesti model načrta v programsko kodo. Razrede razporedimo v komponente in poskrbimo za realizacijo vmesnikov. Model implementacije sestavljata diagram komponent, v katerem prikažemo povezave med komponentami, ki

⁹⁷ angleško: first-cut

⁹⁸ angleško : visibility

⁹⁹ angleško: default value

sestavljajo sistem, in diagram razvrstitve, ki prikazuje fizično razporeditev vozlišč in povezave med njimi.

V aktivnosti **testiranja** preverimo pravilno delovanje sistema. Integracijski test opravimo za vsak dodatek, ki smo ga izdelali znotraj iteracije, systemskega pa ob zaključku iteracije. Osnovni izhod testiranja je model testiranja, ki opisuje, kako je potekalo testiranje sistema. Sestavljen je iz testnih primerov, ki opredeljujejo, katere dele sistema je potrebno testirati, testnih procedur, ki opredeljujejo izvedbo testnih primerov, in testnih komponent, ki avtomatizirajo testne procedure (Jacobson, Booch, Rumbaugh, 1999, str. 313).

5.2. Tabelarni razvoj aplikacij

Tudi metodologija »Tabelarni razvoj aplikacij¹⁰⁰« (v nadaljevanju TAD) je objektno orientirana metodologija namenjena razvoju informacijskih sistemov. Namesto diagramov uporablja za prikaz delovanja organizacije različne tabele. In zakaj ravno tabele? T. Damij (2001) našteva vsaj tri razloge:

- tabele so uporabne za predstavitev dogodkov,
- tabele se na enostaven način kreirajo, popravljajo, razširjajo in
- tabele so zelo pregledne in razumljive.

Metodologijo TAD sestavlja šest faz, ki sledijo ena drugi v naslednjem vrstnem redu:

1. definicija problema,
2. analiza delovanja sistema,
3. prenova poslovnega procesa,
4. razvoj objektnega modela,
5. načrtovanje sistema,
6. implementacija sistema.

5.2.1. Definicija problema

Prva faza metodologije TAD je namenjena identifikaciji in opredelitvi obravnavanega problema. V ta namen organiziramo niz intervjujev z vsemi uporabniki sistema. V tej fazi se omejimo na predstavnike vodstva. Prvi so na vrsti predstavniki najožjega vodstva podjetja¹⁰¹, saj potrebujemo podatke o strateških planih in ciljih podjetja. Cilj intervjujev najožjega vodstva podjetja je torej identifikacija (T. Damij, 2001, str. 21):

¹⁰⁰ angleško: Tabular Application Development

¹⁰¹ angleško: top management

- strateških planov in ciljev podjetja,
- vitalnih izhodov in analiz povezanih s strateškimi plani in cilji,
- problemov pri podpori odločanju in
- strukturne organizacije podjetja.

Ko zberemo te podatke, ponavadi sta potrebna dva ali več srečanj, organiziramo intervjuje še z vodilnimi predstavniki na poslovni in operativni ravni odločanja in raziščemo, kako se strateški cilji uresničujejo v poslovnih in operativnih ciljih. Tako podrobneje spoznamo obravnavani problem in delovanje celotne organizacije ter na koncu pridobimo znanje še o:

- poslovnih ciljih,
- operativnih ciljih,
- celotni strukturi posameznega oddelka ali organizacijske enote (entitete),
- vitalnih izhodih in analizah povezanih s poslovnimi in operativnimi cilji ter
- problemih pri podpori odločanja za potrebe posameznih entitet.

V metodologiji TAD uporabljamo besedo *entiteta* za opis uporabnika, skupine uporabnikov, enote ali oddelka organizacije ali česarkoli, kar je pomembno za delovanje sistema (T. Damij, 2001, str. 22). Predstavlja vir informacij, ki je lahko sama del sistema (notranja entiteta) ali pa je z njim le v interakciji (zunanja entiteta). Pri intervjujih se omejimo le na notranje entitete. Preko njih se informiramo tudi o obnašanju zunanjih.

Rezultate prve faze metodologije zberemo in predstavimo s *tabelo entitet*, ki vsebuje opredelitve entitet, njihovih ciljev, analiz ter odločitvenih problemov. V stolpcih tabele so nanizane entitete, ki so z zvezdico (*) v ustreznem polju presečišča vrstice (i) in stolpca (j) povezane z analizami, pomembnimi za njihovo delo in so vsebovane v vrsticah tabele.

Tabela 3: Primer tabele entitet

		Index	1	2	3	4	5	6	7
Req.	Index	Entity Analysis	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VSZS	VUZS	Direktor
Report	1	Analizirani potresi	*	*	*	*	*		
	2	Informacije o izpisih			*	*	*		
	3	Informacije o delovanju opazovalnic	*	*			*	*	
	4	Informacije o delovanju mreže	*	*			*		
	5	Informacije o intenziteti potresov	*	*	*		*	*	*
	6	Informacije o izdanih poročilih o potresu	*		*		*		
	7	Informacije o lokaciji potresov	*	*			*	*	*
	8	Informacije o nastali škodi	*	*			*	*	*
DS	9	Predpisi						*	*
	10	Seizmološki modeli					*	*	

5.2.2. Analiza delovanja sistema

Fazi definicije problema sledi analiza delovanja sistema. To je najpomembnejša faza metodologije TAD, saj predstavljajo njeni rezultati temelj oziroma vhodne podatke vsem njej sledečim fazam. Tu podrobneje raziščemo delovanje sistema in ga poskušamo čim bolje razumeti, saj le dobro razumevanje le-tega lahko vodi k uspešni izgradnji informacijskega sistema ali njegovi učinkoviti prenovi. Dobiti moramo čisto sliko o delovanju sistema kot celote, kot tudi vsakega njegovega posameznega dela. V ta namen nadaljujemo z intervjuji. Ker je v tem primeru naš cilj identifikacija posameznih aktivnosti¹⁰² in nalog¹⁰³ v sistemu, so tokrat naša ciljna skupina predstavniki posameznih entitet.

Faza analize delovanja sistema ima tri korake. V prvem *identificiramo aktivnosti sistema*, temu sledi *opredelitev nalog v okviru opredeljenih aktivnosti*, na koncu pa *združimo aktivnosti v delovne in poslovne procese*. Tudi tu predstavljajo rezultat našega dela tabele. To sta tabela aktivnosti in tabela nalog.

5.2.2.1. Aktivnosti

Vse aktivnost in dogodke, ki se pojavijo v sistemu, predstavimo s *tabelo aktivnosti*. Ta je organizirana tako, da predstavljajo entitete sistema (notranje in zunanje) stolpce tabele, aktivnosti pa nanizamo v njene vrstice. Zvrstimo jih tako, da si sledijo v enakem vrstnem redu kot v realnem svetu. Da je preslikava realnega sveta še popolnejša, jih med seboj horizontalno in vertikalno povežemo.

Horizontalno povežemo vse entitete, ki sodelujejo pri izvajanju določene aktivnosti. V ta namen vpeljemo dve oznaki:

1. $S = S(i, j)$ = entiteta v stolpcu j je *izvorna*¹⁰⁴ entiteta aktivnosti v i -ti vrstici.
2. $T = T(i, j)$ = entiteta v stolpcu j je *ciljna*¹⁰⁵ entiteta aktivnosti v i -ti vrstici.

Vsaka aktivnost ima eno ali več izvornih oziroma ciljnih entitet, zato moramo oznaki S in T ustrezno indeksirati. Indeks je kar zaporedna številka stolpca izvirne entitete v tabeli aktivnosti (glej tabelo 4 na str. 50).

Vertikalno povežemo aktivnosti, ki jih opravlja ista entiteta glede na časovno zaporedje izvajanja. Za to povezavo vpeljemo dve novi oznaki:

1. $P = P(i, j)$ = aktivnost v vrstici i je *predhodnik*¹⁰⁶ naslednje aktivnosti, ki jo bo izvedla entiteta v stolpcu j .

¹⁰² Aktivnost je skupek nalog, povezanih z določenim vhom in izhodom, ki jo izvaja ena ali več entitet (T. Damij, 2001, str. 27)

¹⁰³ Naloga (angl. task) je elementarno delo, ki ga izvaja določena entiteta v okviru neke aktivnosti (T. Damij, 2001, str. 27)

¹⁰⁴ angleško: source (S)

¹⁰⁵ angleško: target (T)

2. $U = U(i, j)$ = aktivnost v vrstici i je *naslednik*¹⁰⁷ aktivnosti, ki jo je že izvedla entiteta v stolpcu j .

Ker želimo poznati zaporedje aktivnost določene entitete, tudi oznaki P in S indeksiramo, in sicer kar po zaporedni številki vrstice predhodnika.

Tabela 4: Prikaz dela tabele aktivnosti

	Entity Index	Internal						
		1	2	3	4	5	6	7
Index	Activity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VSZS	VUZS	Direktor
1	Sprejem obvestila	T10 P1						
2	Pregled zapisov	U1, P2						
3	Analiza digitalnih seizmogramov	S1, T1 U2, P3						
4	Naroči VOY	S1 U3, P4						
5	Dodaj VOY	T8 U4, P5						
6	Lociranje potresa	S1, T1 U5, P6						
7	Potrdi analizo	S1, T1 U6, P7						
8	Kreiranje izpisa	S1, T1 U7, P8						
9	Pošiljanje izpisa	S1 U8, P9	T1	T1 P9		T1		
10	Kreiranje obvestila	S1, T1 U9, P10						
11	Pošiljanje obvestila	S1 U10	T1	T1 P11		T1	T1	T1
12	Kreiranje vprašalnikov			S3 U9, U11		T3 P12		

5.2.2.2. Naloge

Tabelo nalog razvijamo sočasno s tabelo aktivnosti. Vsaka aktivnost je povezana preko naloge z vsaj eno entiteto. V vsakem nepraznem presečišču vrstice (i) in stolpca (j) definiramo posamezno nalogo entitete iz stolpca j v okviru aktivnosti v vrstici i . Označimo jo s simbolom K_{ij} , kjer je K naloga definirana v okviru določene aktivnosti, i in j pa sta indeksa aktivnosti in entitete tabele aktivnost, ki označujeta nalogo. V tabeli nalog za vsako od njih navedemo :

- kratek opis naloge,
- čas potreben, da entiteta (j) opravi nalogo,
- pogoje, ki morajo biti izpolnjeni za izvedbo naloge, ter
- vhodne elemente, ki so potrebni za izvedbo naloge, in izhodne elemente, ki so rezultati opravljene naloge.

¹⁰⁶ angleško: predecessor (P)

¹⁰⁷ angleško: successor (U)

Če se nam zdi potrebno, lahko v tabelo nalog vključimo tudi druge lastnosti.

V tabeli nalog zavzema vsaka naloga po eno vrstico, aktivnosti, ki so praviloma sestavljene iz več nalog, pa združujejo temu ustrezno število vrstic.

Tabela 5: Del tabele nalog

Activity	Task Code	Description	Time	Condition	Input/Output
9 - Pošiljanje izpisa	K(9,1)	Dežurni seizmolog potrdi izpis in ga pošlje seizmologu, M-seizmologu, VSZS in svetovnim seizmološkim centrom.		Lokacija potresa?	Izpis potresa
	K(9,2)	Seizmolog prejme izpis potresa.			Izpis potresa
	K(9,3)	M-Seizmolog prejme izpis potresa. Ta mu je v pomoč pri kreiranju vprašalnikov o učinkih potresa.			Izpis potresa
	K(9,5)	VSZS prejme izpis potresa.			Izpis potresa
10 - Kreiranje obvestila	K(10,1)	Dežurni seizmolog kreira poročilo o potresu.			Poročilo o potresu
11 - Pošiljanje obvestila	K(11,1)	Dežurni seizmolog poročilu doda podatek, kje so potres čutili, in oceno intenzitete ter ga posreduje CORS-u, medijem, VSZS, VUZS, direktoju, seizmologu in M-seizmologu.		Lokacija potresa? Informacija o intenziteti?	Obvestilo, poročilo o potresu
	K(11,2)	Seizmolog prejme poročilo o potresu.			Poročilo o potresu
	K(11,3)	M-seizmolog prejme poročilo o potresu. Na podlagi le-tega kreira vprašalnike o učinkih potresa.			Poročilo o potresu
	K(11,5)	VSZS prejme poročilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(11,6)	VUZS prejme obvestilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(11,7)	Direktor prejme obvestilo o potresu in ustrezno ukrepa.			Poročilo o potresu
12 - Kreiranje vprašalnikov	K(12,3)	M-seizmolog pripravi vprašalnike o učinkih potresa na ljudi, poslopja, naravo in jih posreduje VSZS v potrditev.		Lokacija, magnituda potresa, število klicev	Vprašalnik
	K(12,5)	VSZS prejme vprašalnik v potrditev.			Vprašalnik

5.2.2.3. Delovni in poslovni procesi

V tretjem koraku, ko smo zaključili intervjuje z uporabniki ter izdelali tabeli aktivnosti in nalog, opredelimo še delovne in poslovne procese. *Delovni proces* je zbirka ene ali več aktivnosti, ki si sledijo v določenem vrstnem redu, zbirka enega ali več delovnih procesov pa je *poslovni proces*.

Po združevanju aktivnosti v delovne in poslovne procese se zopet posvetujemo s predstavniki notranjih entitet in se prepričamo, da je vsebina tabel pravilna in da korektno prikazuje obravnavani sistem.

5.2.3. Prenova poslovnih procesov

Po zaključku druge faze, faze analize delovanja sistema, smo dobro seznanjeni z delovanjem sistema. Prišli smo do točke, ko lahko razmislimo, če je to učinkovito in če nas bo pripeljalo do zelenih ciljev. Premisliti moramo, ali bi bilo smotno kakšno stvar spremeniti. Tega se lotimo v fazi prenove poslovnih procesov. Pri tem nam pomagajo podatki, ki jih imamo

zbrane v tabelah entitet, aktivnosti in nalog. Prenova poslovnih procesov poteka v dveh korakih. Najprej *analiziramo in prenovimo tabele entitet, aktivnosti in nalog*, nato *prevedemo vsebino tabele aktivnosti v procesni model sistema*. Če ocenimo, da prenova ni potrebna, preskočimo prvi korak in izdelamo procesni model sistema na podlagi obstoječe tabele aktivnosti.

5.2.3.1. Analiza in prenova tabel

Korektno pripravljene tabele entitet, aktivnosti in nalog izražajo dejansko stanje in funkcioniranje organizacije. Z njihovo podrobno analizo lahko predlagamo spremembe in s tem izboljšamo delovanje. V ta namen ustanovimo projektno skupino¹⁰⁸. Zaradi pomembnosti naloge jo sestavljajo analitik, predstavniki ožjega vodstva podjetja ter predstavniki vodilnih na poslovni in operativni ravni. Skupaj analizirajo cilje in plane podjetja na vseh ravneh in predlagajo spremembe. V skladu z njimi se spremeni tudi vsebina tabel. V procesu prenove se analizirajo strateški cilji, poslovni in operativni procesi, vse aktivnosti in naloge z različnih zornih kotov. Predlagajo in implementirajo se spremembe in popravki, odstranijo se nepotrebne aktivnosti in naloge, če je potrebno se zaprejo nepotrebne oz. odprejo nove entitete (oddelki, enote), aktivnosti in naloge se zaradi boljše in lažje izvedbe prenesejo z ene na drugo entiteto. Proces analize in prenove je iterativen proces, ki potrebuje nekaj ponovitev, saj lahko vsaka dodatna analiza prinese kamenček več k boljši rešitvi.

5.2.3.2. Procesni model

Pretvorba tabele aktivnosti v procesni model je neobvezen korak, a nam ponuja boljšo predstavbo o povezavah med nalogami, saj je model predstavljen z uporabo diagrama tokov dogodkov¹⁰⁹. Vsako neprazno presečišče tabele aktivnosti notranjih entitet pretvorimo v elementaren proces (označimo ga s krogom), ki predstavlja nalogo K_{ij} (glej str. 50) in oba indeksa vpišemo v krog. Za zunanje entitete neprazno presečišče spremenimo v izvor, označimo ga s pravokotnikov in vanj vpišemo ime entitete (j). Da dokončamo model, moramo izvore in procese med seboj še navpično in vodoravno povezati. Vodoravno povežemo s puščicami izvirne S_i in ciljne T_i entitete, kjer je i indeks izvirne entitete, puščica pa je usmerjena od izvirne proti ciljni entiteti. Navpično povežemo predhodno aktivnost P_i z njenim naslednikom U_i , kjer je i indeks predhodnika, puščica pa je usmerjena od predhodnika k nasledniku. Tako dobimo elementarni diagram toka dogodkov, v katerem označimo še meje delovnih in poslovnih procesov (glej sliko 33 na str. 73). Lahko zrišemo tudi diagrame toka dogodkov na drugih ravneh (aktivnostni ravni, ravni delovnih procesov, ravni poslovnih procesov ali na ravni celotnega sistema).

¹⁰⁸ angleško: team

¹⁰⁹ angleško: data flow chart (DFD)

5.2.4. Razvoj objektnega modela

Četrta faza metodologije TAD je namenjena razvoju objektnega modela sistema in je sestavljena iz šestih korakov. Njeni vhodni podatki so, kot smo omenili na začetku razdelka 5.2.2., tabele entitet, aktivnosti in nalog. Poseben poudarek je na slednji oziroma na njenem delu, kjer imamo opisane vhode in izhode posamezne naloge, navadno v obliki dokumentov. Te podrobno analiziramo.

Prvi korak:

Identificiramo vse funkcionalne odvisnosti, ki obstajajo med ključnimi atributi (to so ponavadi šifre dokumentov, artiklov,...) in ostalimi atributi na dokumentu (glej sliko 35 na str. 75).

Drugi korak:

Analiziramo povezave, ki obstajajo med posameznimi ključnimi atributi, in določimo tipe oziroma števnost povezav (1:1, 1:n, m:n). Povezave v razmerju m:n nadomestimo z dvema povezavama v razmerju 1:n preko asociacijskega razreda (glej sliko 36 na str. 76).

Tretji korak:

Ko končamo analizo vseh obravnavanih dokumentov, te analize integriramo. Skupaj grupiramo funkcionalne odvisnosti, ki so povezane z istimi ključnimi atributi (glej sliko 37 na str. 77).

Četrty korak:

Zamenjamo imena ključnih atributov z imeni njihovih razredov in povežemo vse attribute, vključno s ključnim, z ustreznimi razredi. Definiramo semantično strukturo mrežnega modela tako, da povezave pretvorimo v povezave strukture *Je-povezan-z* in *Je-del* glede na vsebino razmerja (glej sliko 38 na str. 77).

Peti korak:

Rezultat predhodnih korakov pretvorimo v začetni objektni model obravnavanega sistema (glej sliko 39 na str. 79).

Šesti korak:

Začetni objektni model dopolnimo z razredi, ki jih identificiramo med ponovno analizo tabel aktivnosti, nalog in entitet, pa tudi s pomočjo izkušenj analitika. Hierarhično, z uporabo mehanizma dedovanja (povezava strukture *Je*), uredimo razrede začetnega objektnega modela, ki imajo skupne attribute ali povezave. Uporabimo tako generalizacijo, združevanje razredov v nadrazred, kot tudi specializacijo, delitev razreda na podrazrede (glej sliko 40 na str. 80).

5.2.5. Načrt sistema

V prvih štirih fazah metodologije TAD smo analizirali sistem ter mu določili funkcionalno in statično strukturo. V peti fazi se spopademo z načrtom sistema in pripravimo sistem za implementacijo. V prvem koraku *opredelimo operacije*, nato v drugem *izdelamo aplikacijski model*.

Pri opredelitvi operacij si zopet pomagamo s tabelo nalog, zapisujemo pa jih v *tabelo operacij*¹¹⁰. Za vsako zapišemo v tabelo *kodo naloge* (K_{ij}), ki vsebuje operacijo, *ime operacije* in *razrede*, v katerih nastopa. Vsaki operaciji pripada ena vrstica v tabeli operacij, ker pa je lahko vsaka naloga povezana z več operacijami, pripada vsaki nalogi temu številu ustrezno število vrstic v isti tabeli. Ko opredelimo vse operacije, jih dodamo še v končni objektni model in jim definiramo algoritme.

Aplikacijski model temelji na podatkih, zbranih v tabelah entitet in aktivnosti. Prva dva dela sta sestavljena iz poročil in odločitvenih problemov, ki smo jih zbrali v tabeli entitet. Tretji del razberemo iz prvega stolpca tabele aktivnosti. Tako dobljeni aplikacijski model nam poda pregled zahtev, odločitvenih problemov in shemo procesov ter aktivnosti (glej tabelo 16 na str. 82).

5.2.6. Implementacija sistema

Zadnja faza metodologije TAD je faza implementacije. Njeni vhodni podatki so objektni in aplikacijski model ter algoritmi, povezani z njima. Izberemo ustrezno podatkovno bazo in programski jezik ter algoritme, dobljene v peti fazi, ki definirajo aplikacijski model in operacije vseh razredov, prevedemo v programsko kodo.

¹¹⁰ angleško: operation table

6. Razvoj seizmološkega informacijskega sistema

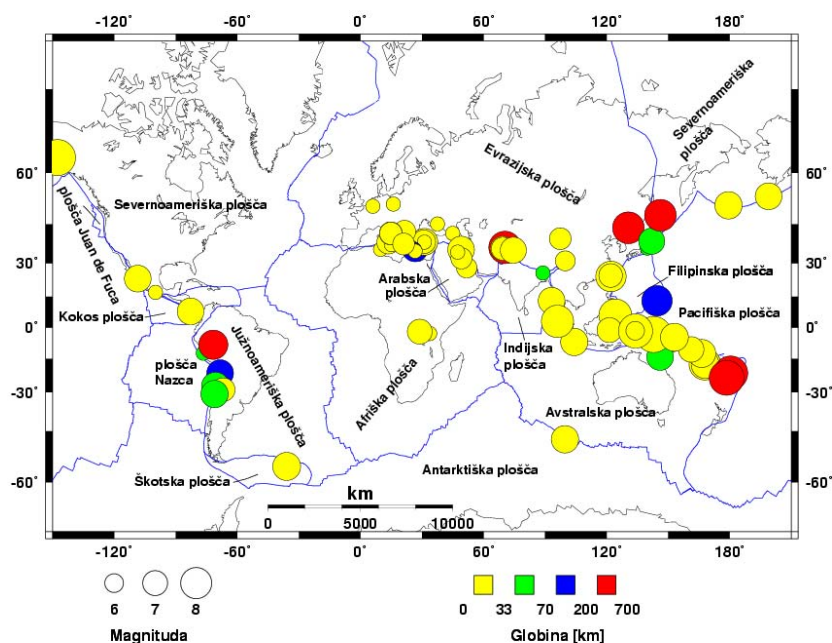
Namen poglavja je predstavitev uporabe metodologije TAD pri razvoju seizmološkega informacijskega sistema. V prvem razdelku poglavja bo predstavljeno delo Urada za seizmologijo. Nato bomo obravnavali prvih pet faze izgradnje sistema po metodologiji TAD. Faza implementacije je v tem delu le omenjena, saj bi potrebovali več časa, sredstev in ljudi, ki bi sodelovali pri njeni dejanski realizaciji.

6.1. Predstavitev Urada za seizmologijo

6.1.1. Potresi

Vsako leto strese Zemljo nekaj sto tisoč potresov, ki presegajo magnitudo 2,0. Njihova žarišča so razporejena predvsem na stikih večjih geotektonskih plošč. Potresno najbolj dejavnejši sta obtihoceanski in sredozemsko-himalajski pas. Slednji vključuje tudi naše kraje. Ob potresih se sprosti velika količina energije, ki se širi kot seizmično valovanje v vse smeri po Zemljini notranjosti in njeni površini.

Slika 28: Porazdelitev najmočnejših potresov leta 2002 na Zemlji, njihova magnituda in globina sta podani z velikostjo in barvo krožca.

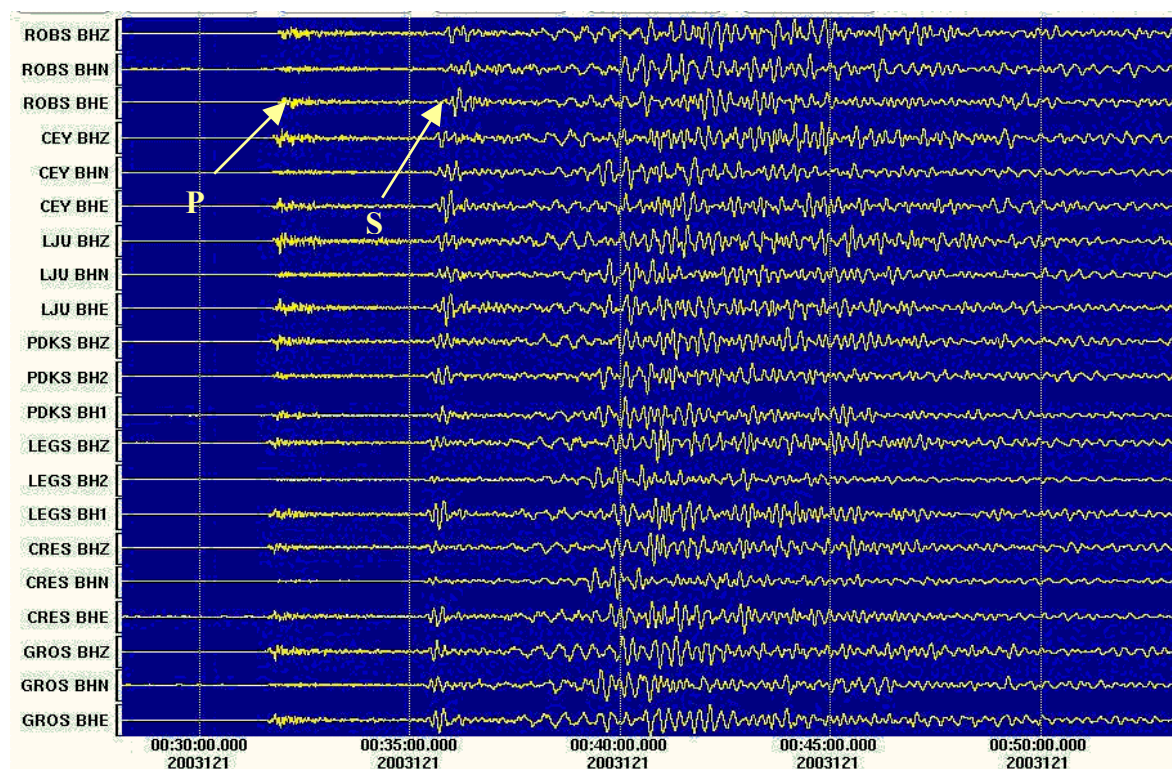


Vir: Seizmološki bilten Urada za seizmologijo, 2002

Ločimo dva razreda seizmičnih valov. Eni potujejo po Zemljini površini, drugi se širijo skozi njeno notranjost. Slednji so hitrejši in jih delimo naprej na dve skupini glede na smer nihanja:

vzdolžni in prečni valovi. Vzдолžno valovanje potuje hitreje kot prečno in ga zato imenujejo tudi primarno ali P valovanje. Prečno valovanje imenujejo sekundarno ali S valovanje. Na potresne opazovalnice vedno pride najprej valovanje iz skupine P valov. Katero, je odvisno od lokacije, globine potresa ter oddaljenosti od opazovalnice (Kulháněk, 1990, str. 13 - 19).

Slika 29: Primer zapisov potresa (Turčija, 1.5.2003 ob 00:27UTC) z označenimi vstopi skupine primarnih (P) in sekundarnih (S) valov



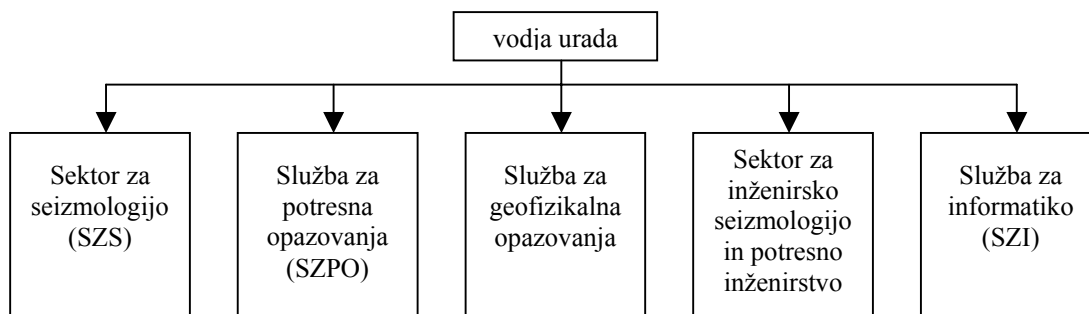
Vir: ARSO, urad za seizmologijo, 2003

6.1.2. Urad za Seizmologijo

Urad za seizmologijo je del v letu 2001 nastale Agencije Republike Slovenije za Okolje (v nadaljevanju ARSO), ki je organ v sestavi Ministrstva Republike Slovenije za okolje, prostor in energijo. Po pravilniku o notranji organizaciji in sistemizaciji mest ARSO so njegove naloge neprekinjeno spremljanje in vrednotenje potresne dejavnosti na ozemlju Slovenije in izven nje ter obveščanje ustreznih državnih organov, sredstev obveščanja in javnosti o pomembnejših dogodkih. Skrbi tudi za ustrezno mednarodno izmenjavo podatkov s svetovnimi seizmološkimi centri, načrtuje in vzpostavlja stalne potresne opazovalnice in mreže opazovalnic na območju Slovenije ter po potrebi namešča začasne potresne opazovalnice. Po potresih izvaja makroseizmične raziskave, zbira in obdeluje zgodovinske podatke o potresih ter izdeluje kataloge potresov. Zbira in preučuje podatke o neotektonskih, geoloških, geofizikalnih in podobnih pojavih, povezanih s potresno dejavnostjo. Opravlja tudi

naloge na področju potresnega inženirstva in inženirske seizmologije, oblikuje strokovne osnove predpisov in normativov za potresno varno gradnjo, pripravlja dokumentacijo za potresno varnost pomembnih objektov ter obdeluje in interpretira zapise pospeškov močnih potresov.

Slika 30: Organizacijska struktura Urada za seizmologijo.



Sama sem zaposlena v Sektorju za seizmologijo, katerega del nalog je tudi:

- obdelava in vrednotenje stalnih zapisov potresne dejavnosti doma in po svetu,
- skrb za obveščanje javnosti in pristojnih ustanov o potresih ter
- mednarodna izmenjava podatkov.

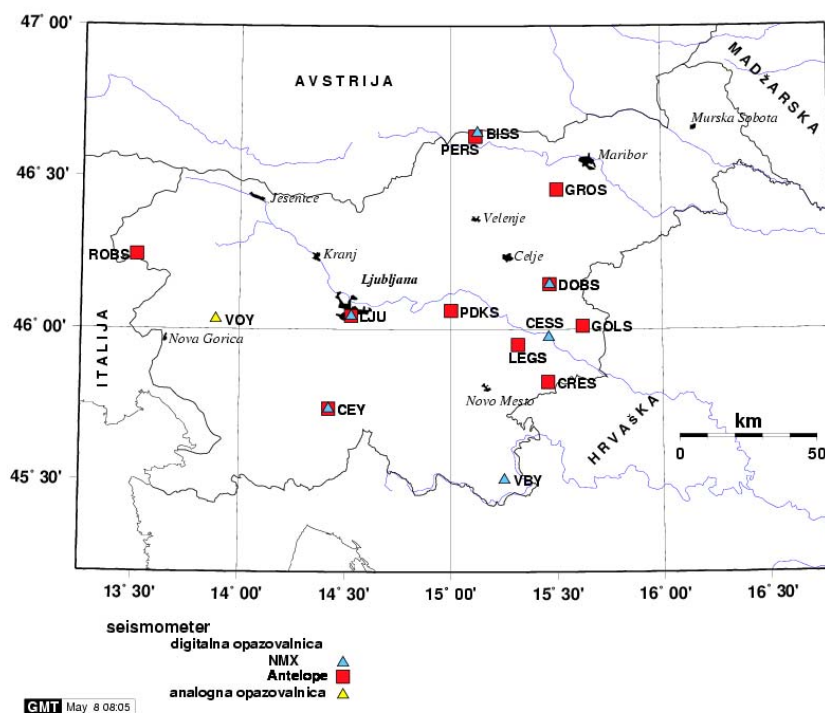
6.1.3. Opis problema

Velikonočni potres leta 1998 je razgalil kar precej pomanjkljivosti v organiziranosti slovenske seizmološke službe. Takratna seizmološka oprema in razporejenost potresnih opazovalnic na ozemlju Slovenije ni bila kos tako močnemu potresu, pa tudi zaposleni so imeli premalo izkušenj za ustrezno ukrepanje ob takšnem dogodku. Izkazalo se je tudi, da so se pričakovanja javnosti z leti spremenila. Odzivni čas ene ure po prejemu obvestila o potresu ne zadostuje več, še posebej boleče pa je bilo dejstvo, da je prišlo v javnost prvo obvestilo z druge strani meje, iz Italije. Izkazalo se je, da je bilo dolgoročno planiranje neustrezno ali pa ga sploh ni bilo. Kljub temu, da dejavnost urada ni usmerjena v ustvarjanje dobička in jo konkurenca ter okolje ne silita v nenehno prilagajanje, ne smemo zanemariti pomembnosti strateškega planiranja. Poudariti moram, da je seizmologija področje, kjer se nekateri še vedno trdno oklepajo tradicionalnih, starejših pristopov in očitno je bila žal tudi v Sloveniji potrebna »šok terapija«, da se je ugotovilo, da je skrajni čas, da se stopi korak naprej. Sprožil se je projekt posodobitve slovenske državne mreže potresnih opazovalnic, ki ga je odobrila tudi Vlada Republike Slovenije. Predvideva postavitve petindvajsetih opazovalnic, ki bodo enakomerno pokrivalo slovensko ozemlje in katerih podatki se bodo prenašali v center za zbiranje podatkov v Ljubljani v realnem času, se tam analizirali in arhivirali. Trenutno (leto 2003) je v državno mrežo potresnih opazovalnic vključenih 10 opazovalnic nove mreže (označeno s kvadratom na sliki 31) in 6 starejših opazovalnic (označeno s trikotnikom na sliki 31). Že

sedaj lahko koristimo nekatere pridobitve nove mreže. Naj naštejemo nekaj koristi, ki jih bo prinesel zaključek projekta, ko bo delovalo vseh petindvajset opazovalnic:

- Avtomatičen izračun lokacije, časa in magnitude potresa, ki se je zgodil v Sloveniji: To so trije podatki, ki zanimajo javnost v prvem trenutku. Pomembno je, da so kvalitetni in čim prej na voljo seizmologu, da pripravi poročilo za javnost in o potresu obvesti za to pristojne službe.

Slika 31: Državna mreža opazovalnic.



Vir: ARSO, urad za seizmologijo, 2003

- Pošiljanje sporočil o potresih seizmologom v obliki SMS sporočil in elektronske pošte: Pomembno je, da smo seznanjeni z dogodkih v Sloveniji in po svetu. Na podlagi prejetih informacije se lahko predvidijo potrebni ukrepi in akcije.
- Avtomatična kontrola delovanja potresnih opazovalnic in mrežnih povezav ter obveščanje o napakah v obliki SMS sporočil in elektronske pošte: Seizmologi so obveščeni o tem, kateri podatki so jim v danem trenutku na voljo, tehnologi, zadolženi za delovanje opreme, pa morajo ob pojavu napak, čim hitreje ukrepati.
- Izmenjava podatkov v realnem času s seizmološkimi institucijami drugih držav: Potresi ne poznajo meja. Lokacija potresa bo bolj natančna, če bo izračunana iz čim več kakovostnih podatkov. Zaželeno je tudi, da so opazovalnice enakomerno razporejene okoli epicentra potresa. Izmenjava podatkov pride do izraza še posebej tedaj, ko imamo opravka s potresi, katerih lokacija je blizu državnih mej. Trenutno

dobivamo podatke z nekaterih opazovalnic iz sosednjih držav, prizadevamo pa si to sodelovanje še povečati.

- Avtomatično naročanje podatkov s tujih opazovalnic z uporabo elektronskih sporočil in enostavna vključitev le-teh v proces analize.

Naštela sem le nekaj prednosti, ne smemo pa pozabiti tudi na možnosti vključevanja v razne domače in mednarodne projekte, na raziskave geološke zgradbe tal, notranje zgradbe Zemlje in podobno.

Vsaka dobra stvar pa lahko potegne za sabo tudi kakšno slabšo. V našem primeru je to kopičenje vedno večjega in večjega števila parametričnih podatkov o potresih, kot so na primer geografske koordinate lokacije, čas potresa, magnituda, intenziteta potresa in podobno. Problem hranjenja le-teh še vedno ni ustrezno rešen. Ne moremo reči, da se število potresov z leti povečuje, dejstvo pa je, da z boljšimi senzorji in gostejšo mrežo opazovalnic lahko zabeležimo večje število potresov, tudi zelo šibkih potresov. Povečevanje števila podatkov presega zmožnosti dosedanje obdelave. Če lahko rečemo, da bomo imeli na eni strani zelo sodobne rešitve, je drugo področje, hranjene podatkov, še vedno povsem na začetku. V preteklosti so se podatki o potresih hranili predvsem v papirnati obliki (katalog potresov), kasneje, odvisno od uporabljenih programov, so se vpisovali v računalnik in se arhivirali na različnih koncih, v različnih medijih (diskete, CD-ji, osebni računalniki,...) in v različnih oblikah. Pomembni podatki so se izgubili, ker niso bili zabeleženi ali pa nihče ne ve, kje se nahajajo, in zato postanejo izmerjene vrednosti neuporabne. Dejstvo, da so podatki hranjeni na različnih krajih in v različnih oblikah, pa tudi otežkoča nadaljnje analize in raziskave, saj je potrebno precej časa le za to, da se jih zbere in preoblikuje v takšno obliko, kot jo potrebujemo.

Da bi se izognili temu stanju, se je pojavila želja po razvoju seizmološkega informacijskega sistema, ki bi v vsakem trenutku omogočal pridobivanje kakovostnih podatkov o delovanju potresnih opazovalnic, lokaciji in času potresa, magnitudi in intenziteti potresa, kakovosti podatkov, omogočal bi hitrejšo izdelavo tedenskih in mesečnih biltenov, ki se pošiljajo v mednarodne seizmološke centre, ter lažjo pripravo mnenj, če je nek potres lahko povzročil določene poškodbe na objektih, saj bi bili vsi podatki zbrani na enem koncu in dosegljivi na enak način. Ker ne bi radi ponavljali napak iz preteklosti in bi se še pred začetkom projekta radi nanj dobro pripravili, se moramo zavedati tudi morebitnih tveganj, ki bi vodila v neuspeh.

- Slabo definiran izvedbeni načrt projekta,
- nerealna pričakovanja uporabnikov,
- nerazumevanje ciljev projekta,
- nepoznavanje delovnih postopkov,

- nenadzorovano naraščanje obsega projekta,
- nedefiniranost kontrolnih točk in
- spreminjanje strategije urada

so nekatera izmed tveganj, ki bi se jim radi izognili oziroma jih odpravili. V ta namen sem se lotila analize obstoječega poslovnega procesa obravnavanega problema in izdelave logičnega modela informacijskega sistema. Rezultati te analize in boljše razumevanje delovanja pa nam bodo služili kot podlaga za pripravo in začetek dejanskega projekta. Potrebno je bilo izbrati metodologijo, ki pokriva vse faze razvoja informacijskega sistema, poudarjena pa mora biti prav analiza procesov. Zaradi pomanjkanja izkušenj na področju načrtovanja informacijskih sistemov sem iskala metodologijo, ki:

- je enostavna za uporabo in dobro strukturirana,
- ima natančno definirane postopke, zaporedja postopkov in izhodne rezultate postopkov in
- ima pregledne in razumljive izhodne podatke.

V ta namen sem izbrala objektno orientirano metodologijo TAD (T. Damij, 2001).

6.1.4. Opredelitev osnovnih pojmov

Preden se lotimo analize, se moramo seznaniti tudi z nekaterimi osnovnimi pojmi s področja seizmologije, ki bodo kasneje nastopali v obravnavanem sistemu.

Digitalno opazovalnico sestavljajo digitalni trikomponentni seizmometer, analogno-digitalni pretvornik in osebni računalnik s sprejemnikom točnega časa GPS in programsko opremo, ki skrbi za zajem podatkov ter komunikacijo s centralnim računalnikom, ki je nameščen v središču za zbiranje podatkov v Ljubljani. Pri starejših opazovalnicah delujejo komunikacije na principu elektronske pošte, kontinuirane zapise z opazovalnic, ki sestavljajo novo državno mreže potresnih opazovalnic, pa imamo v Ljubljani v realnem času, kar omogoča takojšnjo obdelavo potresa. Zapise analiziramo z, v ta namen izdelanimi, računalniškimi programi.

Analogno opazovalnico sestavljajo en vertikalni in dva horizontalna seizmometra s seizmografskimi ojačevalniki ter pisač, ki omogoča zapisovanje na papir. Ti se menjavajo enkrat dnevno in se nato analizirajo.

Magnituda (velikostna stopnja) potresa je mera za sproščeno potresno energijo. Pojem magnitude je v seizmologijo vpeljal ameriški seizmolog C. F. Richter leta 1935. Magnituda je količina, ki je fizikalno določena. Seizmolog jo izračuna iz podatkov, ki jih dajo instrumenti, ki so potres zabeležili. Zaradi matematične (logaritemske) narave izračuna predstavlja vsaka višja stopnja potresa 30-krat več sproščene energije od nižje stopnje. Vse seizmološke

opazovalnice, ki so potres zabeležile, ga načeloma ovrednotijo z enako magnitudo, ne glede na epicentralno¹¹¹ oddaljenost. Malenkostna odstopanja nastanejo zaradi različne geološke zgradbe, na kateri stoji seizmološka opazovalnica.

Intenziteta potresa je mera učinkov potresa na ljudi, živali, zgradbe, predmete in naravo. Intenziteta potresa je vedno celo število, zapišemo pa jo z rimskimi številkami. Vsak potres je ovrednoten z več vrednostmi ali stopnjami potresnih učinkov. Največji učinki potresa so v nadžariščnem območju, nato učinki pojenjajo z oddaljenostjo od nadžarišča ali epicentra. Iz seizmološke zgodovine je znanih več kot petdeset intenzitetnih lestvic. Z razvojem gradbeništva so se tudi te razvijale in se prilagajale načinu gradnje. Danes so v svetu, predvsem pa v Evropi, najbolj uveljavljene tri lestvice:

1. Evropska potresna lestvica (European Macroseismic Scale 1998, EMS-98),
2. Mercalli-Cancani-Siebergova lestvica (MCS) in
3. Medvedov-Sponheuer-Karnikova lestvica (MSK).

V Sloveniji uporabljamo evropsko potresno lestvico, ki ima dvanajst stopenj.

Čas potresa je čas, ko je potres nastal. V seizmologiji uporabljamo univerzalni svetovni čas (v nadaljevanju UTC). Od našega lokalnega srednjeevropskega časa se razlikuje v zimskem času za eno, v poletnem pa za dve uri.

Lokacija potresa je izražena s koordinatami nadžarišča. Določimo jo iz časov prihodov vzdolžnega (P) in prečnega (S) valovanja na potresno opazovalnico ob upoštevanju hitrostnih modelov ter modelov za prostorsko in površinsko valovanje.

Na sliki 32 je primer izračunane lokacije potresa, pomen podatkov je predstavljen v tabeli 6.

Slika 32: Primer izračunane lokacije potresa.

2003	2	2	1107	7.7	L	46.067	14.735	7.0	LJU	18	.4	1.6LLJU	2.1CLJU		
ACTION:SPL 03-02-07 11:14 OP:ic STATUS:														ID:20030202110711	I

Tabela 6: Pomen podatkov o lokaciji potresa predstavljeni na sliki 32

Vrednost	Pomen	Vrednost	Pomen
2003 2 2	datum potresa	18	število faz upoštevanih pri izračunu
1107	ura in minuta potresa	.4	residium
7.7	sekunde	1.6LLJU	magnituda 1
L	tip potresa	2.1CLJU	magnituda 2
46.067	geografska širina	03-02-07 11:14	čas izračuna
14.735	geografska dolžina	ic	operater
7.0	globina potresa	20030202110711	identifikacija potresa
LJU	avtor		

¹¹¹ Nadžarišče ali epicenter potresa je točka na Zemljii površini, ki se nahaja nad žariščem ali hipocentrom potresa.

6.2. Definicija problema

Prva faza metodologije TAD je namenjena identifikaciji in opredelitvi obravnavanega problema. V ta namen organiziramo niz intervjujev z vsemi uporabniki sistema. V tej fazi se omejimo na predstavnike vodstva (T. Damij, 2001, str. 21).

Intervjuji najožjega vodstva podjetja so namenjeni identifikaciji strateškega plana in strateških ciljev podjetja, vitalnih izhodov in analiz, povezanih s strateškim planom in cilji, problemov pri podpori odločanju in strukturne organizacije podjetja. Med srečanji z vodilnimi predstavniki na poslovni in operativni ravni odločanja se seznanimo še s poslovnimi in operativnimi cilji, celotno strukturo posameznega oddelka ali organizacijske enote (entitete), vitalnimi izhodi in analizami, povezanimi s poslovnimi in operativnimi cilji, ter problemi pri podpori odločanja za potrebe posameznih entitet.

Strateški plan Urada za seizmologijo vključuje posodobitev slovenske državne mreže potresnih opazovalnic. Projekt, ki se je pričel leta 1998, predvideva postavitev 25 potresnih opazovalnic (do zaključka leta 2005), ki bodo enakomerno pokrivale slovensko ozemlje in katerih podatki se bodo prenašali v center za zbiranje podatkov v Ljubljani v realnem času, se tam analizirali in arhivirali. V naslednjih petih letih naj bi zgradili še dve lokalni mreži, prvo okoli Ljubljane in drugo okoli Krškega. Za zagotavljanje nemotenega prenosa podatkov v zbirni center v realnem času potrebujemo informacije o delovanju mreže, pri odločanju o najprimernejši opreми za posamezno opazovalnico pa nam bodo koristile informacije o delovanju potresnih opazovalnic. Glavna naloga urada je analiziranje potresne aktivnosti prelomov, ki potekajo po slovenskem ozemlju. Za to potrebujemo informacije o analiziranih potresih in o lokacijah potresov. Podatke, ki se nanašajo na potrese, ki so jih prebivalci Slovenije čutili, moramo posredovati tudi slovenski javnosti in za to pristojnim službam. Za izvrševanje te naloge potrebujemo informacije o intenziteti potresov, o nastali škodi in o izdanih poročilih o potresu. Seveda pa potresi ne poznajo državnih meja, zato je vedno bolj pomembna mednarodna izmenjava podatkov. Da bi vedeli, kateri podatki so bili posredovani, potrebujemo informacije o izpisu potresov. Pri odločanju pa si pomagamo s predpisi in trenutno veljavnimi seizmološkimi modeli.

V tej fazi je potrebno določiti tudi entitete oz. uporabnike, ki sodelujejo s sistemom. Definirala sem naslednje entitete, ki nastopajo v procesu (v oklepajih so zapisane kratice entitet, ki se pojavljajo v tabelah entitet, aktivnosti in nalog):

1. Dežurni seizmolog: Uslužbenec v oddelku za seizmologijo, ki je zadolžen za 24-urno dežurstvo. Sprejema obvestila o potresih, naredi preliminarne analize in obvešča javnost o potresih.
2. Seizmolog: Uslužbenec v oddelku za seizmologijo, ki je zadolžen za dnevno analizo potresov, kontrolo delovanja potresnih opazovalnic in povezav ter analizo zapisov z začasnih opazovalnic.

3. M-seizmolog: Seizmolog, ki skrbi za določitev intenzitete potresov in analizo njegovih učinkov na ljudi, zgradbe in naravo.
4. B-seizmolog: Seizmolog, ki skrbi za izdelavo končnih biltenov.
5. Vodja oddelka za seizmologijo (VOZS): Vodja oddelka skrbi za nemoteno delo celotnega oddelka.
6. Vodja Urada za seizmologijo (VUZS)
7. Direktor: Direktor Agencije Republike Slovenije za Okolje
8. VOY: Seizmološka postaja na Vojskem, s katere dežurni seizmolog naroči podatke.
9. Opazovalec: Zunanji (prostovoljni) sodelavci, ki z izpolnitvijo vprašalnikov pripomorejo k ocenitvi intenzitete potresa.
10. Center za Obveščanje Republike Slovenije (CORS): Dežurna služba, od katere dežurni seizmolog prejme obvestilo in kamor pošlje poročilo o potresu.
11. Svetovni seizmološki centri (Sv. centri): Mednarodni seizmološki centri, kjer se zbirajo podatki v svetovnih bazah podatkov.
12. Mediji: Slovenski mediji (televizija, radio, časopise), kamor dežurni seizmolog pošlje obvestilo o potresu.

Od dvanajstih entitet je prvih sedem notranjih, torej so del obravnavanega sistema, ostale pa so zunanje, saj so s sistemom le v interakciji. Notranje entitete predstavimo v tabeli entitet, ki je organizirana tako, da stolpci predstavljajo notranje entitete, vrstice pa analize in poročila, ki smo jih definirali na osnovi intervjujev.

Tabela 7: Tabela entitet

		Index	1	2	3	4	5	6	7
Req.	Index	Entity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VSZS	VUZS	Direktor
Report	1	Analizirani potresi	*	*	*	*	*		
	2	Informacije o izpish			*	*	*		
	3	Informacije o delovanju opazovalnic	*	*			*	*	
	4	Informacije o delovanju mreže	*	*			*		
	5	Informacije o intenziteti potresov	*	*	*		*	*	*
	6	Informacije o izdanih poročilih o potresu	*		*		*		
	7	Informacije o lokaciji potresov	*	*			*	*	*
	8	Informacije o nastali škodi	*	*			*	*	*
DS	9	Predpisi						*	*
	10	Seizmološki modeli					*	*	

6.3. Analiza delovanja sistema

V drugi fazi, fazi analize poslovnega procesa, podrobno analiziramo poslovni proces. Poskušamo ga natančno določiti in razumeti njegovo funkcioniranje. Ta faza ima tri korake.

Določimo vse aktivnosti, ki nastopajo v njem (prvi korak), in jih prikažemo v *tabeli aktivnosti* (glej tabelo 8 na str. 65). Tabela aktivnosti je organizirana dvodimenzionalno. V stolpcih so entitete, v vrsticah aktivnosti posameznih entitet. Aktivnosti povežemo vertikalno in horizontalno ter s tem ponazorimo potek procesa.

Horizontalno povežemo aktivnost z vsemi entitetami, ki sodelujejo pri njenem izvajanju. S črko S_j označimo entiteto (izvorno entiteto), ki sproži izvajanje izbrane aktivnosti v i -ti vrstici. Indeks j predstavlja zaporedno številko stolpca izvorne entitete v tabeli aktivnosti. Entiteto, ki zaključuje aktivnost (ciljno entiteto), označimo s črko T_j .

Vertikalno povežemo aktivnosti, ki jih opravlja ista entiteta glede na časovno zaporedje izvajanja. Povezave označimo s črkama P_i in U_i . P_i predstavlja predhodno aktivnost aktivnosti U_i določene entitete, indeks i pa je zaporedna številka vrstice v tabeli aktivnosti, v kateri se nahaja predhodna aktivnost.

Sočasno s tabelo aktivnosti kreiramo še *tabelo nalog* (glej tabelo 9 na str. 66). Vsaka aktivnost lahko sestoji iz ene ali več nalog, ki jih podrobneje opišemo v tabeli nalog (drugi korak). Tabela nalog je organizirana tako, da ima v vsaki vrstici posamezno nalogo predstavljeno s kodo (K_{ij}) in imenom, kjer je K naloga definirana v okviru določene aktivnosti, i in j pa sta indeksa aktivnosti in entitete tabele aktivnost, ki označujeta nalogo, v stolpce pa vpisujemo njene značilnosti: kratek opis naloge, čas potreben, da entiteta (j) opravi nalogo, pogoje, ki morajo biti izpolnjeni za izvedbo naloge, ter vhodne in izhodne elemente potrebne za izvedbo naloge oziroma so ti rezultat opravljene naloge.

V tretjem koraku druge faze posamezne aktivnosti združimo v delovne procese, te pa naprej v poslovni proces (tretji korak) (glej tabelo 10 na str. 67). *Delovni proces* je zbirka ene ali več aktivnosti, ki si sledijo v določenem vrstnem redu, zbirka enega ali več delovnih procesov pa je *poslovni proces*.

Faza analize delovanja sistema je ključnega pomena za obravnavani problem. Potrebno je bilo določiti naloge in aktivnosti, jih ustrezno povezati in nato združiti v delovne in poslovne procese. V mojem primeru sem določila 35 aktivnosti, v sklopu katerih sem definirala 58 nalog. Aktivnosti sem povezala v tri delovne procese: *Preliminarna analiza potresa in obveščanje o potresu*, *Dnevna analiza potresov in priprava tedenskega biltena* ter *Priprava mesečnega biltena*.

Definirani delovni procesi tvorijo en poslovni proces, *Opazovanje potresov* (glej tabelo 10 str. 67).

V nadaljevanju je prikazan del tabele aktivnosti in del tabele nalog obravnavanega sistema. V celoti sta tabeli prikazani v prilogi dela.

Tabela 8: Tabela aktivnosti - obstoječe stanje (1/2)

	Entity Index	Internal						External					
		1	2	3	4	5	6	7	8	9	10	11	12
Index	Activity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VSZS	VUZS	Direktor	VOY	opaz	CORS	mediji	Sv. centri
1	Sprejem obvestila	T10 P1									S10		
2	Pregled zapisov	U1, P2											
3	Analiza digitalnih seizmogramov	S1, T1 U2, P3											
4	Naroči VOY	S1 U3, P4						T1					
5	Dodaj VOY	T8 U4, P5						S8					
6	Lociranje potresa	S1, T1 U5, P6											
7	Potrdi analizo	S1, T1 U6, P7											
8	Kreiranje izpisa	S1, T1 U7, P8											
9	Pošiljanje izpisa	S1 U8, P9	T1	T1 P9		T1							T1
10	Kreiranje obvestila	S1, T1 U9, P10											
11	Pošiljanje obvestila	S1 U10	T1	T1 P11	T1	T1	T1	T1			T1	T1	
12	Kreiranje vprašalnikov			S3 U9, U11		T3 P12							
13	Potrditev vprašalnikov			T5, S3		S5 U12			T3				
14	Dnevna analiza		S2, T2 P14										
15	Lociranje potresov		S2, T2 U14, P15										
16	Potrditev analize	T2 P16	S2 U15										
17	Končni dnevni izpis	S1 U16	T1, S2, T2			T2	T2						

Tabela 9: Tabela nalog - obstoječe stanje (1/3)

Activity	Task Code	Description	Time	Condition	Input/Output
1 - Sprejem obvestila	K(1.1)	Dežurni seizmolog prejme s CORS-a obvestilo, da so prebivalci Slovenije čutili potres.			Obvestilo
2 - Pregled zapisov	K(2.1)	Dežurni seizmolog pregleda digitalne seizmograme.		Zapisi so?	
3 - Analiza digitalnih seizmogramov	K(3.1)	Dežurni seizmolog naredi analizo digitalnih seizmogramov. Za vsako opazovalnico določi fazo, vstopne čase, smer vstopa, ...		Potres je?	Analiza
4 - Naroči VDY	K(4.1)	Če oceni, da je potrebno, naroči podatke z analogne opazovalnice na Vojškem.		V DY deluje?	Analiza
5 - Dodaj VDY	K(5.1)	Ko prejme podatke z VDY, jih doda analizi.		Potres je zabeležen na VDY?	Analiza
6 - Lociranje potresa	K(6.1)	Dežurni seizmolog na podlagi analize izračuna lokacijo in magnitudo potresa.			Lokacija
7 - Potrdi analizo	K(7.1)	Dežurni seizmolog preveri, če je izračunana lokacija ustrezna. Če je potrebno, naredi popravke.		Lokacija dobra?	Lokacija, analiza
8 - Kreiranje izpisa	K(8.1)	Dežurni seizmolog kreira izpis potresa in vpiše rezultate izračuna lokacije.			Lokacija, izpis potresa
9 - Pošiljanje izpisa	K(9.1)	Dežurni seizmolog potrdi izpis in ga pošlje seizmologu, M-seizmologu, VSZS in svetovnim seizmološkim centrom.		Lokacija potresa?	Izpis potresa
	K(9.2)	Seizmolog prejme izpis potresa.			Izpis potresa
	K(9.3)	M-Seizmolog prejme izpis potresa. Ta mu je v pomoč pri kreiranju vprašalnikov o učinkih potresa.			Izpis potresa
	K(9.5)	VSZS prejme izpis potresa.			Izpis potresa
10 - Kreiranje obvestila	K(10.1)	Dežurni seizmolog kreira poročilo o potresu.			Poročilo o potresu
11 - Pošiljanje obvestila	K(11.1)	Dežurni seizmolog poročilu doda podatek, kje so potres čutili, in oceno intenzitete ter ga posreduje CORS-u, medijem, VSZS, VUZS, direktoju, seizmologu in M-seizmologu.		Lokacija potresa? Informacija o intenziteti?	Obvestilo, poročilo o potresu
	K(11.2)	Seizmolog prejme poročilo o potresu.			Poročilo o potresu
	K(11.3)	M-seizmolog prejme poročilo o potresu. Na podlagi le-tega kreira vprašalnike o učinkih potresa.			Poročilo o potresu
	K(11.5)	VSZS prejme poročilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(11.6)	VUZS prejme obvestilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(11.7)	Direktor prejme obvestilo o potresu in ustrezno ukrepa.			Poročilo o potresu

Tabela 10: Tabela prikazuje povezanost aktivnosti z delovnimi in poslovnimi procesi

Poslovni proces	Delovni proces	Aktivnost
Opazovanje potresov	Preliminarna analiza potresa in obveščanje o potresu	Sprejem obvestila
		Pregled zapisov
		Analiza digitalnih seizmogramov
		Naroči VOY
		Dodaj VOY
		Lociranje potresa
		Potrdi analizo
		Kreiranje izpisa
		Pošiljanje izpisa
		Kreiranje obvestila
		Pošiljanje obvestila
		Kreiranje vprašalnikov
		Potrditev vprašalnikov
	Dnevna analiza potresov in priprava tedenskega biltena	Dnevna analiza
		Lociranje potresov
		Potrditev analize
		Končni dnevni izpis
		Analiza začasnih
		Analiza LJU (anal.)
		VOY
		ZAMG
		Ponoven izračun lokacij
		Izdelava tedenskega biltena
		Potrditev tedenskega izpisa
	Priprava mesečnega biltena	Analiza vprašalnikov
		Določitev intenzitete
		Določitev Imax
		Bilten NEIC
		Preverjanje odstopanja
		Podatki tujih seiz. centrov
		Ponoven izračun lokacije
		Preverjanje lokacije
		Mesečni izpis
		Potrditev mesečnega izpisa
		Odpošiljanje mesečnega izpisa

6.4. Prenova poslovnih procesov

V fazi prenove poslovnih procesov s pomočjo podatkov, ki jih imamo zbrane v tabelah entitet, aktivnosti in nalog ponovno preučimo obstoječe poslovne procese. Namen faze je identifikacija in implementacija sprememb (npr. znižanje stroškov, krajši čas poslovnega cikla, poenostavitev operacij,...), ki bi vodile k boljšemu delovanju sistema. Prenova poslovnih procesov poteka v dveh korakih:

- analiza in prenova tabele entitet, aktivnosti in nalog,
- vsebino tabele aktivnosti prevedemo v procesni model sistema.

V ta namen ustanovimo projektno skupino, ki zaradi pomembnosti naloge vključuje tudi predstavnike ožjega vodstva podjetja ter predstavnike vodilnih na poslovni in operativni ravni. Ponovno se analizirajo cilji in plani podjetja na vseh ravneh in z različnih zornih kotov ter predlagajo spremembe. Na podlagi tega se spremeni tudi vsebina tabel. Predlagajo in implementirajo se spremembe in popravki, odstranijo se nepotrebne aktivnosti in naloge, če je potrebno se zaprejo nepotrebne oz. odprejo nove entitete (oddelki, enote), aktivnosti in naloge se zaradi boljše in lažje izvedbe prenesejo z ene na drugo entiteto.

Če ocenimo, da prenova ni potrebna, preskočimo prvi korak in izdelamo procesni model sistema na podlagi obstoječe tabele aktivnosti. Tudi izdelava procesnega modela ali diagrama tokov podatkov je izbirna. Zanj se odločimo takrat, ko izboljša preglednost sistema. V nasprotnem primeru preidemo na naslednjo fazo metodologije TAD.

V obravnavanem problemu sem se lotila definiranja nalog, aktivnosti, delovnih in poslovnih procesov vse od začetka in sem v tabele entitet, aktivnosti in nalog že na začetku vključila nekatere aktivnosti, za katere sem ocenila, da je njihovo beleženje nujno potrebno. Vseeno pa sem pri ponovnem pregledu tabel ugotovila, da bi bile potrebne še nekatere izboljšave.

Ugotovila sem, da ni vključene kontrole delovanja potresnih opazovalnic in mrežnih povezav. Deloma se ta izvaja že sedaj, le da v primeru odkritih nepravilnosti te sporočamo naprej odgovornim osebam ustno in nikjer tega ne beležimo. Nimamo zabeleženega ne časa, ne vzroka, ne trajanja napake, niti na kakšen način je bila ta odpravljena. Dejansko so to za nas zelo dragoceni podatki, saj morda čez teden dni še vemo, kaj se je tedaj dogajalo, čez mesec ali leto pa nikakor ne. V ta namen sem vključila v delovni proces »Dnevna analiza potresov in priprava tedenskega biltena« dve dodatni aktivnosti (»Kontrola delovanja opazovalnic« in »Kontrola delovanja mreže«) in s temi povezana dokumenta »Obvestilo o delovanju opazovalnic« in »Obvestilo o delovanju mreže«, ki bi pokrila del prej omenjenih informacij. Kontrolo delovanja mreže in opazovalnic sem vključila tudi v aktivnost »Preliminarna analiza potresa in obveščanje o potresu«, saj ne moremo izvršiti analize, če nam zapisi niso na voljo. Redko se sicer zgodi, da bi naenkrat odpovedale vse povezave in vse opazovalnice,

organizirana imamo namreč dva neodvisna sistema, a zagotoviti je potrebno čim bolj optimalno delovanje.

Posledično, z uvedbo novih aktivnosti, sem dodala v sistem dve novi notranji entiteti:

1. Služba za potresno opazovanje (SZPO): Služba urada, ki skrbi za nemoteno delovanje potresnih opazovalnic.
2. Služba za informatiko (SZI): Služba urada, ki skrbi za delovanje mrežnih povezav.

Načrtovanje informacijskega sistema je potekalo sočasno s procesom modernizacije mreže potresnih opazovalnic, ki še ni zaključen, zato sem morala določene stvari sproti prilagajati. Ko sem izdelala tabele entitet, aktivnosti in nalog ter jih še enkrat podrobneje pregledala, sem ugotovila, da bo zaradi procesa modernizacije nekaj aktivnosti odvečnih. To se nanaša predvsem na aktivnosti, povezane s potresno opazovalnico na Vojskem, ki bo predvidoma še letos zamenjana z novo digitalno opazovalnico, katere podatki se bodo v ljubljanskem središču zbirali v realnem času. Tako ne bo potrebno več naročati podatkov, niti jih kasneje dodajati pri izdelavi tedenskega biltena, saj nam bodo na voljo takoj. Pridobitev bo tudi, da bomo imeli boljšo pokritost zahodnega dela Slovenije, kjer je sedaj (glej sliko 31 na str. 58) ta slaba in kjer si pomagamo s podatki, ki jih dobimo od italijanskih kolegov.

Ravno tako trenutno poteka projekt, katerega cilj je, da bi tudi vse začasne opazovalnice analizirali na enak način kot ostale in sproti, torej, da bi imeli že pri dnevni analizi zbrane vse razpoložljive podatke s vseh slovenskih opazovalnic.

Zgoraj omenjena dejstva sem upoštevala pri prenovi. Izdelala sem nove tabele, v katerih nastopa ena entiteta manj (VOY), dodala pa sem dve novi (SZPO in SZI). Določila sem 35 aktivnosti, v sklopu katerih sem definirala 60 nalog. Nato pa sem na podlagi prenovljenih tabel izdelala še procesni model.

Tabela 11: Prenovljena tabela entitet

		Index	1	2	3	4	5	6	7	8	9
Req.	Index	Entity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VSZS	VUZS	SZPO	SZI	Direktor
Report	1	Analizirani potresi	*	*	*	*	*				
	2	Informacije o izpish			*	*	*				
	3	Informacije o delovanju opazovalnic	*	*			*	*	*		
	4	Informacije o delovanju mreže	*	*			*			*	
	5	Informacije o intenziteti potresov	*	*	*		*	*			*
	6	Informacije o izdanih poročilih o potresu	*		*		*				
	7	Informacije o lokaciji potresov	*	*			*	*			*
	8	Informacije o nastali škodi	*	*			*	*			*
DS	9	Predpisi						*			*
	10	Seizmološki modeli					*	*			

V nadaljevanju so prikazani deli tabele aktivnosti, tabele nalog in procesnega modela obravnavanega sistema po prenovi poslovnih procesov. V celoti sta tabeli in procesni model prikazani v prilogi dela.

Tabela 12: Prenovljena tabela aktivnosti (1/2)

Entity			Internal										External			
	Index	1	2	3	4	5	6	7	8	9	10	11	12	13		
Index	Activity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VŠZS	VUZS	SZPO	SZI	Direktor	opaz	CORS	mediji	Sv. centri		
1	Sprejem obvestila	T11 P1										S11				
2	Pregled zapisov	S1 U1, P2						T1	T1							
3	Analiza digitalnih seizmogramov	S1, T1 U2, P3														
4	Lociranje potresa	S1, T1 U3, P4														
5	Potrdi analizo	S1, T1 U4, P5														
6	Kreiranje izpisa	S1, T1 U5, P6														
7	Pošiljanje izpisa	S1 U6, P7	T1	T1 P7		T1								T1		
8	Kreiranje obvestila	S1, T1 U7, P8														
9	Pošiljanje obvestila	S1 U8	T1	T1 P9		T1	T1			T1		T1	T1			
10	Kreiranje vprašalnikov			S3 U7, U9		T3 P10										
11	Potrditev vprašalnikov			T5, S3		S5 U10					T3					
12	Kontrola delovanja opazovalnic		S2 P12			T2		T2								
13	Kontrola delovanja mreže		S2 U12, P13			T2			T2							
14	Dnevna analiza		S2, T2 U13, P14													
15	Lociranje potresov		S2, T2 U14, P15													
16	Potrditev analize	T2 P16	S2 U15													
17	Končni dnevni izpis	S1 U16	T1, S2, T2			T2	T2									

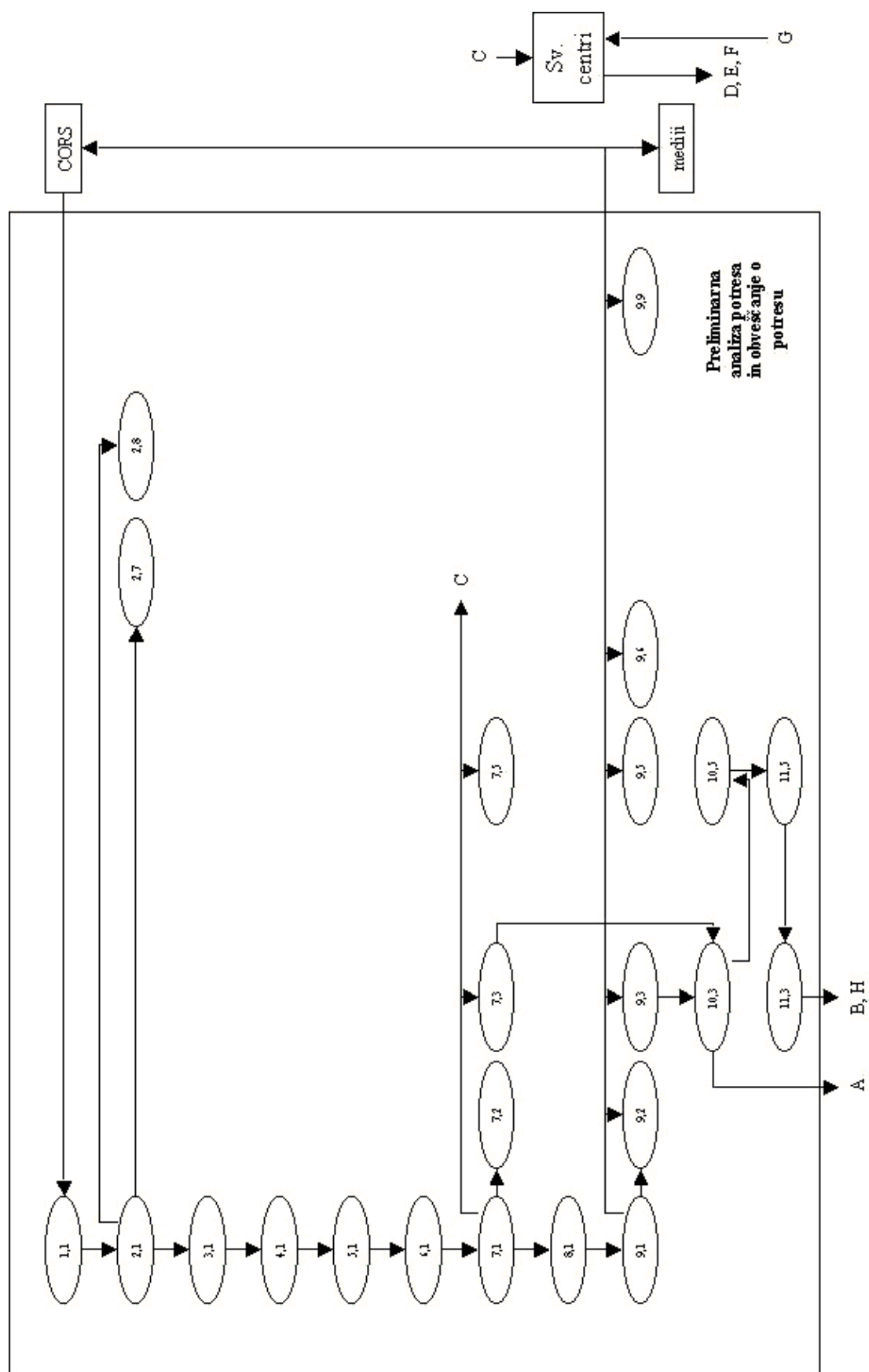
Tabela 13: Prenovljena tabela delovnih in poslovnih procesov

Poslovni proces	Delovni proces	Aktivnost
Opazovanje potresov	Preliminarna analiza potresa in obveščanje o potresu	Sprejem obvestila
		Pregled zapisov
		Analiza digitalnih seizmogramov
		Lociranje potresa
		Potrdi analizo
		Kreiranje izpisa
		Pošiljanje izpisa
		Kreiranje obvestila
		Pošiljanje obvestila
		Kreiranje vprašalnikov
		Potrditev vprašalnikov
	Dnevna analiza potresov in priprava tedenskega biltena	Kontrola delovanja opazovalnic
		Kontrola delovanja mreže
		Dnevna analiza
		Lociranje potresov
		Potrditev analize
		Končni dnevni izpis
		ZAMG
		Ponoven izračun lokacij
		Izdelava tedenskega biltena
		Potrditev tedenskega izpisa
	Priprava mesečnega biltena	Analiza vprašalnikov
		Določitev intenzitete
		Določitev I _{max}
		Bilten NEIC
		Preverjanje odstopanja
		Podatki tujih seiz. centrov
		Ponoven izračun lokacije
		Preverjanje lokacije
		Mesečni izpis
		Potrditev mesečnega izpisa
		Odpošiljanje mesečnega izpisa

Tabela 14: Prenovljena tabela nalog (1/3)

Activity	Task Code	Description	Time	Condition	Input/Output
1 - Sprejem obvestila	K(1.1)	Dežurni seizmolog prejme s CORS-a obvestilo, da so prebivalci Slovenije čutili potres.			Obvestilo
2 - Pregled zapisov	K(2.1)	Dežurni seizmolog pregleda digitalne seizmograme. Preveri, če delujejo povezave. V nasprotnem primeru obvesti SZI. Preveri, če delujejo opazovalnice. V primeru napak obvesti SZPO.		Zapisi so?	Obvestilo o delovanju opazovalnic, obvestilo o delovanju mreže
	K(2.7)	SZPO prejme obvestilo o napakah na opazovalnicah in ustrezno ukrepa.		Tip napake?	Obvestilo o delovanju opazovalnic
	K(2.8)	SZI prejme obvestilo o delovanju mreže in ustrezno ukrepa.		Tip napake?	Obvestilo o delovanju mreže
3 - Analiza digitalnih seizmogramov	K(3.1)	Dežurni seizmolog naredi analizo digitalnih seizmogramov. Za vsako opazovalnico določi fazo, vstopne čase, smer vstopa, ...		Potres je?	Analiza
4 - Lociranje potresa	K(4.1)	Dežurni seizmolog na podlagi analize izračuna lokacijo in magnitudo potresa.			Lokacija
5 - Potrdi analizo	K(5.1)	Dežurni seizmolog preveri, če je izračunana lokacija ustrezna. Če je potrebno, naredi popravke.		Lokacija dobra?	Lokacija, analiza
6 - Kreiranje izpisa	K(6.1)	Dežurni seizmolog kreira izpis potresa in vpiše rezultate izračuna lokacije.			Lokacija, izpis potresa
7 - Pošillanje izpisa	K(7.1)	Dežurni seizmolog potrdi izpis in ga pošlje seizmologu, M-seizmologu, VSZS in svetovnim seizmološkim centrom.		Lokacija potresa?	Izpis potresa
	K(7.2)	Seizmolog prejme izpis potresa.			Izpis potresa
	K(7.3)	M-Seizmolog prejme izpis potresa. Ta mu je v pomoč pri kreiranju vprašalnikov o učinkih potresa.			Izpis potresa
	K(7.5)	VSZS prejme izpis potresa.			Izpis potresa
8 - Kreiranje obvestila	K(8.1)	Dežurni seizmolog kreira poročilo o potresu.			Poročilo o potresu
	K(9.1)	Dežurni seizmolog poročilo doda podatek, kje so potres čutili, in oceno intenzitete ter ga posreduje CORS-u, medijem, VSZS, VUZS, direktoju, seizmologu in M-seizmologu.		Lokacija potresa? Informacija o intenziteti?	Obvestilo, poročilo o potresu
9 - Pošillanje obvestila	K(9.2)	Seizmolog prejme poročilo o potresu.			Poročilo o potresu
	K(9.3)	M-seizmolog prejme poročilo o potresu. Na podlagi tega kreira vprašalnik o učinkih potresa.			Poročilo o potresu
	K(9.5)	VSZS prejme poročilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(9.6)	VUZS prejme obvestilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(9.9)	Direktor prejme obvestilo o potresu in ustrezno ukrepa.			Poročilo o potresu

Slika 33: Procesni model (1/3)



6.5. Objektni model sistema

Čeprav je bilo moje delo osredotočeno predvsem na analizo delovanja sistema, sem v sledečih fazah želela izdelati tudi logično strukturo sistema in opredeliti potrebne operacije. V četrti fazi se bomo lotili izdelave objektnega modela sistema. Četrta faza je sestavljena iz šestih korakov, v katerih bomo analizirali dokumente, ki nastopajo v tabeli nalog. Izdelali smo jo v predhodnih fazah metodologije. Določili bomo funkcionalne odvisnosti atributov in na koncu izdelali semantično mrežo obravnavanega sistema.

Potek faze bom predstavila z dokumentom *Vprašalnik*. Na sliki 34 je primer vprašalnika oziroma podatki, ki so povezani s kreiranjem vprašalnika. Prvi del sestavljajo podatki o potresu, preliminarna ocena intenzitete, tip metode in način izbiranja opazovalcev, nato pa sledijo še podatki o opazovalcih, ki jim bo vprašalnik poslan (Zaradi varovanja osebnih podatkov, so originalna imena in priimki zbrisani.).

Slika 34: Primer podatkov povezanih z dokumentom *Vprašalnik*

Dogodek#: 20020411L6									
Datum: 11 4 2002									
Lokacija#: 04-1756-121.s200211									
Čas: 17h 56m UTC									
Epicenter: 46.147N 15.002E									
Magnituda: 1.1									
Vprašalnik#: trb20020411/1									
Minimalna ocena: 9									
Gostota mreže: 4									
Območje izbrano kot krog:									
koordinati središča: 0.155N 16.340E									
Notranji polmer: 0km									
Zunanji polmer: 27km									
Opazovalec# ... Kraj# ...									
1	46 Priimek1	Ime1	1433	46.068	15.185	5102.6	5514.3	9	5
2	52 Priimek2	Ime2	1222	46.186	14.874	5115.7	5490.3	9	5
3	99 Priimek3	Ime3	1412	46.140	14.966	5110.6	5497.4	9	5
4	101 Priimek4	Ime4	1223	46.173	14.796	5114.3	5484.2	9	5
5	121 Priimek5	Ime5	1282	46.083	14.898	5104.3	5492.1	8	5
6	125 Priimek6	Ime6	1225	46.173	14.681	5114.3	5475.4	8	5
7	266 Priimek7	Ime7	3314	46.271	15.044	5125.2	5503.4	9	5
8	281 Priimek8	Ime8	3270	46.146	15.243	5111.3	5518.8	9	5
9	582 Priimek9	Ime9	8297	46.013	15.174	5096.5	5513.5	8	5
.									
.									
.									

V **prvem koraku** četrte faze bomo z analizo tega dokumenta opredelili funkcionalne odvisnosti med identifikacijskimi atributi in ostalimi atributi, prisotnimi na dokumentu:

Slika 35: Funkcionalne odvisnosti med atributi dokumenta *Vprašalnik*

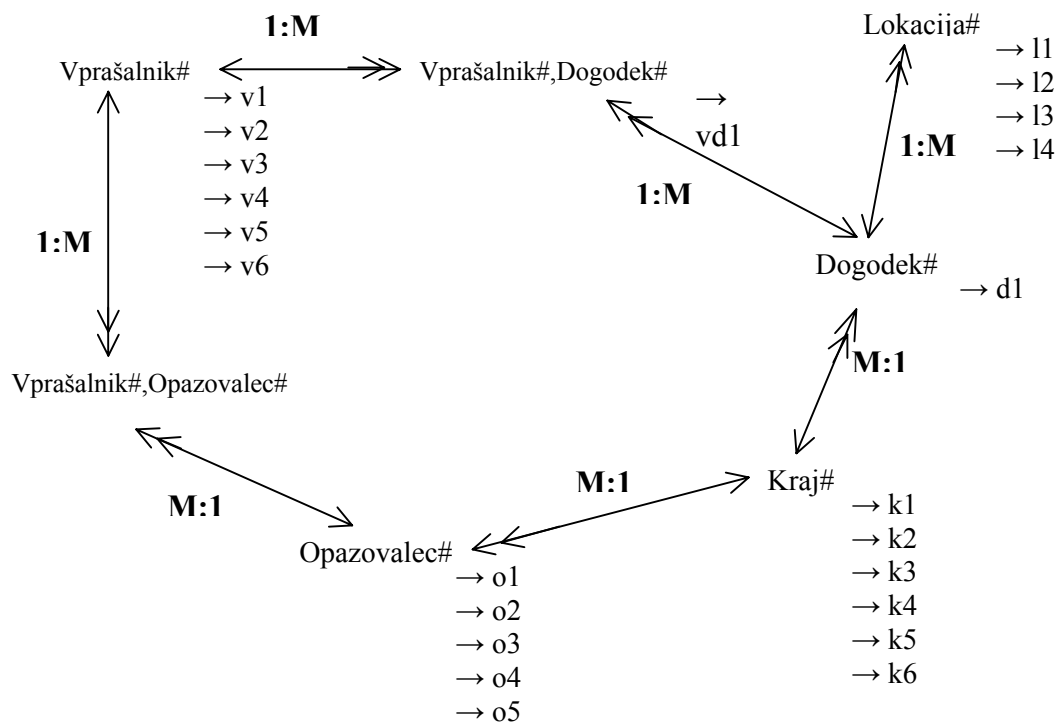
dogodek#	→ datum potresa	d1
lokacija#	→ čas potresa	l1
	→ geografska širina epicentra	l2
	→ geografska dolžina epicentra	l3
	→ magnituda potresa	l4
vprašalnik#	→ minimalna ocena opazovalca	v1
	→ gostota mreže izbora	v2
	→ tip izbora	v3
	→ geografske koordinate središča	v4
	→ zunanji polmer	v5
	→ notranji polmer	v6
vprašalnik#, dogodek#	→ število vprašalnikov	vd1
opazovalec#	→ Ime	o1
	→ Priimek	o2
	→ Naslov	o3
	→ ocena opazovalca	o4
	→ gostota mreže	o5
kraj#	→ ime kraja	k1
	→ poštna številka	k2
	→ geografska širina kraja	k3
	→ geografska dolžina kraja	k4
	→ Gauss-Krügerjeve projekcija koordinat kraja (X)	k5
	→ Gauss-Krügerjeve projekcija koordinat kraja (Y)	k6

Za lažji prikaz rezultatov drugega, tretjega in četrtega koraka sem vsak atribut označila še s kratico.

Nato sem v **drugem koraku** opredelila povezave, ki obstajajo med posameznimi identifikacijskimi atributi. Opredeliti moramo tudi tip povezave (1:1, 1:m, n:m). Povezave

tipa n:m sem nadomestila z dvema povezavama tipa 1:m. V omenjenem primeru imamo dve takšni povezavi: Vprašalnik – Dogodek in Vprašalnik – Opazovalec, zato vpeljemo dva nova razreda, ki ju identificira kombinacija Vprašalnik#, Dogodek# v prvem primeru in Vprašalnik#, Opazovalec# v drugem.

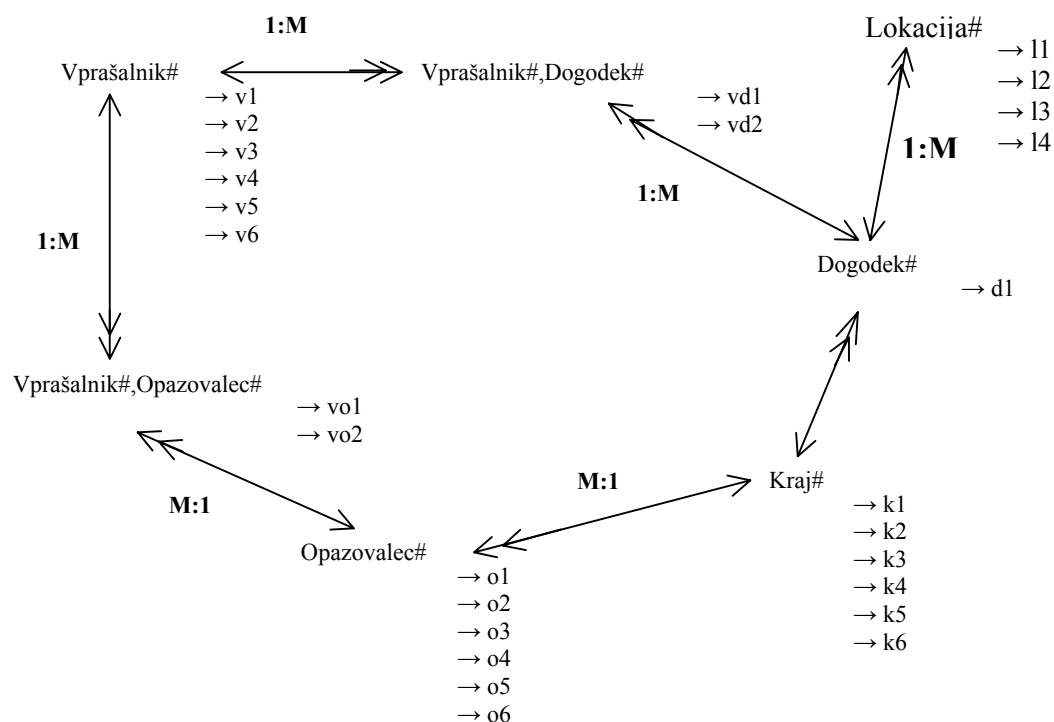
Slika 36: Asociacije med identifikacijskimi atributi



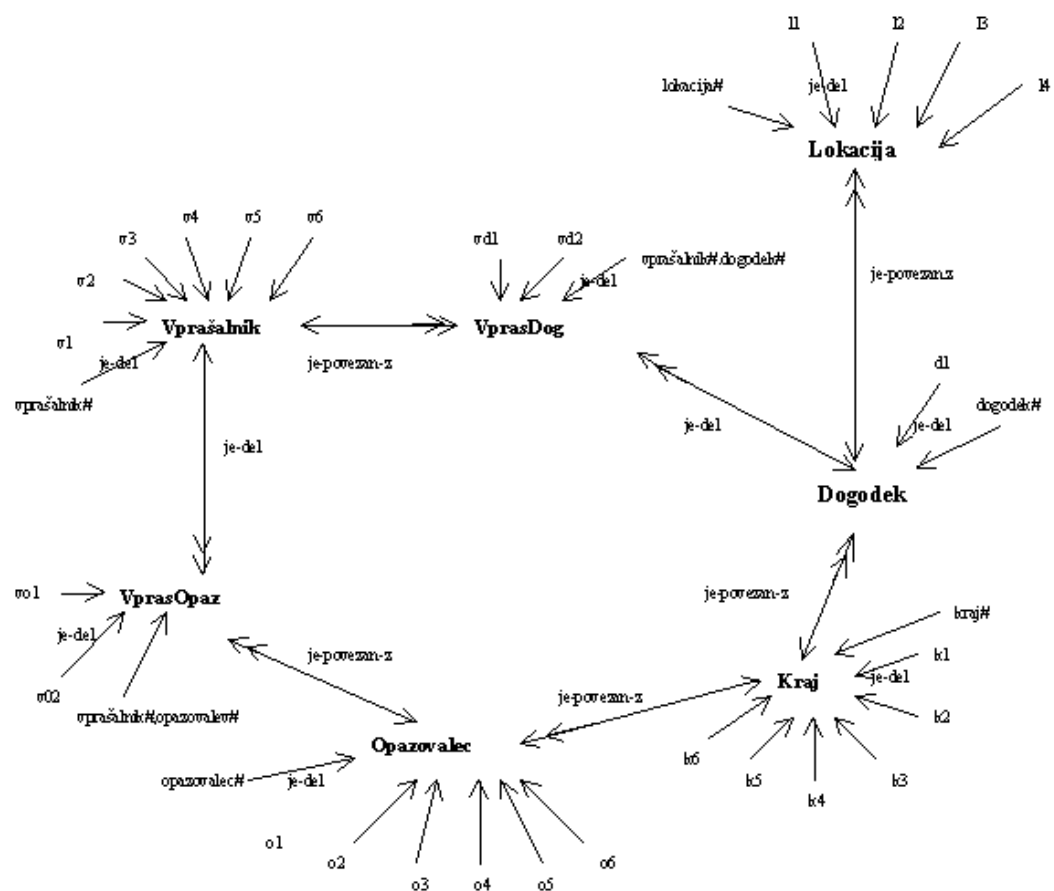
Po končani analizi vseh dokumentov na opisan način te analize v **tretjem koraku** integriramo. Združimo funkcionalne odvisnosti vseh analiziranih dokumentov, ki so povezani z istimi identifikacijskimi atributi. V primeru, ki ga obravnavamo, dodamo attribute, ki niso bili vsebovani v dokumentu *Vprašalnik*. Dodani so atributi, ki se nanašajo na intenziteto potresa, saj se ta oceni na podlagi vrnjenih vprašalnikov. Dodamo attribute vrnjen vprašalnik (vo1), ocenitev vprašalnika (vo2), datum kreiranja vprašalnika (vd2) in telefonska številka opazovalca (o6) (glej sliko 37 na naslednji strani).

Sledi **četrti korak**. Zamenjamo identifikacijske attribute z imeni razredov. Povežemo vse attribute, tudi identifikacijske, z razredi. Relacija je tipa *Je-del* (glej str. 29). Povezave med razredom *Vprašalnik* in *Opazovalec*, ki je tipa m:n, povežemo preko asociacijskega razreda *VprasOpaz* z dvema povezavama tipa 1:m. podobno storimo pri povezavi razredov *Vprašalnik* in *Dogodek*, ki ju povežemo preko asociacijskega razreda *VprasDog*. Povezave med razredi so tipa *Je-povezan-z* ali *Je-del*. Rezultat četrtega koraka omenjenega primera je prikazan na sliki 38 na naslednji strani.

Slika 37: Integracija analize



Slika 38: Razredi in njihove medsebojne povezave



V **petem koraku** spremenimo rezultate prvih štirih korakov v začetni objektni model obravnavanega sistema. V našem primeru smo analizo opravljeno v prvih štirih korakih, prikazali le na manjšem delu objektnega modela. Celotni začetni objektni model sistema je prikazan na sliki 39 na strani 79.

V zadnjem, **šestem koraku**, dopolnimo začetni objektni model. Upoštevamo vse informacije, ki jih imamo zbrane v tabelah entitet, aktivnosti in nalog ter tudi znanje strokovnjakov. Po potrebi dodamo dodatne razrede (Dodala sem razred »Kalibracija«, v katerem beležimo podatkov o kalibraciji merilnikov). Poiščemo razrede, ki imajo skupne lastnosti, in te izločimo v nadrazrede. Če je potrebno, pa kreiramo specializirane razrede. V obravnavanem primeru sem kreirala dva nadrazreda. Prvi, »Bilten«, je povezan z izdelavo dnevnih, tedenskih in mesečnih biltenov, drugi, »Obvest«, pa s pošiljanjem obvestil o delovanju opazovalnic in mreže (glej sliko 40 na str. 80)

Vsi razredi v sistemu imajo osnovne operacije, ki omogočajo kreiranje novega objekta, brisanje, spreminjanje in pregledovanje objekta. Poleg teh imajo nekateri tudi operacije, ki smo jih definirali z analizo procesa in so zapisani v tabeli nalog.

Podrobnejši opis razredov je naveden v prilogi.

6.6. Načrtovanje sistema

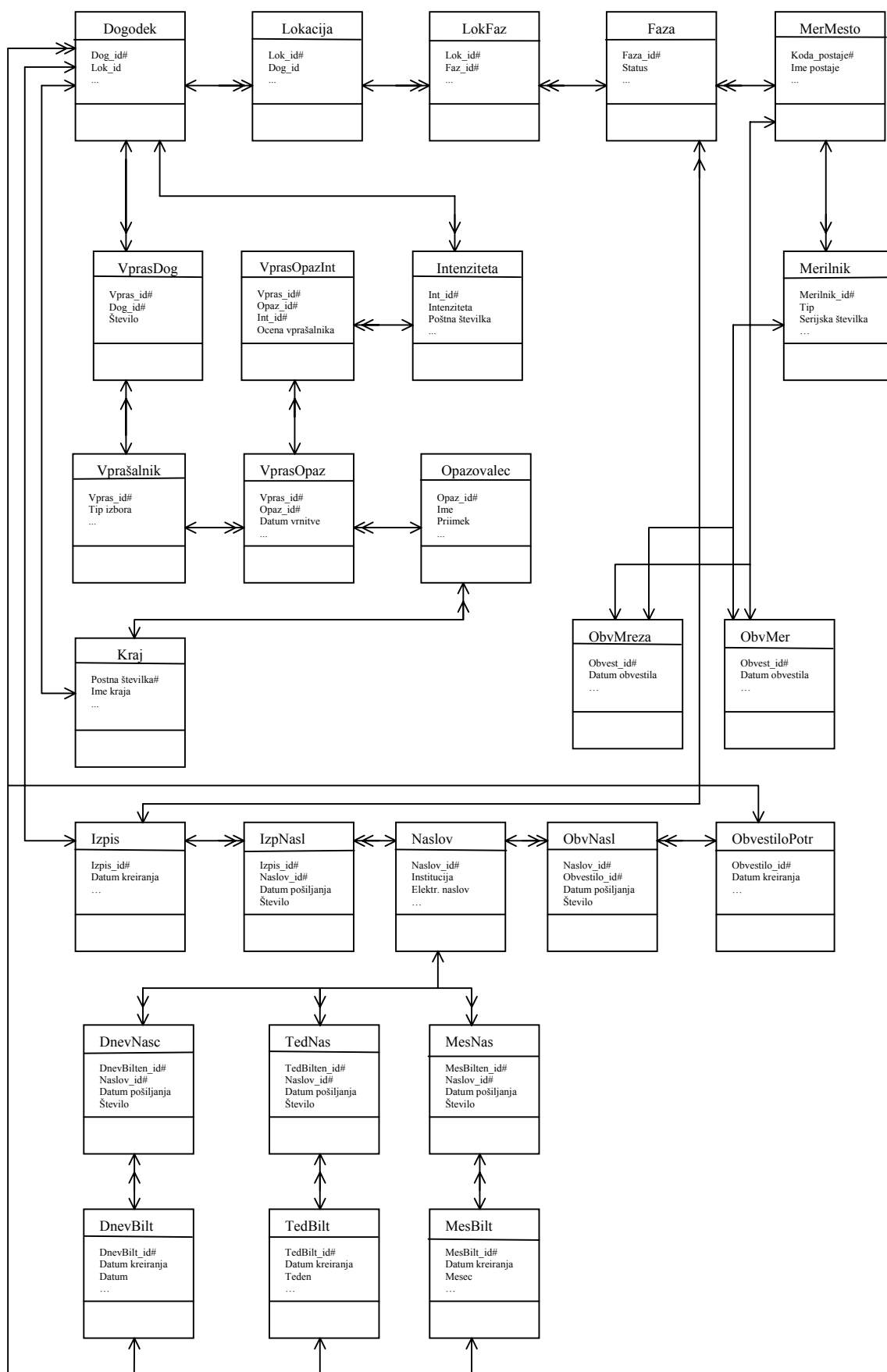
V peti fazi metodologije TAD *definiramo operacije* in izdelamo *aplikacijski model*.

Operacije določimo s pomočjo tabele nalog. Analiziramo vsako nalogo posebej in zapišemo v *tabelo operacij* ime operacije, kodo naloge ter razred, ki mu pripada. Vsaki operaciji pripada ena vrstica v tabeli operacij, ker pa je lahko vsaka naloga povezana z več operacijami, pripada vsaki nalogi temu številu ustrezno število vrstic v isti tabeli. Ko opredelimo vse operacije, jih dodamo še v končni objektni model in jim definiramo algoritme.

Nato sestavimo še aplikacijski model iz podatkov, ki smo jih definirali v tabelah entitet, aktivnosti in nalog. Aplikacijski model temelji na podatkih zbranih v tabelah entitet in aktivnosti. Prva dva dela sta sestavljena iz poročil in odločitvenih problemov, ki smo jih zbrali v tabeli entitet. Tretji del razberemo iz prvega stolpca tabele aktivnosti. Tako dobljeni aplikacijski model nam poda pregled zahtev, odločitvenih problemov in shemo procesov ter aktivnosti (glej tabelo 16 na str. 82).

V obravnavanem sistemu sem identificirala 40 operacij, ki so prikazane v tabeli operacij (glej tabelo 15 na str. 81). Poleg teh imajo vsi razredi tudi običajne operacije za kreiranje, spreminjanje, pregledovanje in brisanje objektov. Pri delu sem se omejila na določitev operacij in izdelavo aplikacijskega modela, nisem pa se lotila definiranja algoritmov operacij in uporabniških vmesnikov za vsak del aplikacijskega modela.

Slika 39: Začetni objektni model



Slika 40: Končni objektni model

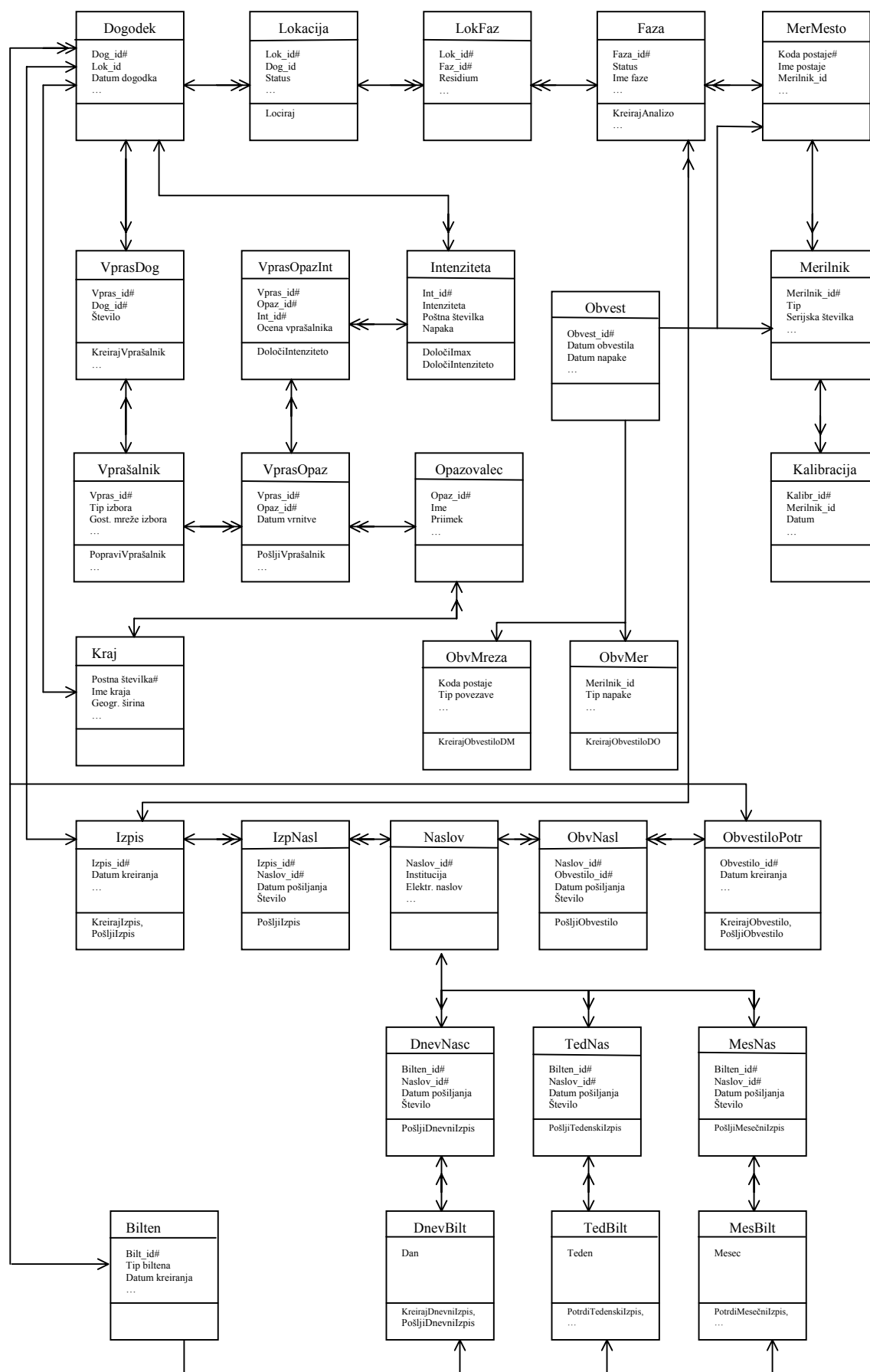


Tabela 15: Tabela operacij

Koda naloge	Operacija	Razred
K _{2,1}	KreirajObvestiloDO	ObvestMer
K _{2,1}	KreirajObvestiloDM	ObvestMreza
K _{3,1}	KreirajAnalizo	Faza
K _{4,1}	Lociraj	Lokacija
K _{5,1}	PotrdiAnalizo	Faza
K _{5,1}	PopraviAnalizo	Faza
K _{6,1}	KreirajIzpis	Izpis
K _{7,1}	PošljiIzpis	Izpis
K _{8,1}	KreirajObvestilo	ObvestPotr
K _{9,1}	PošljiObvestilo	ObvestPotr, ObvNasl
K _{10,3}	KreirajVprašalnik	Vprašalnik, VprasDog
K _{11,5}	PotrdiVprašalnik	Vprašalnik, VprasDog
K _{11,3}	PopraviVprašalnik	Vprašalnik, VprasDog
K _{12,3}	PošljiVprašalnik	Vprašalnik, VprasOpaz
K _{12,2}	KreirajObvestiloDO	ObvestMer
K _{13,2}	KreirajObvestiloDM	ObvestMreza
K _{14,2}	KreirajAnalizo	Faza
K _{15,2}	Lociraj	Lokacija
K _{16,1}	PotrdiAnalizo	Faza
K _{16,2}	PopraviAnalizo	Faza
K _{17,2}	KreirajDnevniIzpis	DnevBilt
K _{17,2}	PošljiDnevniIzpis	DnevBilt, DnevNas
K _{18,1}	DodajPodatke	Faza
K _{19,1}	Lokacija	Lokacija
K _{20,1}	KreirajTedenskiIzpis	TedBilt
K _{21,2}	PotrdiTedenskiIzpis	TedBilt,
K _{21,1}	PošljiTedenskiIzpis	TedBilt, TedNas
K _{22,3}	AnalizaVprašalnika	Vprasalnik, VprasOpaz
K _{23,3}	DoločiIntenziteto	Intenziteta, VprasOpazInt
K _{24,3}	DoločiImax	Intenziteta
K _{25,4}	PrimerjajAnalize	Faza
K _{26,4}	PreveriOdstopanja	Faza
K _{27,4}	DodajPodatke	Faza
K _{28,4}	Lokacija	Lokacija
K _{29,4}	PreveriLokacijo	Lokacija
K _{30,4}	KreirajMesečniIzpis	MesBilt
K _{31,3}	PotrdiMesečniIzpis	MesBilt
K _{31,5}	PotrdiMesečniIzpis	MesBilt
K _{31,4}	PopraviMesečniIzpis	MesBilt
K _{32,4}	PošljiMesečniIzpis	MesBilt, MesNas

Tabela 16: Aplikacijski model

Seizmološki informacijski sistem	Poročila	Analizirani potresi	
		Informacije o izpisih	
		Informacije o delovanju opazovalnic	
		Informacije o delovanju mreže	
		Informacije o intenziteti potresov	
		Informacije o izdanih poročilih o potresu	
		Informacije o lokaciji potresov	
		Informacije o nastali škodi	
	Podpora odločanju	Predpisi	
		Seizmološki modeli	
	Opazovanje potresov	Preliminarna analiza potresa in obveščanje o potresu	Sprejem obvestila
			Pregled zapisov
			Analiza digitalnih seizmogramov
			Lociranje potresa
			Potrdi analizo
			Kreiranje izpisa
			Pošiljanje izpisa
			Kreiranje obvestila
			Pošiljanje obvestila
			Kreiranje vprašalnikov
			Potrditev vprašalnikov
		Dnevna analiza potresov in priprava tedenskega biltena	Kontrola delovanja opazovalnic
			Kontrola delovanja mreže
			Dnevna analiza
			Lociranje potresov
			Potrditev analize
			Končni dnevni izpis
			ZAMG
			Ponoven izračun lokacij
			Izdelava tedenskega biltena
			Potrditev tedenskega izpisa
		Priprava mesečnega biltena	Analiza vprašalnikov
			Določitev intenzitete
			Določitev I _{max}
			Bilten NEIC
			Preverjanje odstopanja
			Podatki tujih seiz. centrov
			Ponoven izračun lokacije
			Preverjanje lokacije
			Mesečni izpis
			Potrditev mesečnega izpisa
			Odpošiljanje mesečnega izpisa

6.7. Implementacija

Zadnja faza metodologije je namenjena implementaciji sistema, ki smo ga razvili v predhodnih fazah. Izbere se ustrezno podatkovno bazo in programski jezik ter algoritme prevede v programsko kodo.

V okviru naloge implementacija ni bila izvršena. Nekaj razlogov sem navedla že v opisu problema. Dodam pa lahko še nekatere:

- Najprej bo potrebno zaključiti projekt posodobitve državne mreže potresnih opazovalnic.
- Še bolj natančno bo potrebno preučiti spremembe, ki jih bo projekt prinesel.
- Raziskati bo potrebno spreminjajoče se svetovne trende.
- Poenotiti bo potrebno formate izpisov in biltenov.
- Nenazadnje pa bo potrebno v delo vključiti tudi večje število ljudi.

7. Sklep

V delu je prikazana raziskava potreb seizmološkega informacijskega sistema. Za delo sem izbrala objektivno orientiran pristop. Predstavitvi njegovih osnovnih konceptov sledi predstavitev dveh metodologij, ki temeljita na tem pristopu. Za Poenoten proces z uporabo Poenotenega jezika za modeliranje UML je značilno, da je preslikava realnosti v model predstavljena z različnimi diagrami.

Sama sem v delu izbrala njej alternativno metodologijo, metodologijo TAD. Namesto diagramskih tehnik so pri slednji preslikave realnosti v model predstavljene z nizom tabel. Metodologija TAD z vsemi njenimi fazami in koraki je v nalogi podrobneje predstavljena. Zanj je značilen zelo sistematičen pristop h gradnji informacijskega sistema in dobro definirane faze ter koraki od začetka gradnje, definicije problema, do njenega konca, implementacije. To je bil eden izmed razlogov, da sem jo uporabila tudi sama, saj je primerna tudi za začetnike oziroma manj izkušene s področja načrtovanja informacijskih sistemov. Njena uporaba v praksi, v primerjavi z metodologijo UML, ne zahteva posebnega predznanja, le dobro poznavanje problemskega področja.

Metodologijo TAD sem uporabila na praktičnem primeru razvoja seizmološkega informacijskega sistema. Analizirala sem obstoječe stanje in določila cilje ter informacijske potrebe. Definirala sem 35 aktivnosti, ki nastopajo v procesu, od sprejema obvestila o potresu do izdelave in pošiljanja mesečnega biltena. Aktivnosti sem prikazala v tabeli aktivnosti. V sklopu teh aktivnosti sem definirala 58 nalog, ki so predstavljene v tabeli nalog. Aktivnosti sem združila v tri delovne procese (»Preliminarna analiza potresa in obveščanje o potresu«, »Dnevna analiza potresov in priprava tedenskega biltena« in »Priprava mesečnega biltena«), ki tvorijo en poslovni proces, »Opazovanje potresov«. Poleg pregleda obstoječega stanja sem v fazi prenove poslovnih procesov predlagala in vključila nekaj sprememb, ki bi omogočale boljše delovanje sistema in pripomogle k večji preglednosti nad dogajanjem. V delovni proces »Dnevna analiza potresov in priprava tedenskega biltena« sem vključila kontrolo delovanja potresnih opazovalnic in kontrolo mrežnih povezav, saj je potrebno zagotoviti čim bolj optimalno delovanje sistema. V povezavi s tem sem v sistem vključila dve novi notranji entiteti (Služba za potresno opazovanje (SZPO) in Služba za informatiko (SZI)). Za nekaj aktivnost in entitet se je izkazalo, da bodo po zaključku modernizacije državne mreže potresnih opazovalnic postale nepotrebne in sem jih zato izpustila. Izdelala sem prenovljene tabele entitet, aktivnosti in nalog. Prenovljena tabela aktivnosti vsebuje 32 aktivnost, v sklopu katerih je definiranih 60 nalog. Izdelala sem tudi objektni model sistema, ki ga sestavlja 29 razredov. V fazi načrtovanja sem se omejila na opredelitev operacij, nisem pa se lotila definiranja algoritmov operacij. Opredelila sem 40 operacij in jih dodala v končni objektni model sistema. V fazi načrtovanja sem izdelala tudi aplikacijski model. Implementacije se nisem lotila, saj je bil moj primarni cilj določitev poslovnih in delovnih procesov ter izdelava logičnega modela sistema. Drugi razlog je, da še vedno poteka proces modernizacije državne

mreže potresnih opazovalnic. To me je oviralo že pri izdelavi pričujoče naloge, saj sem morala določene stvari sproti prilagajati in spreminjati. Za dejansko realizacijo seizmološkega informacijskega sistema bodo prišla prav spoznanja, pridobljena pri izdelavi te naloge, vendar bo potrebno v delo vključiti večje število ljudi.

Uporaba metodologije TAD na obravnavanem problemu se je izkazala za neproblematično. Pri izdelavi tabel sem si pomagala z orodjem TAD, verzija 1.0, ki omogoča hitro in pregledno kreiranje tabel entitet, aktivnosti in nalog ter aplikacijskega modela, ne omogoča pa kreiranja objektnega modela. Osebnost me to niti ni motilo, saj sem želela, kot manj izkušena s področja načrtovanja informacijskih sistemov, resnično dobro preučiti proces izgradnje objektnega modela, podrobno analizirati vse dokumente in razumeti vse korake, ki vodijo do izgradnje končnega objektnega modela. Vnaprej pripravljena orodja lahko zakrijejo kakšno pomembno podrobnost, ki jo je potrebno upoštevati, vsekakor pa so taka orodja dobrodošla tedaj, ko jih uporabljamo pogosto. V omenjenem orodju tudi ni podpore za fazo načrtovanja sistema (razen izdelave aplikacijskega modela) in fazo implementacije. Mogoče je razlog temu tudi to, da je izbor programskega jezika in baze podatkov prepuščen implementacijskim timom. Predvidevam, da bo dodatna programska podpora vključena v naslednje verzije orodja TAD.

Z nalogo sem potrdila hipotezo, da je možno z uporabo metodologije TAD popisati cilje Urada za seizmologijo, na podlagi teh ciljev definirati in pregledno popisati poslovne in delovne procese, ki so pomembni za seizmološki informacijski sistem, predlagati spremembe procesov ter pripraviti logični model sistema.

8. Literatura in viri

8.1. Literatura

1. Arlow Jim, Neustadt Ila: UML and the Unified Process: Practical Object - Oriented Analysis & Design. Boston : Addison-Wesley, 2002. 395 str.
2. Avison D. E., Fitzgerald G.: Information Systems Development: Methodologies, Techniques and Tools. 2nd ed. London : McGraw-Hill, 1996. 505 str.
3. Båth Marcus: Introduction to seismology. Basel : Birkhäuser, 1979. 428 str.
4. Blanc Miloš: Razvoj informacijskega sistema logistike z uporabo metodologije Tabularni razvoj aplikacij (TAD). Magistrsko delo. Ljubljana : Ekonomska fakulteta, 2002. 102 str.
5. Booch Grady: Object-oriented analysis and design with applications. Redwood city : The Benjamin/Cummings, 1994. 589 str.
6. Booch Grady: Object solutions: managing the object-oriented project. Menlo Park : Addison-Wesley, 1996. 323 str.
7. Booch Grady, Rumbaugh James, Jacobson Ivar: The Unified Modelling Language User Guide. Reading : Addison-Wesley, 1999. 482 str.
8. Burleson D. Keith: Practical Application of Object-Oriented Techniques to Relational Databases. New York : John Wiley & Sons, 1994. 250 str.
9. Carmichael Andy: Object Development Methods. New York : SIGS Books, 1994. 347 str.
10. Črv Milan: Objektni pristop k prenovi poslovnih procesov in izgradnji informacijskega sistema, metodološki vidik. Doktorska dizertacija. Ljubljana : Ekonomska fakulteta, 2000. 202 str.
11. Damij Nadja: Uporaba metodologije TAD in UML za razvoj aplikacije Tranzit. Ljubljana : Uporabna informatika, številka 2, 2001. str. 86-94.
12. Damij Talib: An Object-Oriented Methodology for Information System development and Business Process Reengineering. Journal of Object-Oriented Programming. Fort Lee, Vol. 13, No. 4, 2000, str. 23-24.
13. Damij Talib: Načrtovanje Informacijskih sistemov. Ljubljana : Ekonomska fakulteta, 1994. 43 str.
14. Damij Talib: Tabular Application Development for Information Systems. New York: Springer Verlag, 2001. 256 str.

15. Dolgoff Anatole: Physical Geology. Updated version. Boston : Houghton Mifflin co., 1998. 638 str.
16. Donaldson Dewitz Sandra: Systems Analysis and Design and Transition to Objects. New York : McGraw-Hill, 1996. 192 str.
17. Embley W. David, Kurtz D. Barry, Woolfield N. Scott: Object-Oriented Systems Analysis, A Model-Driven Approach. Englewood Cliffs : Prentice-Hall, 1992. 302 str.
18. Finkelstein Clive: Information Engineering, Strategic Systems Development. Sydney : Addison-Wesley, 1992. 660 str.
19. Gibson L. Michael, Huges T. Cary: Systems Analysis and Design. Denver : Boyd & Fraser Publishing Company, 1994. 796 str.
20. Grad Janez, Jaklič Jurij: Baze podatkov. Ljubljana : Ekonomska fakulteta, 1996. 260 str.
21. Gradišar Miro, Resinovič Gortan: Informatika v poslovnem okolju : Ljubljana. Ekonomska fakulteta, 1996. 479 str.
22. Graham Ian: Object Oriented Methods, Second Edition. Workingham : Addison-Wesley, 1994. 473 str.
23. Grünthal Gottfried: European Macroseismic Scale 1998 (EMS-98). Luxembourg : Conseil de l'Europe, Cahiers du Centre Européen de Géodynamique et de Séismologie, Vol. 15, 1998. 99 str.
24. Hammer Michael, Champy James: Reengineering The Corporation, A Manifesto for Business Revolution. New York : Harper Business, 1993. 223 str.
25. Harmon Paul, Taylor A. David: Object in Action, Commercial Applications of Object-Oriented Technologies. Reading : Addison-Wesley, 1993. 232 str.
26. Jacobson Ivar: Object-Oriented Software Engineering: A Use Case Driven Approach. Workingham : Addison-Wesley, 1996. 528 str.
27. Jacobson Ivar, Booch Grady, Rumbaugh James: The Unified Software Development Process. Reading : Addison-Wesley, 1999. 463 str.
28. Khoshafian Setrag, Abnous Razmik: Object Orientation: Concepts, Analysis & Design, Languages, Databases, Graphical user interfaces, Standards. New York : John Wiley & Sons, 1995. 504 str.
29. Kovačič Andrej, Vintar Mirko: Načrtovanje in gradnja informacijskih sistemov. Ljubljana : DZS, 1994. 316 str.

30. Kulhánek Ota: Anatomy of Seismograms. Amsterdam : Elsevier Science Publishers, 1990. 179 str.
31. Kušar Štrukelj Marinka: Objektni pristop k izgradnji dokumentacijskega informacijskega sistema. Magistrsko delo. Ljubljana : Ekonomska fakulteta, 2001. 120 str.
32. Olle T.W., Hagelstein J., Macdonald I.G., Rolland C., Sol H.G., Van Assche F.J.M., Verrijn-Stuart A.A.: Information Systems Methodologies: A framework for understanding. Wokingham : Addison-Wesley, 1991. 401 str.
33. Maclean Roy: Analysing systems, Determining Requirements for Object-Oriented development. New York : Prentice-Hall, 1994. 294 str.
34. Martin James: Principles of Object-Oriented Analysis and Design. Englewood Cliffs : Prentice-Hall, 1984. 300 str.
35. Migliore R.H., Stevens R. E., Loudon D. L., Williamson S.: Strategic Planning for Not-to-Profit Organizations. New York : The Haworth Press, 1995. 200 str.
36. Montgomery Stephen: Object-Oriented Information Engineering, Analysis, Design and Implementation. London : Academic Press, 1994. 353 str.
37. Rozman R., Kovač J., Koletnik F.: Management. Ljubljana : Gospodarski vestnik, 1993. 312 str.
38. Sanders G. Lawrence: Data Modeling. Danwers : Boyd & Fraser, 1995. 148 str.
39. Shearer M. Peter: Introduction to seismology. Cambridge : Cambridge University Press, 1999. 260 str.
40. Sturm Jake.: VB6 UML Design and Development. Bbirmingham : Wrox Press, 1999. 581 str.
41. Taylor A. David: Object-Oriented Technology: A Manager's Guide. Reading : Adison-Wesley, 1990. 147 str.
42. Tomažič R., Novaković S.: Strateško planiranje razvoja informacijskih sistemov kot orodje za racionalizacijo poslovanja. Dnevi slovenske informatike 2000, Portorož : Slovensko društvo informatika, 2000. str. 162-169.
43. Tomažič Spiller M.: Razvoj informacijskega sistema avtomatskih meteoroloških postaj z uporabo metode Tabularni razvoj aplikacij. Magistrsko delo. Ljubljana : Ekonomska fakulteta, 2001. 91 str.

44. Turban E., MacLean E., Wetherbe J.: Information technology for management: making connections for strategic advantage. New York : John Willey & Sons, 1999. 791 str.
45. Watson Gregory H.: Business Systems Engeneering; Managing breaktrought changes for produktivity and Profit. New York : John Willey & Sons, 1994, 287 str.

8.2. Viri

1. Dokumentacija SEISAN.
2. Pravilniku o notranji organizaciji in sistemizaciji mest ARSO.
3. Seizmološki bilten. Ljubljana : Agencija Republike Slovenije za okolje, urad za seizmologijo.
4. Ribarič Vladimir: Seizmičnost Slovenije – Katalog potresov. Ljubljana : Seizmološki zavod RS Slovenije, 1982. 649 str.
5. Ribarič Vladimir: Seizmičnost Slovenije – Katalog potresov, Dodatek 1. Ljubljana : Seizmološki zavod RS Slovenije, 1992. 145 str.
6. Ribarič Vladimir: Seizmičnost Slovenije – Katalog potresov, Dodatek 2. Ljubljana : Seizmološki zavod RS Slovenije, 1994. 94 str
7. USGS: United States Geological Survey. Large / Noteworthy Earthquakes in 2003.
URL: http://wwwneic.cr.usgs.gov/neis/bulletin/03_EVENTS/index.html, 2003.

Priloga

KAZALO PRILOGE

Tabela 1: Tabela entitet (obstoječe stanje)	1
Tabela 2: Tabela aktivnosti (obstoječe stanje)	2
Tabela 3: Delovni in poslovni procesi (obstoječe stanje)	4
Tabela 4: Tabela nalog (obstoječe stanje)	5
Tabela 5: Prenovljena tabela entitet	8
Tabela 6: Prenovljena tabela aktivnosti	9
Tabela 7: Prenovljena tabela delovnih in poslovnih procesov	11
Tabela 8: Prenovljena tabela nalog	12
Slika 1: Procesni model (1/3)	15
Slika 2: Procesni model (2/3)	16
Slika 3: Procesni model (3/3)	17
Slika 4: Diagram delovnih procesov	18
Slika 5: Podrobnejši opis razredov	19

Tabela 1: Tabela entitet (obstoječe stanje)

		Index	1	2	3	4	5	6	7
Req.	Index	Entity Analysis	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VSZS	VUZS	Direktor
Report	1	Analizirani potresi	*	*	*	*	*		
	2	Informacije o izpisih			*	*	*		
	3	Informacije o delovanju opazovalnic	*	*			*	*	
	4	Informacije o delovanju mreže	*	*			*		
	5	Informacije o intenziteti potresov	*	*	*		*	*	*
	6	Informacije o izdanih poročilih o potresu	*		*		*		
	7	Informacije o lokaciji potresov	*	*			*	*	*
	8	Informacije o nastali škodi	*	*			*	*	*
DS	9	Predpisi						*	*
	10	Seizmološki modeli					*	*	

Tabela 2: Tabela aktivnosti (obstoječe stanje)

	Entity Index	Internal							External				
		1	2	3	4	5	6	7	8	9	10	11	12
Index	Activity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VSZS	VUZS	Direktor	VOY	opaz	CORS	mediji	Sv. centri
1	Sprejem obvestila	T10 P1									S10		
2	Pregled zapisov	U1, P2											
3	Analiza digitalnih seizmogramov	S1, T1 U2, P3											
4	Naroči VOY	S1 U3, P4						T1					
5	Dodaj VOY	T8 U4, P5						S8					
6	Lociranje potresa	S1, T1 U5, P6											
7	Potrdi analizo	S1, T1 U6, P7											
8	Kreiranje izpisa	S1, T1 U7, P8											
9	Posiljanje izpisa	S1 U8, P9	T1	T1 P9		T1							T1
10	Kreiranje obvestila	S1, T1 U9, P10											
11	Posiljanje obvestila	S1 U10	T1	T1 P11		T1	T1	T1			T1	T1	
12	Kreiranje vprašalnikov			S3 U9, U11		T3 P12							
13	Potrditev vprašalnikov			T5, S3		S5 U12				T3			
14	Dnevna analiza		S2, T2 P14										
15	Lociranje potresov		S2, T2 U14, P15										
16	Potrditev analize	T2 P16	S2 U15										
17	Končni dnevni izpis	S1 U16	T1, S2, T2			T2	T2						

Entity		Internal						External					
	Index	1	2	3	4	5	6	7	8	9	10	11	12
Index	Activity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VŠZS	VUZS	Direktor	VOY	opaz	CORS	mediji	Sv. centri
18	Analiza začasnih opazovalnic		S2, T2										
19	Analiza LJU (anal.)	T2 P19	S2										
20	VOY	T8 U19, P20							S8				
21	ZAMG	T12 U20, P21											S12
22	Ponoven izračun lokacij	S1, T1 U21, P22											
23	Izdelava tedenskega izpisa	S1 U22	T1 P23										
24	Potrditev tedenskega izpisa	T2, S1	S2 U23		T1		T1						T1
25	Analiza vprašalnikov			T9 P25						S9			
26	Določitev intenzitete			S3, T3 U25, P26									
27	Določitev Imax			S3, T3 U26									
28	Bilten NEIC				T12 P28								S12
29	Preverjanje odstopanja				S4, T4 U28, P29								
30	Podatki tujih seiz. centrov				T12 U29, P30								S12
31	Ponoven izračun lokacije				S4, T4 U30, P31								
32	Preverjanje lokacije				S4, T4 U31								
33	Mesečni izpis			T4 P33	S4	T4 P33							
34	Potrditev mesečnega izpisa			S3 U33	T3, T5 P34	S5 U33							
35	Odpošiljanje mesečnega izpisa				S4 U34								T4

Tabela 3: Delovni in poslovni procesi (obstoječe stanje)

Poslovni proces	Delovni proces	Aktivnost
Opazovanje potresov	Preliminarna analiza potresa in obveščanje o potresu	Sprejem obvestila
		Pregled zapisov
		Analiza digitalnih seizmogramov
		Naroči VOY
		Dodaj VOY
		Lociranje potresa
		Potrdi analizo
		Kreiranje izpisa
		Pošiljanje izpisa
		Kreiranje obvestila
		Pošiljanje obvestila
		Kreiranje vprašalnikov
		Potrditev vprašalnikov
	Dnevna analiza potresov in priprava tedenskega biltena	Dnevna analiza
		Lociranje potresov
		Potrditev analize
		Končni dnevni izpis
		Analiza začasnih
		Analiza LJU (anal.)
		VOY
		ZAMG
		Ponoven izračun lokacij
		Izdelava tedenskega biltena
		Potrditev tedenskega izpisa
	Priprava mesečnega biltena	Analiza vprašalnikov
		Določitev intenzitete
		Določitev Imax
		Bilten NEIC
		Preverjanje odstopanja
		Podatki tujih seiz. centrov
		Ponoven izračun lokacije
		Preverjanje lokacije
		Mesečni izpis
		Potrditev mesečnega izpisa
		Odpošiljanje mesečnega izpisa

Tabela 4: Tabela nalog (obstoječe stanje)

Activity	Task Code	Description	Time	Condition	Input/Output
1 - Sprejem obvestila	K(1.1)	Dežurni seizmolog prejme s CORS-a obvestilo, da so prebivalci Slovenije čutili potres.			Obvestilo
2 - Pregled zapisov	K(2.1)	Dežurni seizmolog pregleda digitalne seizmograme.		Zapisi so?	
3 - Analiza digitalnih seizmogramov	K(3.1)	Dežurni seizmolog naredi analizo digitalnih seizmogramov. Za vsako opazovalnico določi fazo, vstopne čase, smer vstopa, ...		Potres je?	Analiza
4 - Naroči VOY	K(4.1)	Če oceni, da je potrebno, naroči podatke z analogne opazovalnice na Vojskem.		VOY deluje?	Analiza
5 - Dodaj VOY	K(5.1)	Ko prejme podatke z VOY, jih doda analizi.		Potres je zabeležen na VOY?	Analiza
6 - Lociranje potresa	K(6.1)	Dežurni seizmolog na podlagi analize izračuna lokacijo in magnitudo potresa.			Lokacija
7 - Potrdi analizo	K(7.1)	Dežurni seizmolog preveri, če je izračunana lokacija ustrezna. Če je potrebno, naredi popravke.		Lokacija dobra?	Lokacija, analiza
8 - Kreiranje izpisa	K(8.1)	Dežurni seizmolog kreira izpis potresa in vpiše rezultate izračuna lokacije.			Lokacija, izpis potresa
9 - Pošiljanje izpisa	K(9.1)	Dežurni seizmolog potrdi izpis in ga pošlje seizmologu, M-seizmologu, VSZS in svetovnim seizmološkim centrom.		Lokacija potresa?	Izpis potresa
	K(9.2)	Seizmolog prejme izpis potresa.			Izpis potresa
	K(9.3)	M-Seizmolog prejme izpis potresa. Ta mu je v pomoč pri kreiranju vprašalnikov o učinkih potresa.			Izpis potresa
	K(9.5)	VSZS prejme izpis potresa.			Izpis potresa
10 - Kreiranje obvestila	K(10.1)	Dežurni seizmolog kreira poročilo o potresu.			Poročilo o potresu
11 - Pošiljanje obvestila	K(11.1)	Dežurni seizmolog poročilu doda podatek, kje so potres čutili, in oceno intenzitete ter ga postreduje CORS-u, medijem, VSZS, VUZS, direktorju, seizmologu in M-seizmologu.		Lokacija potresa? Informacija o intenziteti?	Obvestilo, poročilo o potresu
	K(11.2)	Seizmolog prejme poročilo o potresu.			Poročilo o potresu
	K(11.3)	M-seizmolog prejme poročilo o potresu. Na podlagi le-tega kreira vprašalnik o učinkih potresa.			Poročilo o potresu
	K(11.5)	VSZS prejme poročilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(11.6)	VUZS prejme obvestilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(11.7)	Direktor prejme obvestilo o potresu in ustrezno ukrepa.			Poročilo o potresu

Activity	Task Code	Description	Time	Condition	Input/Output
12 - Kreiranje vprašalnikov	K(12.3)	M-seizmolog pripravi vprašalnice o učinkih potresa na ljudi, poslopja, naravo in jih postreduje VSZS v potrditev.		Lokacija, magnituda potresa, število klicev	Vprašalnik
	K(12.5)	VSZS prejme vprašalnik v potrditev.			Vprašalnik
	K(13.3)	M-seizmolog prejme od VSZS potrjen vprašalnik (če je potrebno ga spremeni) in ga pošlje opazovalcem.			Vprašalnik
13 - Potrditev vprašalnikov	K(13.5)	VSZS pregleda vprašalnik in ga potrdi ali zahteva popravke.		Območje, število	Vprašalnik
14 - Dnevna analiza	K(14.2)	Seizmolog naredi dnevno analizo digitalnih seizmogramov.		Zapisi so?	Analiza
15 - Lociranje potresov	K(15.2)	Seizmolog izračuna lokacije in magnitude potresov, ki jih je analiziral.			Lokacija
16 - Potrditev analize	K(16.1)	Dežurni seizmolog prejme izračune in jih preveri. Če je potrebno zahteva popravke.		Odstopanja	Analiza, lokacija
	K(16.2)	Seizmolog postreduje izračune dežurnemu seizmologu, da jih la preveri.			Analiza, lokacija
	K(17.1)	Dežurni seizmolog potrdi izračune.		Izračuni so vredni?	Analiza, lokacija
17 - Končni dnevni izpis	K(17.2)	Seizmolog pripravi končni dnevni izpis in ga pošlje ostalim seizmologom ter VSZS.			Dnevni izpis
	K(17.5)	VSZS prejme dnevni izpis.			Dnevni izpis
	K(17.6)	VUZS prejme dnevni izpis.			Dnevni izpis
18 - Analiza začasnih opazovalnic	K(18.2)	Seizmolog naredi analizo seizmogramov začasnih opazovalnic in jih doda dnevni analizi.		Začasne opazovalnice delujejo? Potres zabeležen?	Analiza
19 - Analiza LJU (anal.)	K(19.1)	Dežurni seizmolog prejme analizo LJU in jo doda dnevni analizi.			Analiza
	K(19.2)	Seizmolog naredi analizo analognih seizmogramov (lublanske opazovalnice in jih postreduje dežurnemu seizmologu.		LJU deluje? Potres je zabeležen?	Analiza
20 - VDY	K(20.1)	Dežurni seizmolog prejme analizo VDY in jo doda dnevni analizi.		VDY deluje? Potres je zabeležen?	Analiza
21 - ZAMG	K(21.1)	Dežurni seizmolog prejme avstrijske analize (ZAMG) in jih doda dnevni analizi.		ZAMG deluje?	Analiza
22 - Ponoven izračun lokacij	K(22.1)	Ko doda podatke z LJU, VDY in ZAMG, dežurni seizmolog ponovno izračuna lokacije potresov in njihovo magnitudo.			Analiza, lokacija
23 - Izdelava tedenskega izpisa	K(23.1)	Dežurni seizmolog kreira tedenski izpis potresov in ga postreduje seizmologu, da ga preveri.			Tedenski izpis
	K(23.2)	Seizmolog prejme tedenski izpis, ga preveri in predlaga popravke.		Izračuni dobri? Kaj manjka?	Tedenski izpis

Activity	Task Code	Description	Time	Condition	Input/Output
24 - Potrditev tedenskega izpisa	K(24.1)	Dežurni seizmolog ustrezno popravi tedenski izpis in ga nato pošlje VSZS, VUZS in svetovnim seizmološkim centrom.			Tedenski izpis
	K(24.2)	Seizmolog potrdi tedenski izpis		Tedenski izpis vredni?	Tedenski izpis
	K(24.5)	VSZS prejme tedenski izpis.			Tedenski izpis
	K(24.6)	VUZS prejme tedenski izpis.			Tedenski izpis
25 - Analiza vprašalnikov	K(25.3)	Ko zbere M-seizmolog vprašalnike, jih zanalizira.		Vprašalniki so vrnjeni? Vprašalniki so izpolnjeni?	Vprašalniki
26 - Določitev intenzitete	K(26.3)	S pomočjo vprašalnikov določi intenziteto potresa po krajih.		Število pravilno izpolnjenih vprašalnikov	Intenziteta
27 - Določitev I _{max}	K(27.3)	M-seizmolog določi maksimalno intenziteto potresa in jo vpiše k analizi.			Intenziteta, analiza
28 - Bilten NEIC	K(28.4)	B-seizmolog prejme bilten NEIC in ga primerja z slovenskimi rezultati.		Potresi vsebovani v biltenu NEIC?	Bilten NEIC, analiza, lokacija
29 - Preverjanje odstopanja	K(29.4)	B-seizmolog še enkrat preveri podatke, pri kateri opazi večja odstopanja in naredi ustrezne popravke.		Ujemanje lokacij, časa?	Bilten NEIC, analiza, lokacija
30 - Podatki tujih seiz. centrov	K(30.4)	B-seizmolog doda podatke, ki jih prejme od tujih seizmoloških centrov.		Oddaljenost opazovalnic, magnituda potresa	Analiza
31 - Ponoven izračun lokacije	K(31.4)	B-seizmolog ponovno preračuna lokacije potresov.			Analiza, lokacija
32 - Preverjanje lokacije	K(32.4)	B-seizmolog preveri lokacije potresov.		Odstopanja?	Analiza, lokacija
33 - Mesečni izpis	K(33.3)	M-seizmolog prejme mesečni izpis, da ga preveri. Ob ugotovljenih napakah ustrezno ukrepa.		Večja odstopanja, manjkajoči podatki	Mesečni izpis
	K(33.4)	B-seizmolog pripravi mesečni izpis in ga posreduje M-seizmologu in VSZS. Ob ugotovljenih napakah ustrezno ukrepa.		Večja odstopanja, manjkajoči podatki	Mesečni izpis
	K(33.5)	VSZS prejme mesečni izpis in ga pregleda.			Mesečni izpis
	K(34.3)	M-seizmolog potrdi mesečni izpis		Mesečni izpis vredni?	Mesečni izpis
34 - Potrditev mesečnega izpisa	K(34.4)	B-seizmolog naredi ustrezne popravke mesečnega izpisa.			Mesečni izpis
	K(34.5)	VSZS potrdi mesečni izpis.		Mesečni izpis vredni?	Mesečni izpis
35 - Odpošiljanje mesečnega izpisa	K(35.4)	B-seizmolog posreduje mesečni izpis potresov svetovnim seizmološkim centrom, kjer ga vnesejo v svoje baze podatkov.			Mesečni izpis

Tabela 5: Prenovljena tabela entitet

		Index	1	2	3	4	5	6	7	8	9
Req.	Index	Analysis Entity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VSZS	VUZS	SZPO	SZI	Direktor
Report	1	Analizirani potresi	*	*	*	*	*				
	2	Informacije o izpisih			*	*	*				
	3	Informacije o delovanju opazovalnic	*	*			*	*	*		
	4	Informacije o delovanju mreže	*	*			*			*	
	5	Informacije o intenziteti potresov	*	*	*		*	*			*
	6	Informacije o izdanih poročilih o potresu	*		*		*				
	7	Informacije o lokaciji potresov	*	*			*	*			*
	8	Informacije o nastali škodi	*	*			*	*			*
DS	9	Predpisi						*			*
	10	Seizmološki modeli					*	*			

Tabela 6: Prenovljena tabela aktivnosti

	Entity Index	Internal										External			
		1	2	3	4	5	6	7	8	9	10	11	12	13	
Index	Activity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VZSZ	VUZS	SZPO	SZI	Direktor	opaz	CORS	mediji	Sv. centri	
1	Sprejem obvestila	T11 P1										S11			
2	Pregled zapisov	S1 U1, P2						T1	T1						
3	Analiza digitalnih seizmogramov	S1, T1 U2, P3													
4	Lociranje potresa	S1, T1 U3, P4													
5	Potrdi analizo	S1, T1 U4, P5													
6	Kreiranje izpisa	S1, T1 U5, P6													
7	Posiljanje izpisa	S1 U6, P7	T1	T1 P7		T1								T1	
8	Kreiranje obvestila	S1, T1 U7, P8													
9	Posiljanje obvestila	S1 U8	T1	T1 P9		T1	T1			T1		T1	T1		
10	Kreiranje vprašalnikov			S3 U7, U9		T3 P10									
11	Potrditev vprašalnikov			T5, S3		S5 U10					T3				
12	Kontrola delovanja opazovalnic		S2 P12			T2		T2							
13	Kontrola delovanja mreže		S2 U12, P13			T2			T2						
14	Dnevna analiza		S2, T2 U13, P14												
15	Lociranje potresov		S2, T2 U14, P15												
16	Potrditev analize	T2 P16	S2 U15												
17	Končni dnevni izpis	S1 U16	T1, S2, T2			T2	T2								

Entity		Internal									External			
	Index	1	2	3	4	5	6	7	8	9	10	11	12	13
Index	Activity	Dežurni seizmolog	Seizmolog	M-Seizmolog	B-Seizmolog	VSZS	VUZS	SZP0	SZI	Direktor	opaz	CORS	mediji	Sv. centri
18	ZAMG	T13 P18												S13
19	Ponoven izračun lokacij	S1, T1 U18, P19												
20	Izdelava tedenskega izpisa	S1 U19	T1 P20											
21	Potrditev tedenskega izpisa	T2, S1	S2 U20			T1	T1							T1
22	Analiza vprašalnikov			T10 P22							S10			
23	Določitev intenzitete			S3, T3 U22, P23										
24	Določitev I _{max}			S3, T3 U23										
25	Bilten NEIC				T13 P25									S13
26	Preverjanje odstopanja				S4, T4 U25, P26									
27	Podatki tujih seiz. centrov				T13 U26, P27									S13
28	Ponoven izračun lokacije				S4, T4 U27, P28									
29	Preverjanje lokacije				S4, T4 U28									
30	Mesečni izpis			T4 P30	S4	T4 P30								
31	Potrditev mesečnega izpisa			S3 U30	T3, T5 P31	S5 U30								
32	Odpošiljanje mesečnega izpisa				S4 U31									T4

Tabela 7: Prenovljena tabela delovnih in poslovnih procesov

Poslovni proces	Delovni proces	Aktivnost
Opazovanje potresov	Preliminarna analiza potresa in obveščanje o potresu	Sprejem obvestila
		Pregled zapisov
		Analiza digitalnih seizmogramov
		Lociranje potresa
		Potrdi analizo
		Kreiranje izpisa
		Pošiljanje izpisa
		Kreiranje obvestila
		Pošiljanje obvestila
		Kreiranje vprašalnikov
		Potrditev vprašalnikov
	Dnevna analiza potresov in priprava tedenskega biltena	Kontrola delovanja opazovalnic
		Kontrola delovanja mreže
		Dnevna analiza
		Lociranje potresov
		Potrditev analize
		Končni dnevni izpis
		ZAMG
		Ponoven izračun lokacij
		Izdelava tedenskega biltena
		Potrditev tedenskega izpisa
	Priprava mesečnega biltena	Analiza vprašalnikov
		Določitev intenzitete
		Določitev I _{max}
		Bilten NEIC
		Preverjanje odstopanja
		Podatki tujih seiz. centrov
		Ponoven izračun lokacije
		Preverjanje lokacije
		Mesečni izpis
		Potrditev mesečnega izpisa
		Odpošiljanje mesečnega izpisa

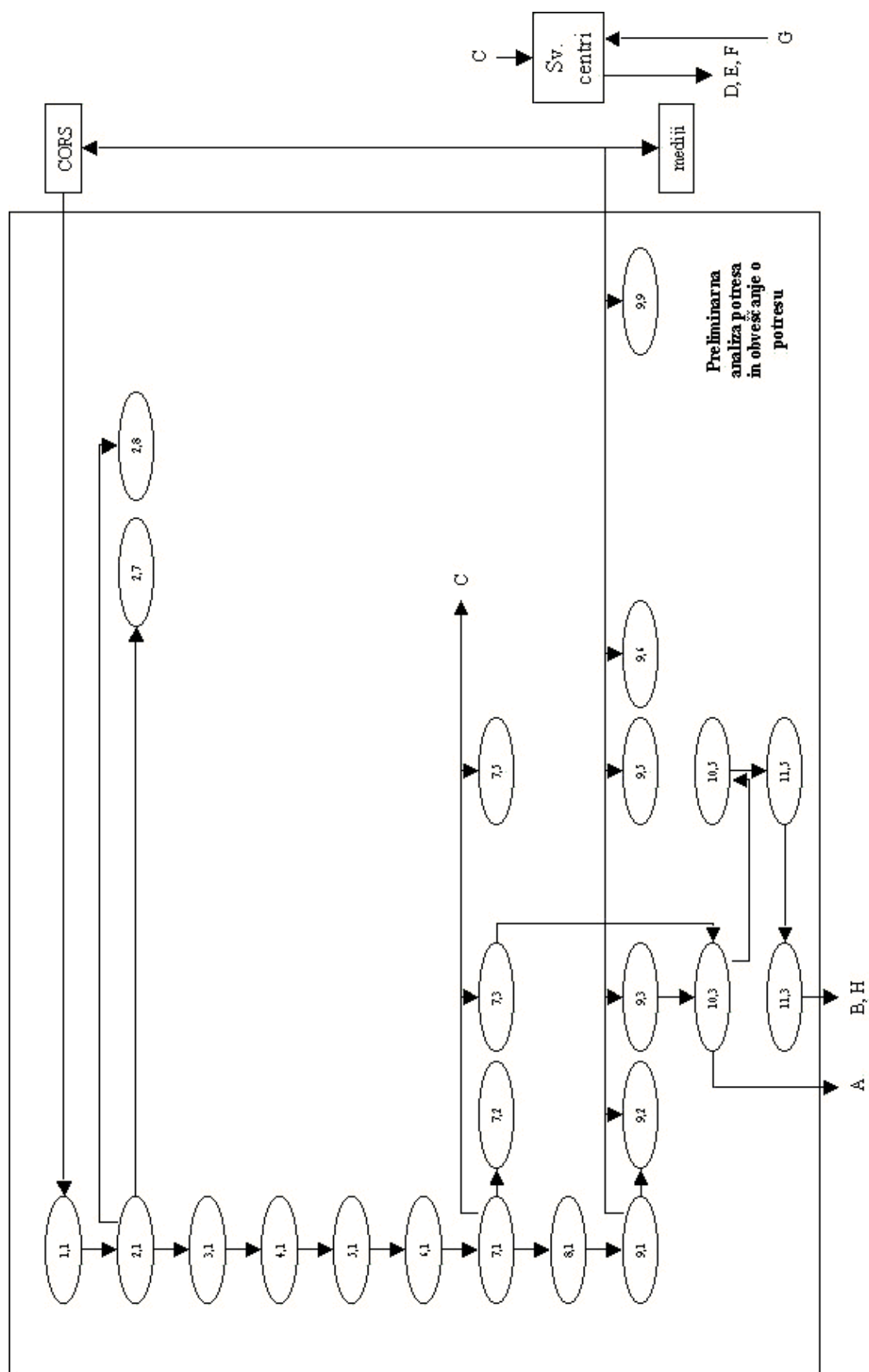
Tabela 8: Prenovljena tabela nalog

Activity	Task Code	Description	Time	Condition	Input/Output
1 - Sprejem obvestila	K(1.1)	Dežurni seizmolog prejme s CORS-a obvestilo, da so prebivalci Slovenije čutili potres.			Obvestilo
2 - Pregled zapisov	K(2.1)	Dežurni seizmolog pregleda digitalne seizmograme. Preveri, če delujejo povezave. V nasprotnem primeru obvesti SZI. Preveri, če delujejo opazovalnice. V primeru napak obvesti SZPO.		Zapisi so?	Obvestilo o delovanju opazovalnic, obvestilo o delovanju mreže
	K(2.7)	SZPO prejme obvestilo o napakah na opazovalnicah in ustrezno ukrepa.		Tip napake?	Obvestilo o delovanju opazovalnic
	K(2.8)	SZI prejme obvestilo o delovanju mreže in ustrezno ukrepa.		Tip napake?	Obvestilo o delovanju mreže
	K(3.1)	Dežurni seizmolog naredi analizo digitalnih seizmogramov. Za vsako opazovalnico določi fazo, vstopne čase, smer vstopa, ...		Potres je?	Analiza
4 - Lociranje potresa	K(4.1)	Dežurni seizmolog na podlagi analize izračuna lokacijo in magnitudo potresa.			Lokacija
5 - Potrdi analizo	K(5.1)	Dežurni seizmolog preveri, če je izračunana lokacija ustrezna. Če je potrebno, naredi popravke.		Lokacija dobra?	Lokacija, analiza
6 - Kreiranje izpisa	K(6.1)	Dežurni seizmolog kreira izpis potresa in vpiše rezultate izračuna lokacije.			Lokacija, izpis potresa
7 - Pošiljanje izpisa	K(7.1)	Dežurni seizmolog potrdi izpis in ga pošlje seizmologu, M-seizmologu, VSZS in svetovnim seizmološkim centrom.		Lokacija potresa?	Izpis potresa
	K(7.2)	Seizmolog prejme izpis potresa.			Izpis potresa
	K(7.3)	M-Seizmolog prejme izpis potresa. Ta mu je v pomoč pri kreiranju vprašalnikov o učinkih potresa.			Izpis potresa
	K(7.5)	VSZS prejme izpis potresa.			Izpis potresa
8 - Kreiranje obvestila	K(8.1)	Dežurni seizmolog kreira poročilo o potresu.			Poročilo o potresu
9 - Pošiljanje obvestila	K(9.1)	Dežurni seizmolog poročilu doda podatek, kje so potres čutili, in oceno intenzitete ter ga posreduje CORS-u, medijem, VSZS, VUZS, direktorju, seizmologu in M-seizmologu.		Lokacija potresa? Informacija o intenziteti?	Obvestilo, poročilo o potresu
	K(9.2)	Seizmolog prejme poročilo o potresu.			Poročilo o potresu
	K(9.3)	M-seizmolog prejme poročilo o potresu. Na podlagi le-tega kreira vprašalnike o učinkih potresa.			Poročilo o potresu
	K(9.5)	VSZS prejme poročilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(9.6)	VUZS prejme obvestilo o potresu in ustrezno ukrepa.		Magnituda, lokacija potresa	Poročilo o potresu
	K(9.9)	Direktor prejme obvestilo o potresu in ustrezno ukrepa.			Poročilo o potresu

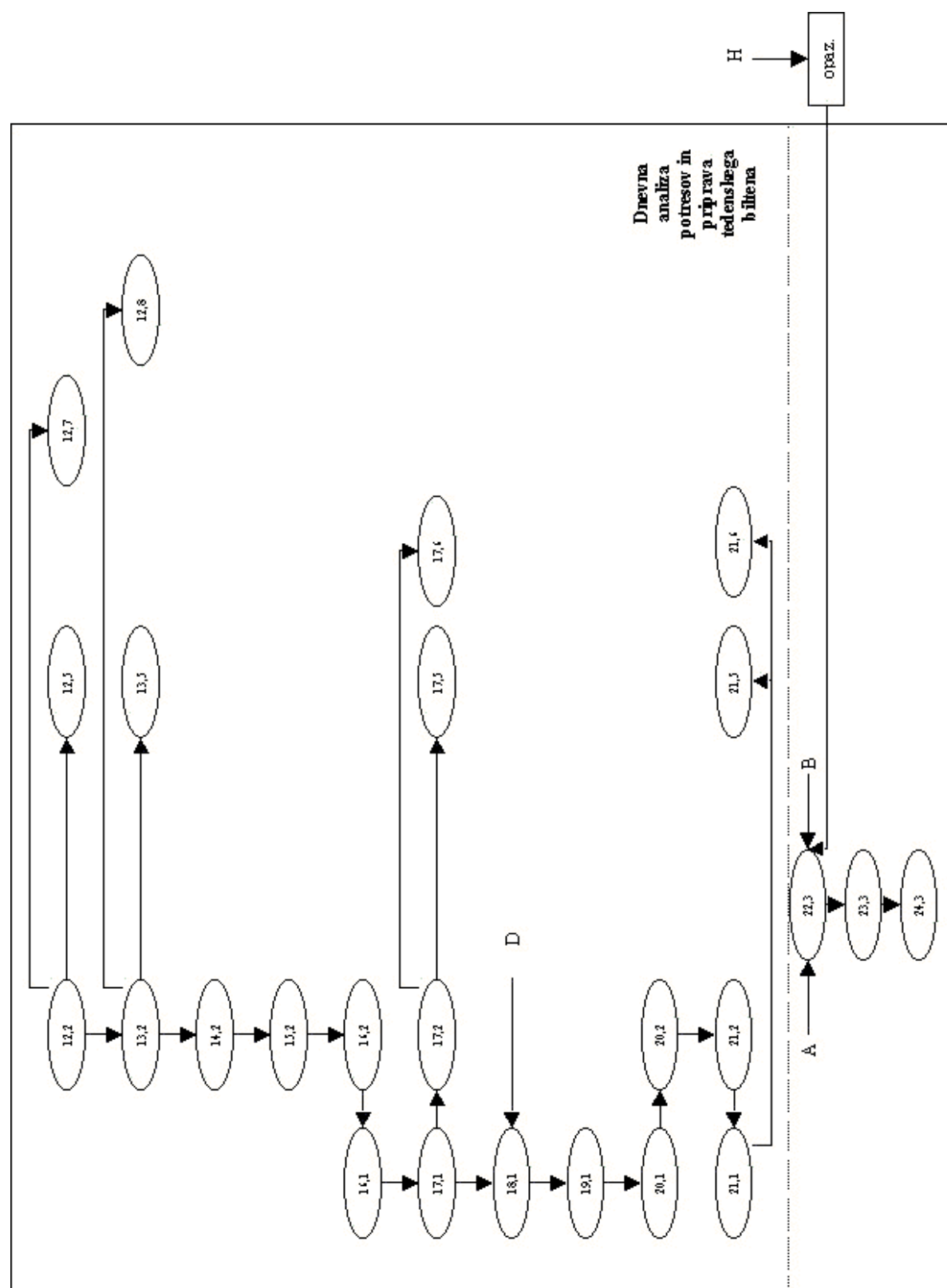
Activity	Task Code	Description	Time	Condition	Input/Output
10 - Kreiranje vprašalnikov	K(10.3)	M-seizmolog pripravi vprašalnik o učinkih potresa na ljudi, poslopja, naravo in jih posreduje VSZS v potrditev.		Magnituda, lokacija potresa, število klicev	Vprašalnik
	K(10.5)	VSZS prejme vprašalnik v potrditev.			Vprašalnik.
11 - Potrditev vprašalnikov	K(11.3)	M-seizmolog prejme od VSZS potrjen vprašalnik (če je potrebno ga spremeniti) in ga pošlje opazovalcem.			Vprašalnik
	K(11.5)	VSZS pregleda vprašalnik in ga potrdi ali zahteva popravke.		Območje, število	Vprašalnik
	K(12.2)	Seizmolog, ki opravlja dnevno analizo, preveri delovanje opazovalnic. O napakah obvesti VSZS in SZPO.		Opazovalnice delujejo?	Obvestilo o delovanju opazovalnic
12 - Kontrola delovanja opazovalnic	K(12.5)	VSZS prejme obvestilo o delovanju opazovalnic in, če je potrebno, ukrepa.		Tip napake?	Obvestilo o delovanju opazovalnic
	K(12.7)	SZPO prejme obvestilo o delovanju opazovalnic in ustrezno ukrepa.		Tip napake?	Obvestilo o delovanju opazovalnic
	K(13.2)	Seizmolog, ki je zadolžen za dnevno analizo, preveri delovanje mreže. O napakah obvesti VSZS in SZI.		Mreža deluje?	Obvestilo o delovanju mreže
13 - Kontrola delovanja mreže	K(13.5)	VSZS prejme obvestilo o delovanju mreže in, če je potrebno, ukrepa.		Tip napake?	Obvestilo o delovanju mreže
	K(13.8)	SZI prejme obvestilo o delovanju mreže in ustrezno ukrepa.		Tip napake?	Obvestilo o delovanju mreže
	K(14.2)	Seizmolog naredi dnevno analizo digitalnih seizmogramov.			Analiza
14 - Dnevna analiza					
15 - Lociranje potresov	K(15.2)	Seizmolog izračuna lokacije in magnitude potresov, ki jih je analiziral.			Lokacija
16 - Potrditev analize	K(16.1)	Dežurni seizmolog prejme izračune in jih preveri. Če je potrebno zahteva popravke.		Odstopanja	Analiza, lokacija
	K(16.2)	Seizmolog posreduje izračune dežurnemu seizmologu, da jih ta preveri.			Analiza, lokacija
17 - Končni dnevni izpis	K(17.1)	Dežurni seizmolog potrdi izračune.		Izračuni so vredni?	Analiza, lokacija
	K(17.2)	Seizmolog pripravi končni dnevni izpis in ga pošlje ostalim seizmologom ter VSZS.			Dnevni izpis
	K(17.5)	VSZS prejme dnevni izpis.			Dnevni izpis
	K(17.6)	VUZS prejme dnevni izpis.			Dnevni izpis
18 - ZAMG	K(18.1)	Dežurni seizmolog prejme avstrijske analize (ZAMG) in jih doda dnevni analizi.		ZAMG deluje?	Analiza

Activity	Task Code	Description	Time	Condition	Input/Output
19 - Ponoven izračun lokacij	K(19.1)	Ko doda podatke z ZAMG, dežurni seizmolog ponovno izračuna lokacije potresov in njihovo magnitudo.			Analiza, lokacija
20 - Izdelava tedenskega izpisa	K(20.1)	Dežurni seizmolog kreira tedenski izpis potresov in ga posreduje seizmologu, da ga preveri.			Tedenski izpis
	K(20.2)	Seizmolog prejme tedenski izpis, ga preveri in predlaga popravke.		Izračuni dobri? Kaj manjka?	Tedenski izpis
	K(21.1)	Dežurni seizmolog ustrezno popravi tedenski izpis in ga nato pošlje VSZS, VUZS in svetovnim seizmološkim centrom.			Tedenski izpis
21 - Potrditev tedenskega izpisa	K(21.2)	Seizmolog potrdi tedenski izpis		Tedenski izpis vreden?	Tedenski izpis
	K(21.5)	VSZS prejme tedenski izpis.			Tedenski izpis
	K(21.6)	VUZS prejme tedenski izpis.			Tedenski izpis
	K(22.3)	Ko zbere M-seizmolog vprašalnike, jih zanalizira.		Vprašalniki so vrnjeni? Vprašalniki so izpolnjeni?	Vprašalniki
23 - Določitev intenzitete	K(23.3)	S pomočjo vprašalnikov določi intenziteto potresa po krajih.		Število pravilno izpolnjenih vprašalnikov	Intenziteta
24 - Določitev I _{max}	K(24.3)	M-seizmolog določi maksimalno intenziteto potresa in jo vpiše k analizi.			Intenziteta, analiza
25 - Bilten NEIC	K(25.4)	B-seizmolog prejme bilten NEIC in ga primerja z slovenskimi rezultati.		Potresi vsebovani v biltenu NEIC?	Bilten NEIC, analiza, lokacija
26 - Preverjanje odstopanja	K(26.4)	B-seizmolog še enkrat preveri podatke, pri kateri opazi večja odstopanja in naredi ustrezne popravke.		Ujemanje lokacij, časa?	Bilten NEIC, analiza, lokacija
27 - Podatki tujih seiz. centrov	K(27.4)	B-seizmolog doda podatke, ki jih prejme od tujih seizmoloških centrov.		Oddaljenost opazovalnic, magnituda potresa	Analiza
28 - Ponoven izračun lokacije	K(28.4)	B-seizmolog ponovno preračuna lokacije potresov.			Analiza, lokacija
29 - Preverjanje lokacije	K(29.4)	B-seizmolog preveri lokacije potresov.		Odstopanja	Analiza, lokacija
30 - Mesečni izpis	K(30.3)	M-seizmolog prejme mesečni izpis, da ga preveri. Ob ugotovljenih napakah ustrezno ukrepa.		Večja odstopanja, manjkajoči podatki	Mesečni izpis
	K(30.4)	B-seizmolog pripravi mesečni izpis in ga posreduje M-seizmologu in VSZS. Ob ugotovljenih napakah ustrezno ukrepa.		Večja odstopanja, manjkajoči podatki	Mesečni izpis
	K(30.5)	VSZS prejme mesečni izpis in ga pregleda.			Mesečni izpis
	K(31.3)	M-seizmolog potrdi mesečni izpis		Mesečni izpis vreden?	Mesečni izpis
	K(31.4)	B-seizmolog naredi ustrezne popravke mesečnega izpisa.			Mesečni izpis
31 - Potrditev mesečnega izpisa	K(31.5)	VSZS potrdi mesečni izpis.		Mesečni izpis vreden?	Mesečni izpis
	K(32.4)	B-seizmolog posreduje mesečni izpis potresov svetovnim seizmološkim centrom, kjer ga vnesejo v svoje baze podatkov.			Mesečni izpis

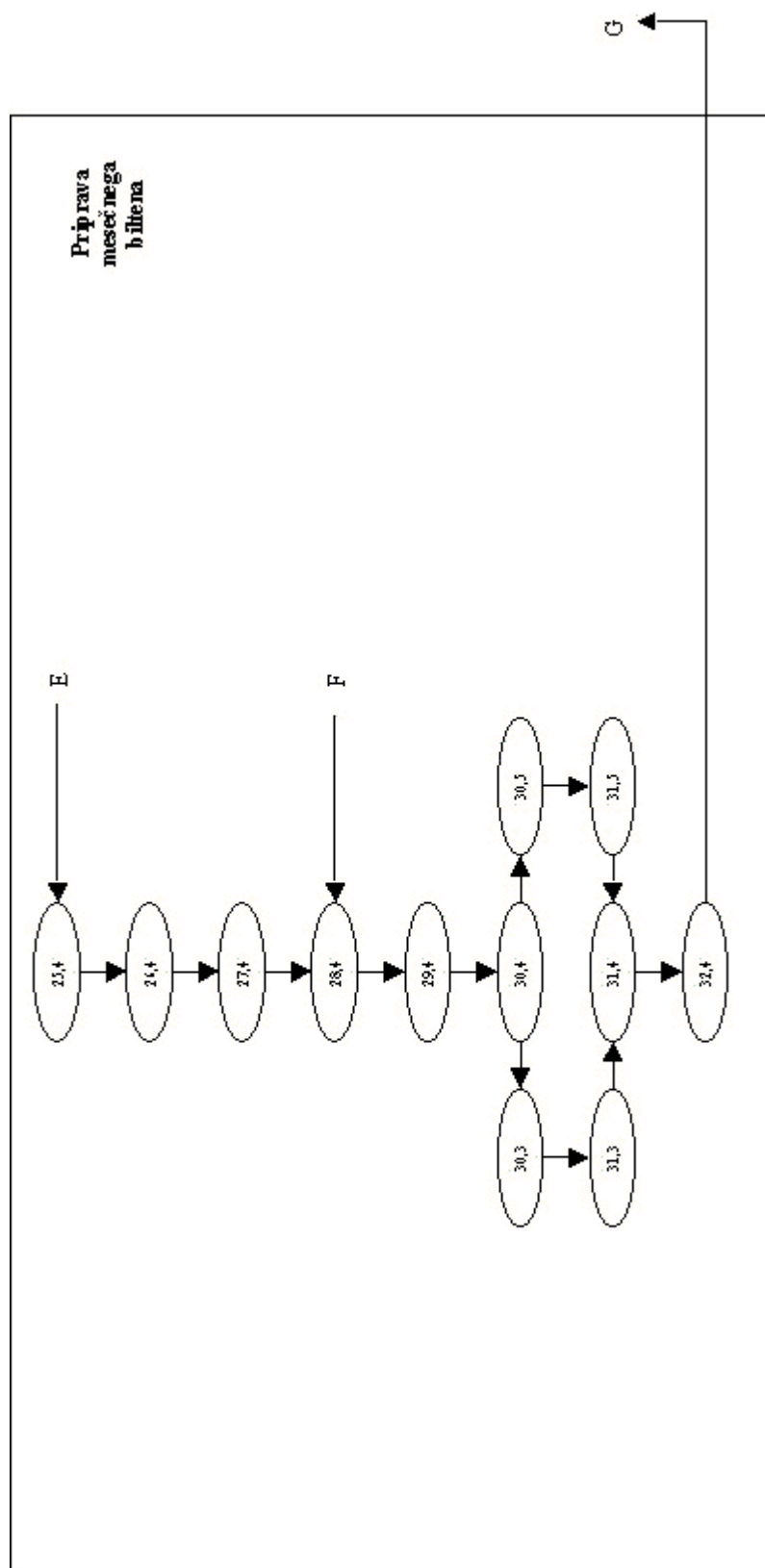
Slika 1: Procesni model (1/3)



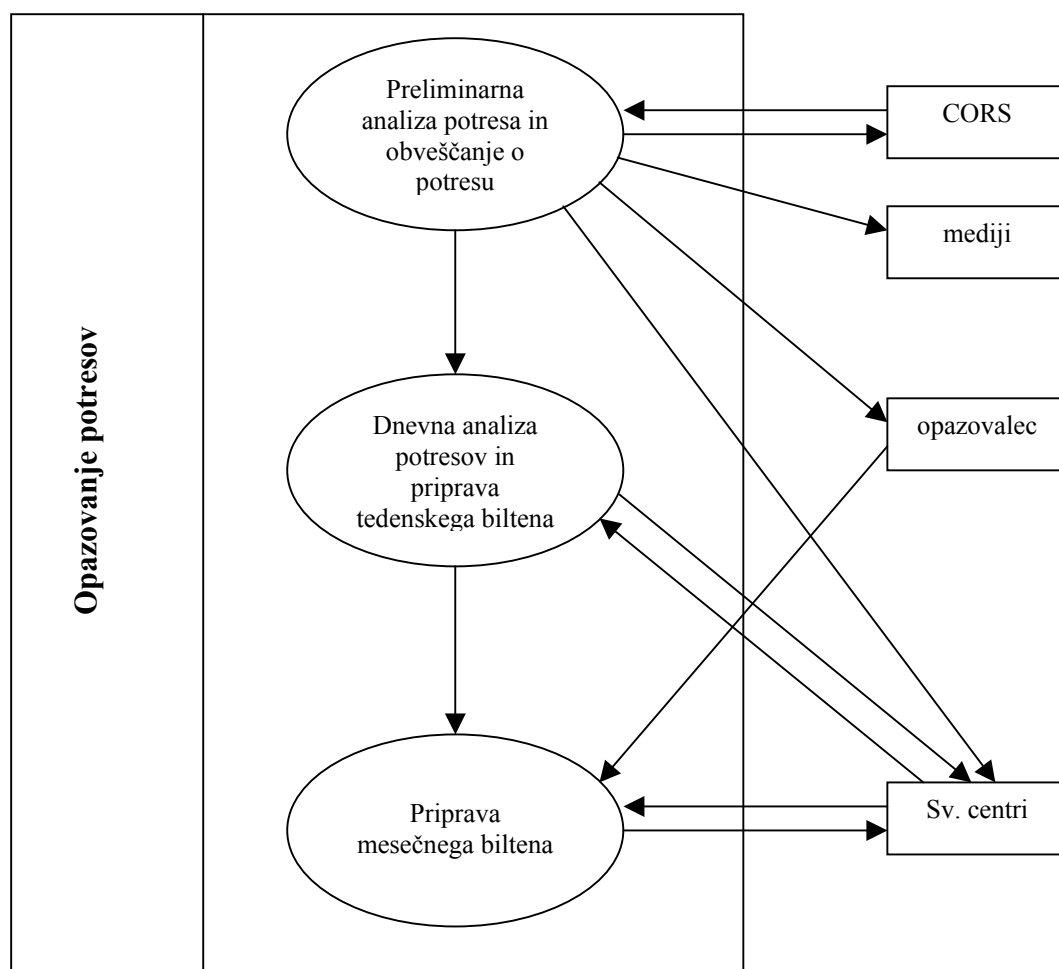
Slika 2: Procesni model (2/3)



Slika 3: Procesni model (3/3)



Slika 4: Diagram delovnih procesov



Slika 5: Podrobnejši opis razredov

V razredih so vpisane le operacije, ki sem jih definirala v fazi načrtovanja sistema in so zbrane v tabeli operacij (glej tabelo 15 na str. 81). Poleg teh imajo razredi tudi operacije za kreiranje, spreminjanje, brisanje in pregledovanje objektov.

Dogodek Dog_id# Lok_id Datum dogodka Status Avtor Datum_vnosa Datum_spremembe	Lokacija Lok_id# Dog_id Status Datum Ura Minuta Sekunda Geogr. širina Geogr. dolžina Globina Nsta Nass Ndef F-globine F-časa F-lokacije Tip Magnituda Tip magnitude Avtor Datum_vnosa Datum_spremembe Lociraj, PreveriLokacijo	LokFaza Lok_id# Faz_id# Residium Razdalja Azimut	Faza Faza_id# Status Ime faze Koda mreže Koda postaje Datum Ura Minuta Sekunda Natančnost Prvi premik Utež Komponenta Amplituda Perioda Avtor Datum_vnosa Datum_spremembe KreirajAnalizo, PopraviAnalizo, PotrdiAnalizo, DodajPodatke, PrimerjajAnalize, PreveriOdstopanja	Opazovalec Opaz_id# Ime Priimek Ulica Hišna št. Poštna številka Ocena opaz. Gostota mreže Avtor Datum_vnosa Datum_spremembe
Vprašalnik Vpras_id# Tip izbora Gost. mreže izbora Min. ocena opaz. Geogr. širina Geogr. dolžina Zunanji polmer Notranji polmer KreirajVprasalnik, PopraviVprasalnik, PotrdiVprasalnik, PošljiVprasalnik, AnalizaVprasalnika	Bilten Bilt_id# Datum kreiranja Tip	VprasOpaz Vpras_id# Opaz_id# Datum vrnitve PošljiVprasalnik, AnalizaVprasalnika	VprasDog Vpras_id# Dog_id# Število KreirajVprasalnik, PopraviVprasalnik, PotrdiVprasalnik	Kraj Poštna številka Ime kraja Geogr. širina Geogr. dolžina Nadmorska višina Gauss-Krug. proj. X Gauss-Krug. proj. Y Geologija tal
Intenziteta Int_id# Intenziteta Poštna številka Napaka DoločiIntenziteto, DoločiImax	Merilnik Merilnik_id# Tip Serijska številka Proizvajalec Servis Garancija	Kalibracija Kalibracija_id# Merilnik_id# Datum kalibracije Vrednosti Pripombe	MerMesto Koda postaje# Ime postaje Merilnik_id# Geogr. širina Geogr. dolžina Nadmorska višina Lokalna strukt. tal Poštna številka	VprasOpazInt Vpras_id# Opaz_id# Int_id# Ocena vprašalnika DoločiIntenziteto
Naslov Naslov_id# Inštitucija Elektr. naslov Opis Status	Izpis Izpis_id# Datum kreiranja KreirajIzpis, PošljiIzpis	IzpNasl Izpis_id# Naslov_id# Datum pošiljanja Število PošljiIzpis	Obvest Obvest_id# Datum obvestila Datum napake	ObvMreza Koda postaje Tip povezave Napaka KreirajObvestiloDM
ObvNasl Obvestilo_id# Naslov_id# Datum pošiljanja Število PošljiObvestilo	TedNas Bilt_id# Naslov_id# Datum pošiljanja PošljiTedenskiIzpis	MesNas Bilt_id# Naslov_id# Datum pošiljanja PošljiMesečniIzpis	ObvestiloPotr Obvestilo_id# Datum kreiranja Ocena intenzitete Dodatni podatki KreirajObvestilo, PošljiObvestilo	ObvMer Merilnik_id# Tip napake KreirajObvestiloDO
DnevNas Bilt_id# Naslov_id# Datum pošiljanja PošljiDnevniIzpis		DnevBilt Dan KreirajDnevniIzpis, PošljiDnevniIzpis	TedBilt Teden KreirajTedenskiIzpis, PotrdiTedenskiIzpis, PošljiTedenskiIzpis	MesBilt Mesec KreirajMesečniIzpis, PotrdiMesečniIzpis, PopraviMesečniIzpis, PošljiMesečniIzpis