

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

Asko Oruste 153677IAPM

**ÄRI- JA TEHNILISE ANALÜÜSI
DOKUMENTEERIMISE LAHENDUSE
VÄLJATÖÖTAMINE NING RAKENDAMINE
REGISTRITE JA INFOSÜSTEEMIDE
KESKUSE NÄITEL**

Magistritöö

Juhendaja: Maili Markvardt

MSc

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Asko Oruste

08.05.2017

Annotatsioon

Antud magistritöö peamiseks eesmärgiks on luua ühtne sobilik standartne tööprotsessi mall infosüsteemi dokumentatsiooni loomiseks. Samuti töötada välja metoodika tööprotsessi läbiviimiseks, mis toetab loodud tööprotsessi malli.

Töö käigus selgitatakse, kuidas antud eesmärgid saavutatakse. Esmalt uuritakse, kuidas näevad välja rakendatavas ettevõttes töö kirjutamise hetkel kasutusel olevad tööprotsessid ja vormid. Seejärel valitakse alternatiivsete lahendustena kolm kolmanda osapoole tarkvara, mida analüüsitakse ning hinnatakse esmavajaduste vastavust.

Töö tulemusena on valminud analüüsiv ning hindav kirjeldus olemasolevate lahenduste lähenemisest sarnasele probleemile. Samuti on töö tulemusena valminud süsteemianalüüs süsteemile, mis muudab analüüsi dokumenteerimise protsessi ühtlasemaks ning lõppkasutajale kiiremaks ja ülevaatlikumaks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 67 leheküljel, 9 peatükki, 18 joonist, 16 tabelit.

Abstract

The elaboration and implementation of business and technical analysis documenting solution for Centre of Registers and Information Systems

The goal of this master's thesis is to design a single standard workflow template for information systems documentation. Also work out the methodology to implement the workflow that supports the designed template.

During the course of the thesis it is explained how these goals are achieved. First, it is researched how the current processes and forms look like in the proposed company. With this research the initial needs are put to place for alternative solutions and for the proposed solution. Three alternative third-party solutions are selected for analysis and the evaluation of initial needs.

The outcome for this thesis is analytical evaluation for the existing solutions on the approach to a similar problem. The second outcome of this thesis is a system analysis documentation for the proposed solution that should make the analysis documentation process more consistent and faster and clearer for the end-user.

The thesis is in Estonian and contains 67 pages of text, 9 chapters, 18 figures, 16 tables.

Lühendite ja mõistete sõnastik

RIK	Registrite ja Infosüsteemide Keskus
SAD	Süsteemi analüüs ja disain

Sisukord

Jooniste loetelu	9
Tabelite loetelu	10
1 Sissejuhatus	11
1.1 Taust ja probleem	11
1.2 Ülesande püstitus	11
1.3 Metoodika	12
1.4 Ülevaade tööst	12
2 Registre ja Infosüsteemide Keskus (RIK)	14
2.1 Organisatsioonist	14
2.2 Osapooled	14
3 Ülevaade RIK-is kasutusel olevatest äri- ja süsteemianalüüsi tavadest	15
3.1 Ärianalüüs	15
3.1.1 Analüüsi teostamise protsess	15
3.1.1.1 Osapooled	16
3.1.1.2 Kasutusel olevad head tavad/praktikad	16
3.1.1.3 Kasutusel olevad halvad tavad/praktikad	17
3.1.2 Dokumenteerimise praktikad, metoodikad	17
3.1.2.1 Nõuete kirjeldamine ning haldamine	18
3.1.2.2 Osapoolte vaheline kommunikatsioon	20
3.1.2.3 Kasutatavad tehnoloogilised lahendused dokumenteerimisel	20
3.2 Süsteemianalüüs	21
3.2.1 Analüüsi teostamise protsess	22
3.2.1.1 Osapooled	22
3.2.1.2 Süsteemianalüüsi tüübid	22
3.2.1.3 Kasutusel olevad head tavad/praktikad	25
3.2.1.4 Kasutusel olevad halvad tavad/praktikad	25
3.2.2 Dokumenteerimise praktikad, metoodikad	26
3.2.2.1 Nõuete/reeglite kirjeldamine	26
3.2.2.2 Kasutatavad tehnoloogilised lahendused dokumenteerimisel	27
4 Alternatiivsed lahendused	29
4.1 Atlassian Confluence ülevaade	29
4.1.1 Teadmuspank	30

4.1.2	Dokumentatsioon ja dokumenteerimine.....	30
4.1.3	Välised dokumendid.....	30
4.1.4	Maksumus	31
4.2	Axure RP ülevaade	32
4.2.1	Prototüüpimine	33
4.2.2	Dokumentatsioon ja dokumenteerimine.....	33
4.2.3	Välised dokumendid.....	34
4.2.4	Maksumus	34
4.3	Targetprocess ülevaade.....	35
4.3.1	Teadmuspank	35
4.3.2	Dokumentatsioon ja dokumenteerimine.....	35
4.3.3	Välised dokumendid.....	36
4.3.4	Maksumus	36
4.4	Üldised probleemid.....	36
5	Võimalikud analüüsi meetodikad	38
5.1	Agiilne modelleerimine ja dokumenteerimine	38
5.1.1	Modelleerimise praktikad.....	38
5.1.2	Dokumenteermise praktikad	39
5.2	Kerge dokumentatsioon (<i>Lightweight Documentation</i>).....	39
5.2.1	Mõõdikud	39
5.3	<i>Lean</i> printsiibid.....	39
5.3.1	Mõõdikud	40
6	Alternatiivsete lahenduste hinnang	41
6.1	Hinnangute selgitused.....	42
6.1.1	Atlassian Confluence.....	42
6.1.2	Axure RP	44
6.1.3	Targetprocess	45
7	Pakutav lahendus	48
7.1	Nõuded.....	48
7.1.1	Andmeväljad	48
7.1.2	Nõude liigid.....	51
7.1.2.1	Nõude liikide omavahelised seosed ning arhitektuur	53
7.1.3	Nõude staatused.....	56

7.1.3.1	Äri- ja funktsionaalsed nõuded.....	57
7.1.4	Vormid ja prototüübid.....	60
7.2	Tellimine.....	63
7.2.1	Tellimuse koostamine ja esitamine	64
7.2.1.1	Toote omaniku vaade.....	64
7.2.1.1.1	Äri- ja funktsionaalsed nõuded	67
7.2.1.1.2	Vormid ja prototüübid.....	67
7.2.1.2	Ärianalüütiku vaade.....	68
7.2.1.2.1	Äri- ja funktsionaalsed nõuded	69
7.2.1.2.2	Vormid ja prototüübid.....	70
7.3	Taasesitus ja dokumenteerimine.....	70
7.3.1	Nõuete taasesitus	70
7.3.2	Tellimuste prototüüpimine	72
7.4	Rollid ja õigused.....	74
8	Pakutava lahenduse hinnang	75
9	Kokkuvõte.....	78
10	Kirjanduse loetelu	79

Jooniste loetelu

Joonis 1: Nõuete töötlemise alamosad.	18
Joonis 2: Atlassian Confluence haldusliides.	29
Joonis 3: Axure RP haldusliides.....	33
Joonis 4: Axure RP maksumuse plaanid.	34
Joonis 5: Targetprocess haldusliides.	35
Joonis 6: Targetprocess maksumuse plaanid.	36
Joonis 7: Projekti, funktsionaalsuse ning nõuete omavahelised seosed.	54
Joonis 8: Nõude seoste vormiline näidis.	55
Joonis 9: Nõuete seoste arhitektuur.....	55
Joonis 10: Nõude staatuste struktuur.....	59
Joonis 11: Ärinõude vaatamise vormi näidis.	60
Joonis 12: Nõude tagasiside vormi näidis.	61
Joonis 13: Nõude muutmise modaal.	62
Joonis 14: Uue nõude lisamise modaal.	63
Joonis 15: Tellimuse vaatamise vorm.	68
Joonis 16: Arendustellimuse lisamise vorm.....	70
Joonis 17: Axure RP vormi komponentide kirjeldamine.	71
Joonis 18: Weps rakenduse näide.....	73

Tabelite loetelu

Tabel 1: Confluence maksumuse sõltuvus kasutajate arvust (kasvav meeskond).	31
Tabel 2: Confluence maksumuse sõltuvus kasutajate arvust (server).	31
Tabel 3: Confluence maksumuse sõltuvus kasutajate arvust (andmeladu).	32
Tabel 4: Alternatiivsete lahenduste hinnangute värvikoodid ning selgitused.	41
Tabel 5: Alternatiivsete lahenduste üldised hinnangud.	41
Tabel 6: Nõude vaatamise modaali peamised andmeväljad.	48
Tabel 7: Nõude lisamise/muutmise modaali peamised andmeväljad.	50
Tabel 8: Nõude muutmise/lisamise muud andmeväljad.	51
Tabel 9: Nõude liigid ja selgitused.	51
Tabel 10: Nõude staatused ja selgitused	56
Tabel 11: Tellimuse koostamise ja esitamise andmeväljad.	64
Tabel 12: Tellimuse prioriteetide kirjeldused.	66
Tabel 13: Tellimuse staatuste kirjeldused.	66
Tabel 14: Rollide selgitused.	74
Tabel 15: Õiguste kirjeldused CRUD maatriksina.	74
Tabel 16: Pakutava lahenduse üldised hinnangud.	75

1 Sissejuhatus

1.1 Taust ja probleem

Igal projektijuhil, analüütikul, arendajal või testijal on oma valdkonnast lähtuvalt enda kindel käekiri nii dokumenteerimisel, planeerimisel, arendamisel jne. Enamasti sõltub käekiri arendusprojekti suurusest ning arendusprotsessist. Oma lõputöös keskendun arendusprojekti analüüsi dokumenteerimise tööprotsessile – analüüsi protsess/praktikad, dokumentatsioon (äri- ja tehniline analüüs), juhendid, planeerimine. Töökohast (Registrite ja Infosüsteemide Keskus) tulenevalt on lõputöö fookus antud protsessi standardi/metoodika loomisele ning ühtlustamisele üle kõikide töös olevate kui ka tulevaste ettevõtte projektide raames.

Hetkel puudub ettevõttes ühtne metoodika või protsess arenduse projekti dokumenteerimisel või analüüsimisel. Olukorrast lähtuvalt kasutavad erinevad analüütikud mitmeid erinevaid praktikaid, vastavalt oma kogemusele, muutes tellijale või lõppkasutajale dokumentatsioonist aru saamise keerulisemaks – olenevalt dokumenteerijast, tuleb lugeda igat dokumenti eri moodi. Samuti kannatab analüütikute omavaheline suhtlus ning üksteisest arusaam - sellega seoses on raskendatud ühest projektist teise integreerimine või mõne teise analüütiku tööde ülevõtmine.

Ettevõtte töötajate seas on antud probleem seni läbi käinud väga üldistavalt ning reaalsed lahendused puuduvad, mistõttu vajadus ettevõtte sisesele ühisele protsessile on olemas.

1.2 Ülesande püstitus

Töö peamised eesmärgid on sõnastatud järgnevalt:

- Luua ettevõttele sobilik standardne tööprotsessi mall infosüsteemi dokumentatsiooni loomiseks.
- Töötada välja metoodika tööprotsessi läbiviimiseks, mis toetab loodud tööprotsessi malli.
- Analüüsida ning luua tööprotsessi malli ning metoodikat toetav näidis-lahendus koos minimaalse funktsionaalsusega vormidel põhineva prototüübiga.

1.3 Metoodika

Esmalt tuleb selgeks teha, kuidas näevad välja ettevõttes praegused kasutusel olevad tööprotsessid ja vormid ning millised on ühendavad kokkulepped. Selleks on töö autor viinud läbi mitmeid intervjuusid ettevõtte spetsialistidega (äri- ning süsteemianalüütikud) ning samuti panustab omaenda kogemusega antud teema vallas.

Seejärel on valitud kolm kolmanda osapoolse tarkvaralist lahendust, mis osaliselt või täielikult, kuid mugandustega, toetavad kirjeldatud protsesse ja vorme. Iga lahenduse juures on valitud ning ära kirjeldatud kolm peamist sisuliselt toetavat funktsionaalsust ning üks üldine funktsionaalsus. Kõik kolm lahendust on hinnatud ning valideeritud 13 erineva mõõdiku näol.

Hindamise ja valideerimise süsteem (mõõdikud) on töö autori poolt loodud ning iga mõõdikut on eraldi põhjalikult selgitatud. Süsteem põhineb erinevatel arendusprintsipiidel ning dokumenteerimise praktikatel, mis keskenduvad rohkem agiilsele lähenemisele. Mõõdikute lõplikul valikul on lähtutud ettevõtte strateegiast ning lähenemisest töö teema valdkonnale individuaalsete arendusprojektide raames.

Töös pakutakse samuti välja omapoolne põhiline idee ning lahendus. Lahenduse funktsionaalsused on ideeliselt välja töötatud ja loodud eesmärgiga toetada olemasolevaid ühiseid protsesse kaasates alternatiivsete lahenduste tugevaid omadusi ja pakkudes välja uusi lähenemisi antud probleemile. Sarnaselt kolmanda osapoolte lahendustele hinnatakse omapoolset lahendust loodud hindamise süsteemi alusel. Ära on kirjeldatud kolm põhilist funktsionaalsust, millest kaks on realiseeritud – minimaalse funktsionaalsusega prototüübi näol.

Töö tulemusena valmib analüüsiv ning hindav kirjeldus olemasolevate lahenduste lähenemisest sarnasele probleemile. Samuti valmib töö tulemusena süsteemianalüüs süsteemile, mis muudab analüüsi dokumenteerimise protsessi ühtlasemaks ning lõppkasutajale kiiremaks ja ülevaatlikumaks. Töö viimase tulemuse valideerimiseks on loodud minimaalse funktsionaalsusega vormidel põhinev veebipõhine näiterakendus. Vastavalt analüüsile on näiterakendust võimalik hõlpsasti taas-luua.

1.4 Ülevaade tööst

Töö on jaotatud kolme peamise osa vahel. Töö esimeses osas antakse ülevaade organisatsioonist ning selle kõikidest osapooltest. Samuti kirjeldatakse ettevõttes hetkel

kasutusel olevaid analüüsimise ja dokumenteerimise protsesse ning nende minimaalseid vajadusi.

Töö teine osa keskendub alternatiivsete lahenduste ja nende funktsionaalsuste hindamisele ning ülevaatlilikule analüüsile. Hindamiseks ja valideerimiseks on autor loonud eraldiseisva hindamise süsteemi, mis antud osas on samuti ära kirjeldatud.

Töö kolmandas osas selgitatakse välja töötatud omapoolset lahendust ning selle toimimise põhimõtet. Lahendust hinnatakse juba mainitud hindamise süsteemi alusel. Lõpuks valideeritakse töö tulemuse näiterakenduse realiseerimisega ning antakse hinnang selle kasutusele ja võimalikele edasiarendustele.

2 Registrate ja Infosüsteemide Keskus (RIK)

2.1 Organisatsioonist

Registrate ja Infosüsteemide Keskus (RIK) on Justiitsministeeriumi haldusala asutus, mille eesmärk on luua häid integreeritud e-teenuseid pakkuv innovaatiline keskkond riigihaldus-, õigus- ja kriminaalpoliitika efektiivsemaks rakendamiseks. RIK arendab ning haldab riigile ja kodanikele väga olulisi registreid ja infosüsteeme nagu näiteks e-äriregister, e-notar, e-kinnistusraamat, kohtuinfosüsteem, kinnipeetavate register, karistusregister, e-toimik, elektrooniline Riigi Teataja [1].

RIK-i peamiseks tegevusvaldkonnaks on Justiitsministeeriumi valitsemisala info- ja sidesüsteemide arendamine ja haldamine ning info- ja sidetehnoloogiaalaste teenuste pakkumine [2]. Vastavalt tegevusvaldkonnale on RIK-i missiooniks olla riigisektorile asjatundlik ja paindlik partner, pakkudes kvaliteetset IT-teenust ideest teostuseni. RIK-i visiooni on olla hinnatuim riiklike IT-teenuste arendaja ja haldaja.

2.2 Osapooled

Antud magistritöö on eelkõige, skoobi kitsendamise eesmärgil, suunatud infosüsteemide, teenuste või eraldiseisvate rakenduste äri- ning tehnilise analüüsi teostajatele ja dokumenteerijatele. Vastavalt RIK-i struktuuriüksuste ülesehitusele keskendub töö arendusosakonna töötajatele (süsteemianalütikud, tehnilised arhitektid) ning kõikidele sisuosakondadele – riikliku sunni registre osakond, kohturegistrate osakond, tugiinfosüsteemide osakond, õiguskaitse infosüsteemide osakond, mis puutuvad kokku rohkem ärilise poolega ning teostavad ärianalüüsi (ärianalütikud, infosüsteemi haldurid, tehnilised arhitektid).

Magistritöös pakutavat lahendust on võimalik laiendada kõikidele infosüsteemide arendusega kokku puutuvatele osapooltele (projektijuhid, testijad, arendajad, tellijad, toote omanikud jne).

3 Ülevaade RIK-is kasutusel olevatest äri- ja süsteemianalüüsi tavadest

Analüüs on tegevus, mille sisuks on millegi (alg)komponentideks/(alg)osadeks lahutav uurimine ja eritlus. Leitud komponentide/osade omavaheliste suhete ilmutamine ja struktureerimine [3].

RIK-is on infosüsteemide ja teenuste analüüsi teostamine jaotatud selgelt kaheks:

- ärianalüüs, mille peamine eesmärk on kaardistada tellija(te) poolt infosüsteemile seatud vajadusi ja eesmärke ning samuti infosüsteemi enda sisulisi reegleid ja ärinõudeid;
- süsteemianalüüs, mis tegeleb peamiselt infosüsteemi disaini ning tehnilise lahenduse loomisega, arvestades sealjuures ärianalüüsi poolt paika pandud vajaduste ning eesmärkidega.

Infosüsteemi dokumentatsioon ning selle selgus sõltub suuresti mõlema jaotusega seotud osapoolte kommunikatsioonist ning koostööst.

3.1 Ärianalüüs

Ärianalüüs ehk arendustellimus on kirjeldus kliendi vajadustest ning abivahend vähendamaks arenduskulusid. Ärianalüüsi eesmärgiks on anda arendusosakonnale lähtematerjal tellitava projekti eelanalüüsi koostamiseks ja sisend süsteemianalüüsi tegemiseks [4].

Ärianalüütiku ülesanneteks RIK-s on:

- Äriprotsesside ning kasutajavajaduste analüüs ja dokumenteerimine;
- Valminud rakenduste arendustellimusele vastavuse hindamine;
- Infosüsteeme puudutavate nõuete kirjeldamine;
- Uute teenuste ja lahenduste analüüs;
- Infosüsteeme puudutavate parandusettepanekute tegemine;
- Äriprotsesse ja infosüsteemide arendusi puudutava dokumentatsiooni koostamine ja korrashoid, sh ajakohastamine.

3.1.1 Analüüsi teostamise protsess

Järgnev peatükk annab põgusa ülevaate kuidas ärianalüüsi protsess RIK-s toimib, kes ja kuidas on kaasatud osapooltena, millised on kasutusel olevad head tavad ja halvad praktikad.

Suures pildis on analüüsi teostamise protsess iga projekti raames erinev ning sõltub suuresti projekti ja töö iseloomust ning lõpptulemuse sisust. Samuti projektimeeskonna liikmete kokkuleppetest.

3.1.1.1 Osapooled

Ärianalüüsi teostamise protsessi on kaasatud mitmed osapooled, kellest peamised on ärianalüütik, tellija või lähteülesande koostaja, lõppkasutaja. Vajadusel on samuti kaasatud süsteemianalüütikud, arendajad, testijad, infosüsteemi haldurid (kusjuures võivad haldurid olla ka ärianalüütiku, tellija ja lõppkasutaja rollis), infrastruktuuride osakonna esindaja, infoturbe talituse esindaja, kasutusmugavuse eksperdid, asutusvälised infosüsteemi puudutava valdkonna eksperdid (nende alla käivad sisulised kui ka tehnilised eksperdid).

Protsessis omavad osapooled ekspertliku arvamuse avaldaja rolli ning samuti kontrolliva vaatleja rolli. Viimase puhul võib näitena välja tuua tellijat, kes valideerib, et tema kõikide vajaduste ja eesmärkidega on arvestatud.

3.1.1.2 Kasutusel olevad head tavad/praktikad

Parima tulemuse tagamiseks on asutuse siseselt loodud ärianalüüsi koostamise hea tava dokument [4], mis on kogumik nõuannetest, millega võiks hea ärianalüütik oma igapäevatöös arvestada. Olulisemad neist, mida ärianalüütikud peamiselt jälgivad on:

- **Teiste kaasamine** – seda tava rakendatakse kõikide osapoolte suhtes võimalikult varakult, analüüsi alguses. Peamised eesmärgid on sellel kogemuste vahetamine ning teiste arvamuste, plusside-miinuste arvestamine;
- **Kasutajate vajadustega arvestamine** – tava hõlmab enda alla nii arvestamist kasutajate vajadustega ning tellitava teenuse/funktsionaalsuse kasutajasõbralikusega ja –mugavusega. Viimase arvestamise puhul on avaliku sektori jaoks abimaterjalina loodud käsiraamat kasutajasõbralike e-teenuste disainimise kohta [5];
- **Äriprotsesside modelleerimine** – antud tava eesmärgiks on vähendada paberprotsesside vastavust infosüsteemide mudelis ja kasutada rohkem infotehnoloogilistest lahendustest tulenevaid võimalusi. Tava on võimalik rakendada mitmet moodi, ühed võimalikud neist on järgnevad:
 - „5 MIKS-i“ meetod [6] – meetod on hea praktika eesmärgiga jõuda probleemi tuumani ja aru saada probleemi mõjust, et pakkuda võimalusel parim lahendus. Meetodit praktiseeritakse kõige enam kliendi või tellijaga suhtlemisel;

- **Joonised ja ekraanipildid (prototüüpimine)** – kirja pandu paremaks mõistmiseks ning oma ideede edasi kommunikeerimiseks on antud lähenemine väga hea. Joonised ja pildid annavad protsessist või tehnilisest lahendusest kiire ning selge ülevaate ja jätab analüüsi dokumendi tekstilisele osale vähem mitmiktõlgendamise ruumi;
- **4-silma kontroll** – meetod enda töö valideerimiseks, n.-ö. oma idee tutvustamine inimesele, kes sellega varasemalt pole tutvunud või kokku puutunud, antud juhul mõni teine ärianalüütik näiteks.

3.1.1.3 Kasutusel olevad halvad tavad/praktikad

Halvad tavad, praktikad on pigem otsesed vastandid headele tavadele ning igas projektis võivad need esineda erineval kujul. Tavad ning praktikad mille vastu kõige enam eksitakse on järgmised:

- **Ebapiisav dokumentatsioon** – detailne nõuete dokumenteerimine pole mingi projekti osa raames vajalik ärianalüütiku arvates ning teema üldist loogikat on võimalik taastada tutvudes programmi koodiga. Lahendused on küll olemas kuid puudub igasugune kirjalik alus, mis neid lahendusi põhjendavad või reguleerivad ning keegi ei ole enam teadlik, miks antud lahendus tehti;
- **Ühtse stiili eiramine** – antud praktilal on mitu erinevat esinemisviisi: esimene neist on ühe projekti raames ei kasutada mitme analüütiku poolt ühtset stiili, mis on iseenesest mõistetav ning tuleneb analüütikute kogemustest; teine neist on, et analüütik ei kasuta enda nõuete kirjeldamisel ja dokumenteerimisel ühtset stiili;
- **Teema/projekti olulisus ja tulevik** – pigem rohkem psühholoogiline takistus, kus analüütik ei tunne huvi teema vastu ning ei hooli projekti pikema perspektiivi eesmärkidest ning üldistest poliitikatest – kvantitatiivne analüüsimine.

3.1.2 Dokumenteerimise praktikad, meetodikad

Järgnev peatükk annab põgusa ülevaate kuidas ärianalüüsi RIK-s dokumenteeritakse, kuidas analüüsi osapooltele välja kommunikeeritakse, millised on kasutusel olevad head või halvad meetodikad ja praktikad. Uuritud meetodikaid ja praktikaid arvestan võimaliku lahenduse pakkumisel.

Ärianalüüsi dokumendi sisule on RIK-i arendusosakond kehtestanud nõuded [7], mis on nende jaoks oluline ning mida nad dokumendis näha soovivad. Ärianalüüsi dokumente on RIK-is kasutusel vormiliselt mitu erinevat – pisiarendused vormistatakse Trac-i või JIRA

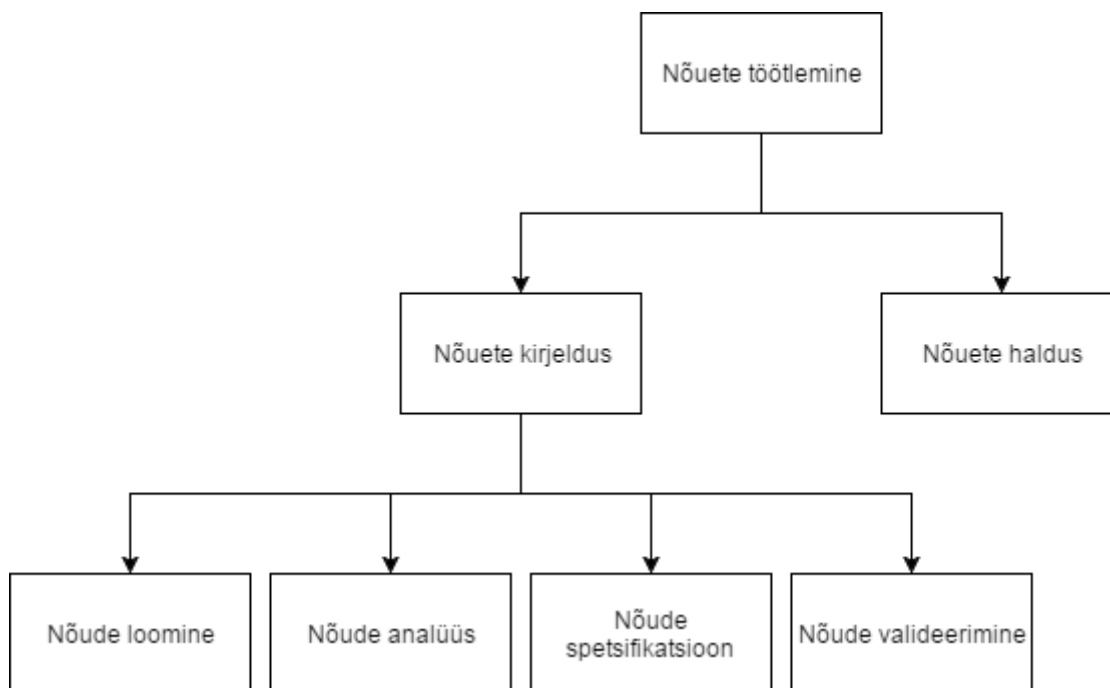
piletitena; majasisese arenduse korral korrektselt vormistatud ning dokumenteeritud ärianalüüsi dokument; hangete puhul hankeobjekti tehniline kirjeldus; taotluste puhul, taotluse vormi sisustamine.

3.1.2.1 Nõuete kirjeldamine ning haldamine

Kõige olulisem ärianalüüsi dokumendi osa on ärinõuded – nõuded mis kirjeldavad arendatavat infosüsteemi ning selle toimimise eesmärki. RIK-is on ärianalüütikud koostöös loonud nõuete kirjeldamise ja haldamise põhimõtete ja reeglite dokumendi [8], mis koosneb soovituslikult kohustuslikest põhimõtetest, mida ärianalüütikud peaksid nõuete kirjeldamisel dokumentatsioonis jälgima.

Nõuete töötlemine jaguneb üldjuhul kaheks – nõuete kirjeldamiseks ning nõuete halduseks. Nõuete kirjeldamine tähendab omakorda nõude loomist, analüüsi, täpsustamist (spetsifikatsiooni näol) ja valideerimist.

Nõuete kirjeldamise ning haldamise erinevad alamosad võtab kokku järgnev Joonis 1:



Joonis 1: Nõuete töötlemise alamosad.

Nõude loomine on võrdväärne skoobi paikapanekuga, ettevalmistusega ja erinevate kliendi poolsete soovide kirjapanekuga (osapooltes on tavaliselt projektijuht, ärianalüütik ja klient või teenuse omanik). Edasine on nõude selgroo loomine nõude analüüsi ning täpse spetsifikatsiooni kirjapaneku näol ärianalüütiku poolt. Valideerimine on 4-silma kontrolli

meetod kliendi/teenuse omanikuga, mille raames annab klient või teenuse omanik nõudele tagasisidet ning võtab nõude vastu ehk kinnitab.

Nõuete haldus on keerulisem dokumenteerimise protsess ning eeldab väga palju suure pildi teadmist ning kõikide nõuetega kursis olemist – eelkõige on see oluline nõuete seoste haldamisel. Soovituslik on kasutada ühe projekti raames nõuete kirjeldusel ühtset kokku lepitud vormindust, et muuta haldamine kordades lihtsamaks. Probleemiks on siinkohal asjaolu kui ühe projekti raames teostavad ärianalüüsi mitmed ärianalüütikud ning igal analüütikul on oma nõuete haldamise praktika. Teiseks probleemiks on kujunenud nõuete haldamisel ja kirjeldamisel kasutatakse palju ainult tekstitöötlusprogramme mis on üleküllustatud erinevate viidetega teistele dokumentidele ning nõuetele ning tihtipeale ei ole võimalik seda kõike pikemas perspektiivis hallata.

Nõuete haldus ning vormistamine hõlmab üldisemalt enda alla järgmist:

- **Nõude baasi määramine** – üle vaadatud ja kokkulepitud nõude põhi suuremate muudatuste järgimiseks. Tihedalt on seotud nõude versioneerimisega, iga suurema muudatuse korral suuredatakse baasi numbrit;
- **Versioonihaldus** – kokkulepitud versioonide kasutamine. Projektiti on see erinev ning võib projekti raames muutuda. Näiteks on nõude kogunumber 100.0.123, kus esimene number vasakult (100) tähistab nõude numbrit, teine number (0) baasi numbrit ja kolmas number (123) konkreetse baasi versiooni numbrit;
- **Mõjuanalüüs** – individuaalse nõude uurimus, teadmaks mis süsteemi osi puudutab nõude muutmine ning kas on veel nõudeid (peamiselt seotud nõuded), mida oleks vaja muuta antud muudatuse raames;
- **Nõude parameetrid** – kindlad nõude osad, mis peavad igal nõudel olema. Projektiti on need kokkulepitud üldiselt, kuid võib esineda erinevusi ühe projekti erinevate analüütikute suhtes;
- **Nõude seoste haldamine** – omavahel seotud nõuete haldamine. Näiteks, kui on nõue X seotud nõudega Y (Y nõue on X alamnõue näiteks), siis nõude Z seostamisel nõudega Y tuleb see ka seostada nõudega X. Mõjuanalüüsi muudab selline lähenemine kordades lihtsamaks, s.t. kas nõude Z muutmisel peaksime muutma ainult Y nõuet või ka X nõuet.

3.1.2.2 Osapoolte vaheline kommunikatsioon

Kommunikatsioonivorme ärianalüüsi dokumendi nõuete valideerimiseks, täpsustamiseks ja kinnitamiseks on RIK-s kasutusel erinevaid. Olulisemad nendest on järgmised:

- **Koosolek** – nõuete tutvustamine ning ärianalüütiku ideede valideerimine inimsuhtluse ning omavahelise arutelu teel. Koosolekute raames võib välja tulla uusi vajadusi ja lahendusi ning samuti on koosolekuid võimalik kasutada oma nõuete ning mõtete suuliseks kinnitamiseks. Koosolekud on olulised kommunikatsioonivahendid pigem prioriteetsete ning käimasolevate teemade suhtes;
- **Kirjavahetus** – kirjavahetustega edastatakse otsuseid ning kinnitusi lisaks suulistele kinnitustele. Samuti tulevad kirjavahetuses välja pikemat arutelu vajavad küsimused ning küsimused mis on suunatud tulevikule, kuid on oluline, et neist hetkel jääks kirjalik jälg parema ülevaate saamiseks.
- **Digitaalne individuaalne suhtlus** – alternatiiv koosolekule, kuid sama mõttega.

3.1.2.3 Kasutatavad tehnoloogilised lahendused dokumenteerimisel

Järgnev peatükk annab ülevaate kasutusel olevatest ärianalüüsi dokumenteerimise lahendustest. Antud lahendusi kasutatakse iga projekti raames erinevalt ning on eelnevalt kokkulepitud projektimeeskonnaga. Tihti esineb olukordi kus dokumenteerimise lahendusi vahetatakse või kasutatakse mitut lahendust paralleelselt korraga.

Trac põhine teadmuspank

Iga nõue vormistatakse Trac-i piletina ning tekib nõuete teadmuspank. Analüüsi dokumendis viidatakse nõuetele linkides ära defineeritud nõude Trac-i pilet.

Trac-i puhul on probleemiks, et seda kasutatakse peamiselt probleemide ja vigade raporteerimiseks arendajatele, mistõttu segunevad nõuete piletid vigade ning probleemide piletitega, kaob ühtne numeratsioon nõuetel ning pikemas perspektiivis kaob järg, millised on nõuetena mõeldud analüüsi piletid ning millised on arendusepiletid ühe projekti raames.

Atlassian Confluence põhine teadmuspank

Confluence on meeskonnatöö terviklahendus ning kommerts viki. Võimaldab luua dokumente, organiseerida dokumendihaldust ning on antud neljast dokumenteerimise lahendustest kõige võimalusterohkem.

Nõuete kirjeldamiseks kasutatakse Confluence-i funktsionaalsust *product requirements* [9]. Igal projektil on eeldefineeritud vormindus nõuetele. Kuna projektide põhiselt on viidud ka dokumentatsioon Confluence-i põhiseks on süsteemi sisene viitamine ning teadmuspanga loomine üpris kerge. Probleemne on nende projektide puhul, mis kasutavad Trac-i põhist lähenemist, kus viidatakse Confluence-i välises dokumentatsioonis Confluence-s loodud nõuetele. Viiteid pole võimalik nii hästi hallata.

Confluence lahendus on teadmuspanga loomisel hea, kuid kõikide funktsionaalsuste ja teenuste suurepäraseks kasutamiseks tuleb ettevõttel teha suuremamahulisi väljaminekuid (olenevalt kasutajate arvust).

Veebisaidi põhine teadmuspank

Asutuse siseselt on mõningate projektide raames loodud teadmuspangana eraldiseisev lihtsama otsinguga veebilehekülj. Iga nõue on kirjeldatud eraldi leheküljena. Dokumentatsiooni koostamisel viidatakse nõude leheküljele.

Antud teadmuspanga puhul on probleemiks otsingu täpsus ning puudulik seoste haldus. Seoste haldus on toimiv vaid juhul kui inimfaktor seoseid pidevalt uuendab, mistõttu on vigade tekkimise oht väga kõrgel tasemel.

Microsoft Word põhine teadmuspank

Antud teadmuspank põhineb dokumentide vahelisel ristviitamisel ning dokumentide viitamisel. Dokumente hoiustatakse ühtses kohas, mis on projektimeeskonnale ligipääsetav. Peamiselt on see kasutuses olnud vanemate pikemate projektide puhul. Tihtipeale kasutatakse seda meetodit koos mõne teise teadmuspanga meetodiga ning pigem üritatakse üle minna uuematele lahendustele, aga on siiski oluline pidevalt Wordi põhine teadmuspank üle kontrollida.

Probleemidena on siin kindlasti kindla ülevaate puudulikkus ning uuendamata andmed. Samuti eksisteerib võimalus puudulike dokumentide suhtes, millele on viidatud või on need dokumendid poolikud.

3.2 Süsteemianalüüs

Süsteemianalüüsi eesmärk on aidata arendajal efektiivselt ja jälgitavalt realiseerida tellitud ja kokkulepitud (mitte)funktsionaalsus olemasolevatel projekti tingimustel olemasolevas

kontekstis nii, et lõppkasutajad, infosüsteem ja võimalikud sidussüsteemid sellest võidaks [10].

3.2.1 Analüüsi teostamise protsess

Järgnev peatükk annab põgusa ülevaate kuidas süsteemianalüüsi protsess RIK-s toimib, kes ja kuidas on kaasatud osapooltena, millised on kasutusel olevad head tavad ja halvad praktikad.

3.2.1.1 Osapooled

Sarnaselt ärianalüüsile on süsteemianalüüsi teostamise protsessi on kaasatud mitmed osapooled, kellest peamised on süsteemianalüütik, ärianalüütik (või ärinõuete koostaja), arendaja, projektijuht. Vajadusel on samuti kaasatud tellijad, testijad, infosüsteemi haldurid (kusjuures võivad haldurid olla ka ärianalüütiku, süsteemianalüütiku ja testija rollis), infrastruktuuride osakonna esindaja, infoturbe talituse esindaja, kasutusmugavuse eksperdid, asutusvälised infosüsteemi puudutava valdkonna eksperdid (nende alla käivad sisulised kui ka tehnilised eksperdid).

Protsessis omavad osapooled ekspertliku arvamuse avaldaja rolli ning samuti kontrolliva vaatleja rolli. Viimase puhul võib näitena välja tuua ärianalüütikut, kes valideerib, et tellija kõikide vajaduste ja eesmärkidega on arvestatud.

3.2.1.2 Süsteemianalüüsi tüübid

Süsteemianalüüsi mõiste alla ei ole koondatud vaid detailse analüüsi teostamine, vaid ka arvamuste, hinnangute ja ülevaadete andmine. RIK-s on kasutusel põhiliselt viis süsteemianalüüsi tüüpi:

- Arvamus
- Ülevaade
- Projektisisene eelanalüüs
- Eelanalüüs
- Detailanalüüs

Arvamus, ülevaade ja projektisisene eelanalüüs on oma olemuselt sarnased ning võivad olla kombinatsioonid erinevatest analüüsi tüüpidest. Nimetatud tüüpide puhul on oluline, et tulemuse soovija annaks oma lõpptulemusest selgelt ja põhjalikult teada. Eelanalüüsi näol on tegemist põhjaliku uurimustulemusega. Eelanalüüs valmib mitme osapoole koostööna ning on hinnanguks mingi projekti realiseerimiseks. Oma vormilt on tegemist ametlikuma

dokumendiga kui arvamus või ülevaade. Detailanalüüs on selgelt eristuv kõikidest analüüsi tüüpidest – nii mahult kui keerukuselt.

Arvamus on süsteemianalüütiku poolt antav põhjendatud hinnang/vastus mis põhineb tema eelnevatel teadmistel ja kogemustel. Arvamus ei tähenda, et olemasoleva infosüsteemi jaoks hakatakse midagi arendama.

Arvamuse osapoolteks on reeglina arvamuse küsija, kes võib olla nii tellija, projektijuht, ärianalüütik, haldur jne, ja arvamuse andja, kelleks on süsteemianalüütik. Süsteemianalüütikul on arvamuse koostamisel alati võimalus kaasata teisi süsteemianalüütikuid või arhitekte, arendajaid.

Ülevaatlik analüüs on süsteemianalüütiku teadmistele ja kogemustele toetuv info. Ülevaade infosüsteemi kohta on veidi pealiskaudsem kui süsteemi dokumentatsioon ning sõltub oluliselt infosüsteemi dokumentatsioonist endast. Ülevaade peab sisaldama kokkuvõtet, mis vastab ülevaate koostamise põhivajadusele.

Ülevaate osapoolteks on reeglina arvamuse küsija, kes võib olla nii tellija, projektijuht, ärianalüütik, haldur jne, ja arvamuse andja, kelleks on süsteemianalüütik. Süsteemianalüütikul on ülevaate koostamisel alati võimalus kaasata teisi süsteemianalüütikuid või arhitekte, arendajaid.

Projektisisene eelanalüüs on suunatud lähteülesande ja skoobivälise planeeritavate arendustele/laiendustele/muudatustele. Tegemist on arendustellimuse pealt uuritud ja põhjendatud täiendus jooksva projekti jaoks.

Antud eelanalüüs peab kindlasti andma edasi infot analüüsi mahu kohta; arenduse mahu ja keerukuse kohta; tellitavast funktsionaalsusest ning prioriteedist.

Projektisiseseks eelanalüüsi osapoolteks on peamiselt projektijuht ning süsteemianalüütik. Vajadusel on kaasatud tehniline arhitekt ning arendaja.

Eelanalüüs on uue infosüsteemi või infosüsteemi suuremahulise uue funktsionaalsuse tähendus lähteülesande/arendustellimuse põhjal. Eelanalüüsi lõppeesmärgiks on välja selgitada projekti kogumaksumus ja ajakulu. Eelanalüüsi põhjal on tellijal võimalik välja selgitada kas ja millal seda projekti tegema hakatakse.

Eelanalüüs annab ülevaate analüüsi ja arenduse mahu kohta; projekti potentsiaalse maksumuse kohta; raud- ja tarkvaralise ressursi kirjeldusest; soovitusliku arendustiimi ja haldustiimi kirjeldusest; tellitavate funktsionaalsuste kirjeldustest.

Eelanalüüs oma olemuselt sarnaneb lähteülesandele. Seetõttu loetakse eelanalüüsi läbiviimist mingil määral väikeseks projektiks, millel on vastutav projektijuht.

Detailanalüüs on süsteemianalüütiku poolt põhjalikult uuritud ärinõuetest ja äriprotsessidest kinni pidav loodava või täiendatava infosüsteemi funktsionaalsuse süstemaatiline, struktureeritud, jälgitav ja arendajale selgelt ja detailselt esitatud täpne kirjeldus, mida arendaja saab töösse võtta [10].

Detailanalüüsi iseloomulikud sisuomadused on järgmised:

- Ärianalüütiku poolt sõnastatud ärinõuete lahtimõtestamine ja struktureerimine;
- Süsteemsete nõuete leidmine ja ilmutamine;
- Ärinõuete ja süsteemsete nõuete kinnitamine ärianalüütikuga/tellijaga;
- Infosüsteemi esialgse suhteid näitava mudeli koostamine;
- Äriprotsesside lahtimõtestamine ning uurimine;
- Andmemudeli koostamine, täpsustamine ja täiendamine kogu projekti jooksul;
- Kasutajaliidese prototüübi realiseerimine;
- Infosüsteemi arhitektuuri väljatöötamine;
- Funktsionaalsuse selgitamine ja täpsustamine;
- Kogu projekti analüüsi seisu ja sisu tundmine;
- Korduv-analüüsi teostamine – olemasolevate teenuste/kasutuslugude ülevaatamine ja kaasajastamine.

Detailanalüüsi raames valmivad järgmised komponendid iga projekti raames:

- Loodava süsteemi keskne kirjeldus ja detailne spetsifikatsioon;
- Konseptuaalsed andmemudelid, mille iga komponent on detailselt kirjeldatud ning nõuetega kaetud;
- Andmemudelid, mille iga komponent on detailselt kirjeldatud ning nõuetega kaetud;
- Üldiste kasutuslugude/protsesside/stsenaariumite kirjeldused äri- ja süsteemsete nõuete alusel;
- Süsteemi protsessimudelid ja nende selgitavad kirjeldused;

- Liidestuste kirjeldused;
- Kasutajaliidese prototüüp, kus
 - kõik vormid ja vormiosad on katud nõuetega
 - selgelt näidatud liikumine erinevate kasutajaliidese vormide/osade vahel;
- Infosüsteemi juurdepääsuõiguste kirjeldused ja nõuded;
- Infosüsteemi logimise funktsionaalsuse kirjeldused ja nõuded;
- Testimise stsenaariumid.

3.2.1.3 Kasutusel olevad head tavad/praktikad

Sarnaselt ärianalüütikutele on loodud ka süsteemianalüütikutele nimekiri headest tavadest [11], mida soovituslikult jälgida. Olulisemad nendest ning tihedamalt kasutuses olevad nendest on järgmised:

- **Mitu lahendust** – kõike on võimalik lahendada mitut moodi, sama kehtib ka infotehnoloogiliste lahenduste puhul. Üks headest tavadest on pakkuda mitu läbimõeldud lahendust ning mõju infosüsteemile, mille vahel tellijal on võimalik valida;
- **Huvi ümbritseva vastu** – siia alla käivad nii äripool, programmi kood, baasi haldus, projekti üldseis, mida tulevik toob jne;
- **Piltide ja skeemide kasutamine** – sageli on lihtsam mõista visuaalset pilti kui pikka rida teksti;
- **4-silma kontroll** – meetod enda töö valideerimiseks, n.-ö. oma idee tutvustamine inimesele, kes sellega varasemalt pole tutvunud või kokku puutunud (antud juhul mõni teine süsteemianalüütik näiteks).

3.2.1.4 Kasutusel olevad halvad tavad/praktikad

Halvad tavad, praktikad on pigem otsesed vastandid headele tavadele ning igas projektis võivad need esineda erineval kujul. Tavad ning praktikad mille vastu kõige enam eksitakse on järgmised:

- **Spetsifikatsiooni ühtsus** – spetsifikatsioon pole ühtne: esineb mitmel erineval kujul; kõigil on erinev ülesehitus ja loogika; dubleeritud mitmesse erinevasse kohta;
- **Pimesi lähenemine** – antud praktilal on mitu erinevat väljendumise viisi:
 - „mina olen kõige targem“ printsiip;
 - teostuse saavutan üks-ühele nii nagu paluti, ise ei mõtle (huvi puudumine on näiteks üks selle alus);

- projektijuht on ülim, vastutustundetu lähenemine mille raames lasub kogu otsustusvõime ja vastutus projektijuhil;
- **Ainult tekst** – 500 leheküljeline detailanalüüsi dokument, mis koosneb peamiselt mitmeti mõistetavatest nõuetest ja kirjeldustest ning piltlikud ettekujutused on jäänud süsteemianalüütiku pähe.

3.2.2 Dokumenteerimise praktikad, meetodikad

Järgnev peatükk annab põgusa ülevaate kuidas süsteemianalüüsi RIK-s dokumenteeritakse, kuidas neid osapooltele välja kommuniqueeritakse, millised on kasutusel olevad head või halvad meetodikad ja praktikad. Uuritud meetodikaid ja praktikaid arvestan võimaliku lahenduse pakkumisel.

Sarnaselt ärianalüüsile on samuti süsteemianalüüsi nõuetele kehtestatud miinimumnõuded [12] vormistusele, sõnastusele kui ka sisule endale. On oluline, et iga teenuse, kasutusloo ja keskse kirjelduse sisuks on ärinõue ning sellele vastav detailne funktsionaalne nõue/süsteemne nõue/reegel.

3.2.2.1 Nõuete/reeglite kirjeldamine

Lähtuvalt miinimumnõuetest peab iga nõude või reegli kirjelduses olema välja toodud järgnev:

- **Nõude tekkimise aeg** – aeg millal nõude loodi esimest korda;
- **Viimase muutmise aeg** – oluline on hoida nõuded ajakohased ning antud väli muudab samuti versioneerimise lihtsamaks;
- **Seis** – nõude hetkestaatus. Kohati nimetab iga süsteemianalüütik enda jaoks staatuseid erinevalt, kuid on oluline, et kaetud oleksid vähemalt elementaarsed pakutud/ kinnitatud/ töös/ tühistatud;
- **Identifikaator** – nõude unikaalne identifikaator mis määrab ära nõude grupi ning nimetuse. Hilisemad viited teostatakse identifikaatorile;
- **Autor** – nõude esialgne autor. Peamiselt informatiivne eesmärgiga kelle poole vajadusel pöörduda, et mõista nõude algset vajalikkust;
- **Muutja** – kes viimasena nõuet on muutnud ning mis põhjusel on ta seda teinud. Muutmisel ei muutu muutja autoriks vaid muutub nõude vastutavaks isikuks;
- **Kinnitaja** – siin rollis on tellija või ärianalüütik, kes kinnitab nõude ning selle võib edasi anda arendusele.

Nõuete ning reeglite sisust peab kindlasti selguma andmeväljade kasutus (nii sisuliste kui ka tehniliste andmeväljade) ja andmete liikumine. Kõiki nõudeid ja reegleid ja nende sisu toetavad vajadusel pildid, diagrammid, tabelid või loendid. Kogu kirjelduste tulemusel tekib nõuetele detailne kirjeldus ehk detailanalüüs.

Teenuste või kasutuslugude kirjeldus ei pea olema üks kõikehõlmav dokument (sõltub kuidas grupeeritakse). Vajaduspõhiselt võib teha iga teenuse/kasutusloo raames eraldi dokumendi, kuid on oluline, et säilib ühtne jälgitav struktuur.

3.2.2.2 Kasutatavad tehnoloogilised lahendused dokumenteerimisel

Süsteemianalüüsi nõuded ja reeglid baseeruvad peamiselt ärinõuete teadmuspangadel. Kuna ärianalüüsi juures keskendusin pigem teadmuspangade võimalikele kavandamistele, siis siin peatükis on fookus detailanalüüsi vormistamisel.

Detailanalüüsi loomisel on kasutusel kõige enam kolm erinevat lahendust ning oleneb suuresti projekti olemusest ning sisust, samuti süsteemianalüütikute harjumustest. Tihti on kirjeldatud lahendused ka omavahel seotud ning neid kasutatakse paralleelselt.

Tegemist on lõpptulemust dokumenteerivate lahendustega. Kõigele lisanduvad erinevad protsesside modelleerimist ja andmemudelite loomist jms toetavad tarkvarad, mille puhul on igal süsteemianalüütikul oma kompetents.

Atlassian Confluence

Sarnaselt ärianalüüsi dokumentatsioonile on üksikute projektide raames ka süsteemianalüütikute poolt kasutusele võetud Atlassian Confluence poolt pakutavad lahendused.

Kõige positiivsem Confluence lahenduse juures süsteemianalüütikule on erinevate nõuete viitamine projekti raames ning üldisem vormistamine – nõudeid on võimalik vormistada kindla malli alusel. Samuti ligipääsetavus ning õigustepõhisus.

Olulisem puudus Confluence lahenduse juures on puudulik hea funktsionaalsus erinevate liidestuste analüüsi kirjutamiseks, kuna pole kergelt võimalik viidata teise projekti nõuetele, veelenam kui teine projekt ei kasuta ise Confluence lahendust. Teiseks puuduseks on võimalus eristada nõuet reeglist jms. Nõuete numbrite formuleerimisel on võimalik neid küll eristada, kuid objekti tasandil on nad Confluence lahenduse puhul samal tasemel, mis pole õige.

Microsoft Word

Kõige laialdasemalt kasutatav lahendus dokumenteerimisel ning mille jaoks pole vaja erilisi oskusi või teadmisi. Äärmisel juhul peaksid olema oskused elementaarse vormistamise osas.

Olulised puudujäägid on reeglite ning teiste analüüsi dokumendite viitamisel. Peamised probleemid piirduvad vormistuse ebaühtluse tasandil – vormistamise metoodika määrab iga analüütik endale ise ning selle tulemusel on tekkivad dokumendid suuresti sisult erinevad ning projektiti raskesti loetavad (eeldavad erinevat lugemisstiili ning lähenemist), muutes mitme analüütiku omavahelise kommunikatsiooni keerulisemaks. Nõuete või reeglite haldamisel tekib samuti probleeme olenevalt kas andmeid on dokumentides dubleeritud või mitte – peab ühe nõude muudatust „jälitama“ kogu projekti dokumentide raames.

Prototüüp

Prototüüpimine annab väga hea piltliku ülevaate infosüsteemi funktsionaalsusest ning toimivast loogikast kui ka elementide ja vormide paigutusest ning disainist. Prototüüpimise korral kõrvutatakse juurde dokumenti, mis kirjeldab tehniliselt nõudeid, infosüsteemi arhitektuuri ning muid detailanalüüsi käigus kirjeldamist vajavaid komponente sh. andmeväljade kasutus ning andmete liikumine erinevate komponentide vahel.

Prototüübid on peamiselt miniatuurse funktsionaalsusega rakendust jälgendavad veebileheküljed ning mitte lihtsamad pildid vormidest või infosüsteemist.

Prototüüpimise üks headest omadustest on nõuete haldus (prototüüpimise korral kirjutatakse nõuded ka prototüüpi), kuid palju oleneb haldamine prototüüpija oskustest.

Peamised ohukohad prototüüpimisel on analüütiku *front-end* arendamise oskus (HTML, CSS, Javascript). Prototüüp on ainult niivõrd hea kui heaks on analüütik selle suuteline kirjutama. Teine ohtlik omadus on piirduda vaid prototüübiga ning ilma dokumenti kõrvale loomata. Tahes tahtmata kaob ära osa informatsiooni infosüsteemi arhitektuurist ning selle omadustest, jättes programmeerijatele kordades rohkem vastutust ning fantaasia ruumi infosüsteemi arendamisel.

4 Alternatiivsed lahendused

Võtsin alternatiivsete lahenduste valimisse kolm olemuselt erinevat tarkvara, millest kõik puudutavad analüüsimise protsessi erinevaid aspekte. Confluence – kiidetud ning osaliselt RIK-is kasutusel olev dokumenteerimise tarkvara. Tegemist on samuti kõige populaarseima dokumenteerimise süsteemiga; Axure RP – UX disainerite poolt kiidetud prototüüpimise abivahend. Tegemist on funktsionaalsuselt kõige ulatuslikuma prototüüpimise vahendiga; Targetprocess – agiilset metoodikat pooldav projekti haldamise tarkvara, mis võtab fookusesse andmete visualiseerimise.

Lahenduste häid ning halbu omadusi uuritakse nii äri- kui süsteemianalüütiku vaatest ning samuti dokumenteerimiste võimalustest ning selle lihtsusest. Omadused on valitud lähtudes hetkeseisu kirjeldusest tulenevatest vajadustest, mis on kirjeldatud antud töö peatükis 0.

4.1 Atlassian Confluence ülevaade

Confluence [13] on 2004. aastal Atlassian-i poolt välja antud meeskondlikule koostööle suunatud tarkvara. Oma olemuselt on Confluence viki mis võimaldab luua mahukaid teadmuspanku ja dokumentatsiooni ning neid jagada oma meeskonnale.

The screenshot shows the Atlassian Confluence interface. The main page is titled "Large Scale Reserv" and is created and last modified by Jerry 9 months ago. It features a metadata table, a "Goal" section, and a "User Stories" table. A "JIRA links" popup menu is open, showing a list of JIRA issues and sprints.

Target release	SeeSpace
Document status	COMPLETED
Document owner	@Ryan Le
Developers	@Kevin C
QA	@Emmet
Epic	TIS-59

Goal
Enable larger groups to use the S...

User Stories

#	User Story Title	Priority
1	Multiple travel providers in one reservation TIS-60 (Open)	1: Must Have
2	Multiple hotels in one reservation TIS-61 (Open)	1: Must Have

JIRA links

- Sprints
 - Sprint 5
10 days remaining
- Epics
 - Large Scale Deployments
Large Scale Deployments
- Issues
 - TIS-63 (Open)
Customer Discount Model
 - TIS-62 (Open)
Multiple flights on one reservation
 - TIS-61 (Open)
Multiple hotels in one reservation
 - TIS-60 (Open)
Multiple travel providers in one reservation

Joonis 2: Atlassian Confluence haldusliides.

4.1.1 Teadmuspank

Confluence võimaldab luua mitmekülgset teadmuspanka, mida meeskond saab lihtsalt organiseerida ning ligi pääseda ühest kohast. Kuigi Confluence-i lahenduse põhiselt on teadmuspank loodud projekti ning meeskonna jaoks, on seda võimalik mugandada ärianalüüsi teadmuspangaks. Teadmuspankasid on Confluence-s kahesuguseid – projekti põhised, mis kujutatakse rohkem projekti esilehena ning sinna pannakse kirja kõik projektiga seonduv; projekti nõuete põhine, mis on nimekiri projekti või funktsionaalsuse (olenevalt dokumentatsiooni hierarhiast) nõuetest. Viimane on analüütikutele rohkem ülevaatlikum.

Ülevaatlikuse puhul on Confluence hea lahendus, kuid viki lehtedel ehitatud lahendus eeldab iga lehe muutmist eraldi ning on keeruline hoida seoseid erinevate lehtede vahel kontrolli all – puudub samaaegne üldine ülevaade.

4.1.2 Dokumentatsioon ja dokumenteerimine

Confluence dokumentatsioon põhineb viki lehtedel mida on võimalik vastavalt vajadustele hierarhiliselt või lehtede mallide põhiselt üles seada. Nõuete ning dokumentatsiooni loomine on tehtud võimaluste rohkelt ning kasutajale loogiliselt arusaadavana. Samas on dokumentatsiooni haldamine kasutajasõbralikkuse vaene. Dokumentatsioon on lehe põhiselt versioneeritud ning igat lehte on võimalik kõigil kasutajatel samaaegselt muuta. Kõiki lehti on võimalik sisu põhiselt kommenteerida, mis on tagasisidele ning koostööle väga hea omadus.

Dokumentatsiooni juures säästab Confluence aega mallide genereerimisel ning aitab kaasa haldamisele, kuid enamasti tuleb dokumentatsiooni näiteks staatuseid või malle hallata käsitsi ning kõik peab olema projektipõhiselt kokkuleppeline.

4.1.3 Välised dokumendid

Confluence pakub dokumentide (.docx; .pdf jne) üleslaadimisvõimalust, nende kiiret läbivaatust ning sisuotsingut. Lahendus on sarnane dokumendihalduse süsteemidele ning on kasulik kui tarkvara omab varasemat dokumentatsiooni ja projekti raames minnakse üle Confluence lahendusele.

Confluence omalt poolt ei paku eraldi seisvate dokumentide (viki leht/nõuded -> PDF/Word dokument) loomise võimalust. Selle tulemusena pole võimalik kergelt luua eraldiseisvaid spetsifikatsioone või kasutajajuhendeid, manuaale.

4.1.4 Maksumus

Confluence pakub nelja erinevat maksumuse meetodit – väikestele meeskondadele, kasvavatele meeskondadele (pilvepõhise majutusega), ise majutuvat serverit või andmeladu [14].

Väikesed meeskonnad – \$10 kuus, andes Confluence lahendusele ligipääsu kuni kümnele kasutajale.

Kasvavad meeskonnad – sõltuvalt kasutajate arvust on erinevad hinnad (kuupõhine tasumine):

Tabel 1: Confluence maksumuse sõltuvus kasutajate arvust (kasvav meeskond).

Kasutajate arv	Maksumus (<i>kuus</i>)
15	\$50
25	\$100
50	\$200
100	\$300
500	\$500
2000	\$2000

Server – eraldiseisev server on võimalik soetada ühekordse maksumusega ning sõltub kasutajate arvust:

Tabel 2: Confluence maksumuse sõltuvus kasutajate arvust (server).

Kasutajate arv	Maksumus (<i>ühekordne</i>)
10	\$10
25	\$1400
50	\$2500
100	\$4600
250	\$9000
500	\$14000
2000	\$18000
10000	\$23000
10000+	\$28000

Andmeladu – iga aastane maksumus ning sõltub kasutajate arvust:

Tabel 3: Confluence maksumuse sõltuvus kasutajate arvust (andmeladu).

Kasutajate arv	Maksumus (ühekordne)
500	\$12000
1000	\$24000
2000	\$48000
3000	\$72000
4000	\$96000
5000	\$120000
10000	\$130000
15000	\$140000
20000	\$150000
25000	\$160000
30000	\$170000
35000	\$180000
40000	\$190000
40000+	\$200000

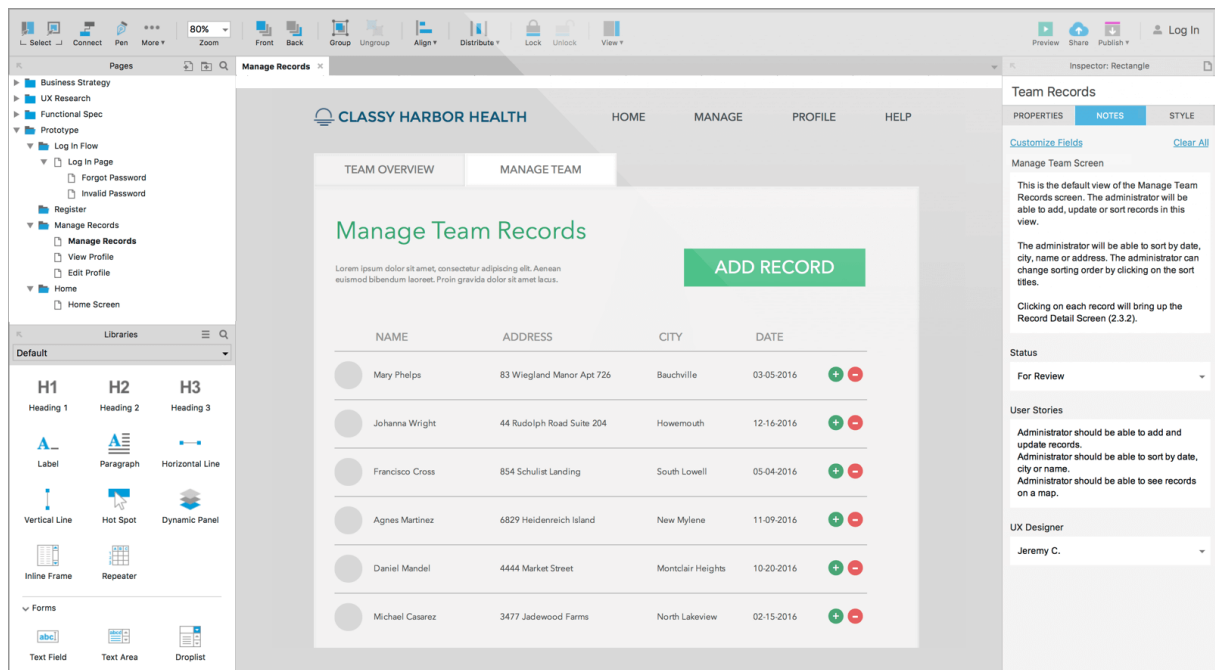
Confluence mõistes on kasutaja isik, kellele on loodud eraldiseisev nimeline konto.

Registrite ja Infosüsteemide Keskuse seisukohalt lähtudes oleks kõige optimaalsem lahendus 500 kasutajale mõeldud kasvava meeskonna lahendus. Sellega oleksid kaetud olemasolevad võimalike kasutajate arv ning lahendus on ka tuleviku mõistes jätkusuutlikum.

Oluline on meeles pidada, et Atlassian lahenduste puhul on mõistlik võtta Confluence'iga koos teised Atlassiani poolt pakutavad tooted [15], et saavutada parim tulemus.

4.2 Axure RP ülevaade

Axure RP [16] on suurepärase prototüüpimise abivahend mis on suunatud veebi- ja töölaua rakenduste disainimiseks. Lisaks prototüüpimisele võimaldab Axure dokumenteerida kõiki prototüübi osasid ning luua igale osale spetsifikatsioon.



Joonis 3: Axure RP haldusliides.

4.2.1 Prototüüpimine

Axure RP võimaldab disainida kasutajaliideseid ning liidese käitumismustreid (näiteks veebi elementide peitmine või näitamine kindla tegevuse peale – nupu või lingi vajutamine). Prototüüpimist toetavad mitmed vidinad (*widgets*) sõrestiku ja/või vormide loomiseks. Prototüüpimisel loodud vorme ja vidinaid saab taaskasutada või luua eraldiseisvaid alusmalle.

Lihtsamate vormide prototüüpimine ei eelda *front-end* kogemust või programmeerimise oskust. Prototüüpide loomine toimib lihtsal kasutajasõbralikul *drag-and-drop* meetodil. Keerulistema funktsionaalsete prototüüpide loomiseks on vaja vähemalt elementaarsemaid veebirakenduste loomise kogemusi ning arusaama terminitest.

Prototüüpe luues, tekib Axure kasutajal harjumus mõelda lahendust välja lähtudes sellest, kuidas neid Axure programmis on võimalik luua, mitte kuidas lõpp-kasutajale seda saaks kõige paremini esitleda.

4.2.2 Dokumentatsioon ja dokumenteerimine

Axure RP's on võimalik kõiki prototüübi osi (elemente, vidinaid) eraldiseisvalt dokumenteerida. Dokumenteerimine on jaotatud märkusteks ja tehniliseks spetsifikatsiooniks (töövoo või elemendi kirjeldus). Elemente taaskasutades dubleeritakse kogu elemendi dokumentatsioon.

Tegemist on otseselt prototüübi komponentide tehniliste nõuetega ning dokumentatsiooniga, kuid vajadusel on võimalik kokkuleppeliselt neid laiendada infosüsteemi äri- ja funktsionaalseteks nõueteks. Prototüüp-enne dokumenteerimise meetod.

4.2.3 Välised dokumendid

Erinevalt Confluence lahendusest pakub Axure RP dokumentide loomise võimalust. Kirjeldatud tehnilist spetsifikatsiooni on võimalik väljastada Word dokumentidena või HTML lehtedena.

Axure RP ei võimalda üles laadida dokumente (kui need ei ole eelnevalt vastavas keskkonnas loodud).

4.2.4 Maksumus

Axure RP pakub kolme liiki maksumuse plaani – pro, meeskondadele, ettevõttele. Võimalus on rakendust rentida kuu kaupa või välja osta [17].

The image shows three pricing plans for Axure RP: PRO, TEAM, and ENTERPRISE. Each plan includes a description, pricing options (monthly subscription and one-time purchase), and a list of included features.

Plan	Target Audience	Subscription Price	One-time Price	Includes
PRO	For professionals creating prototypes and specifications	\$29 per user per month	\$495 per user	• Sketch & Wireframe • Flow Diagrams • Advanced Prototyping • Mobile Prototyping • Notes and Documentation • Axure Share Publishing
TEAM	For teams co-authoring prototypes and specifications	\$49 per user per month	\$895 per user	Everything in PRO + Co-authoring (Team Projects) + Revision History + Team Project Hosting on Axure Share
ENTERPRISE	For companies requiring an on-premises solution for publishing and co-authoring prototypes and specifications (not available as a perpetual license)	\$99 per user per month	-	Everything in TEAM plus Axure Share On-Premises + Publish to your own servers + Host Team Projects on-premises + SAML-based single sign-on + Administrator accounts + Author and Reviewer accounts + Manage Workspace access

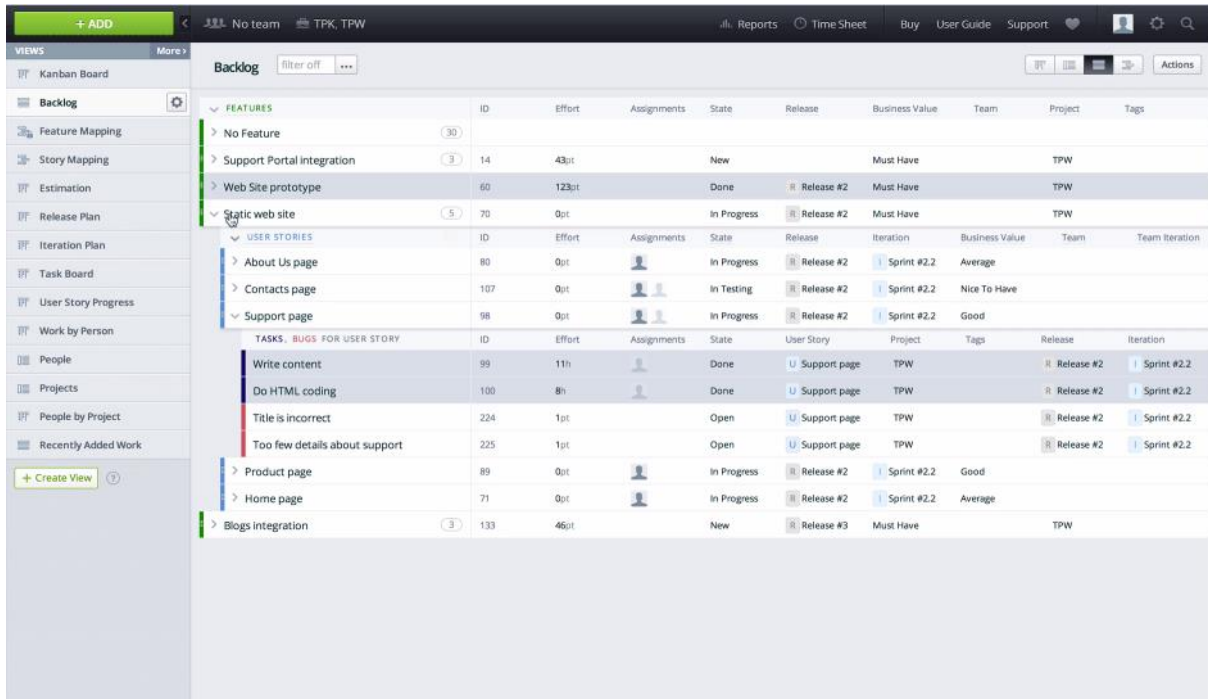
Joonis 4: Axure RP maksumuse plaanid.

Kasutajate all mõistab Axure RP kahte tüüpi kasutajaid – individuaalseid või ettevõtte kasutajaid. Ettevõtte kasutajaid on võimalus pidevalt hallata – ühe kasutaja eemaldamisel on võimalik ta asendada teisega.

Registrite ja Infosüsteemide Keskuse seisukohalt lähtudes oleks kõige optimaalsem lahendus ettevõttele (*Enterprise*) suunatud lahendus. Kasutajad pole piiratud ning on võimalus litsentse juurde hankida vastavalt vajadusele.

4.3 Targetprocess ülevaade

Targetprocess [18] on visuaalne veebipõhine projekti juhtimise ja haldamise tarkvara mis keskendub agiilsele tarkvaraarendus metoodikale. Eelkõige toetab *Scrum*, *Kanban* ja *SAFe* meetodeid. Tarkvara on võimalik mugandada vastavalt projekti meeskonna lähenemisele tarkvara arendusele.



VIEW	Backlog	filter off	...	Actions				
FEATURES								
No Feature	(30)							
Support Portal Integration	(3)	14	43:00	New	Must Have	TPW		
Web Site prototype		60	123:00	Done	Release #2	Must Have	TPW	
Static web site	(5)	70	0:00	In Progress	Release #2	Must Have	TPW	
USER STORIES								
About Us page		80	0:00	In Progress	Release #2	Sprint #2.2	Average	
Contacts page		107	0:00	In Testing	Release #2	Sprint #2.2	Nice To Have	
Support page		98	0:00	In Progress	Release #2	Sprint #2.2	Good	
TASKS, BUGS FOR USER STORY								
Write content		99	11:00	Done	Support page	TPW	Release #2	Sprint #2.2
Do HTML coding		100	8:00	Done	Support page	TPW	Release #2	Sprint #2.2
Title is incorrect		224	1:00	Open	Support page	TPW	Release #2	Sprint #2.2
Too few details about support		225	1:00	Open	Support page	TPW	Release #2	Sprint #2.2
Product page		89	0:00	In Progress	Release #2	Sprint #2.2	Good	
Home page		71	0:00	In Progress	Release #2	Sprint #2.2	Average	
Blogs integration	(3)	133	46:00	New	Release #3	Must Have	TPW	

Joonis 5: Targetprocess haldusliides.

4.3.1 Teadmuspank

Targetprocess on võimalik toetama panna piletipõhist teadmuspanka, mida on lihtne hallata tulenevalt Scrum ja Kanban projekti haldamise lahendusest, mida antud rakendus toetab. Teadmuspankade vaateid on võimalik luua igal kasutajal vastavalt vajadusele, sealhulgas prioritseerida ning kategoriseerida oma tööd.

Tulenevalt vajaduspõhistest vaadetest on kellegi teise tööd üle võtta keeruline ning tuleb vajadusel vaated sobivaks ümber korrigeerida, mis on ajakulukas. Erinevate vaadete loomisega kaob tervikliku pildi ülevaade.

4.3.2 Dokumentatsioon ja dokumenteerimine

Dokumentatsiooni luuakse kategooriate (*features*) kaupa kasutuslugudena. Kasutuslugusid on võimalik prioritseerida ning planeerida vastavalt arendusülesannetena (*task*). Kasutuslugusid on võimalik omavahel siduda „märksõnaskaemi“ stiilis, mis kaotab ära vajaduse teostada seoste analüüsi (*impact analysis*).

4.3.3 Välised dokumendid

Sarnaselt Confluence'le on Targetprocess'is kõikidele piletitele võimalik juurde lisada Word'i dokumente või muid faile (pilte, prototüüpe, mudeleid). Küll aga, erinevalt Confluence lahendusest ei võimalda Targetprocess sisuotsinguid.

4.3.4 Maksumus

Targetprocess pakub samuti kolme liiki maksumuse plaani, millest üks on tasuta, kuid piirangutega – meeskonna, firma ja ettevõtte oma [19].

The image shows a pricing comparison chart for Targetprocess. The title is "Plans and pricing". There are three columns representing different plans: Team, Company, and Enterprise. Each plan lists its features and pricing.

Plan	Features	Pricing
Team	1000 entities, Basic support, Hosted in the cloud	\$0 per user / month Unlimited users
Company	Unlimited entities, Standard support, Hosted in the cloud, Single sign-on	\$20 per user / month with annual billing or \$25 per user / month
Enterprise	Premium support, Private cloud or On-Premise, Product training, Custom development	Enterprise customers may purchase the following additional features:

Joonis 6: Targetprocess maksumuse plaanid.

Kasutajate all mõistab Targetprocess ainult individuaalseid kasutajaid.

Registrite ja Infosüsteemide Keskuse seisukohalt lähtudes oleks kõige optimaalsem lahendus firma lahendus. Seda juhul kui soovitakse mitme projekti põhisk lahendust luua. Ainult väiksemate projektide raames võib kasutada tasuta lahendust.

4.4 Üldised probleemid

Läbivaks probleemiks kõikide eelnevalt kirjeldatud alternatiivsete lahenduste puhul on see, et need on terviklahendused. Terviklahenduste sihtgruppiks on alati meeskond. See tähendab, et

kogu lahendus peab toetama korraga tervet meeskonda, mitte meeskonna kindlat osa (näiteks arendajat, testijat või analüütikut).

Terviklahenduste puhul hakkavad meeskonna erinevad harud sama lahendust mugandama enda valdkonna jaoks. See loob olukorra kus visuaalselt sama lahendust hakkatakse kasutama erinevatel eesmärkidel. Veel enam võib mugandamise protsess erineda projektiti mis muudab meeskonna liikme roteerumise protsessi projektide vahel ajamahukaks ning konfliktseks erinevate arvamuste tõttu.

5 Võimalikud analüüsi metoodikad

Süsteemi analüüs ja disain (SAD) on termin kirjeldamaks metoodikaid arendamaks infosüsteeme mis ühildavad endas infotehnoloogia, inimesed ja andmed toetamaks ärilisi vajadusi [20]. SAD viitab protsessile mis uurib ärilist situatsiooni eesmärgiga seda parandada erinevate meetoditega või protseduuridega.

Pole olemas ühtset, lihtsat tehnikat või metoodikat, mida arendusmeeskonnal on võimalus kasutada, et arendada suurepärase tarkvara. Metoodikad on loodud selleks, et muuta infosüsteemide arendamine lihtsamaks [21].

Valisin erinevate arenduse ning analüüsi metoodikate vahelt välja uuenduslikumad lähenemise meetodid, mis täna on valitud ettevõttes kasutusel rohkem ning mida pigem soovitatakse kasutada. Antud metoodikaid kasutan alternatiivsete lahenduste kui ka enda lahenduse valideerimiseks ning hindamiseks mõõdikutena.

5.1 Agiilne modelleerimine ja dokumenteerimine

Agiilne modelleerimine on praktilal põhinev metoodika tarkvara efektiivseks modelleerimiseks ja dokumenteerimiseks [22]. Tegemist on printsiipide ja praktikate kogumiga, mida kasutatakse eelkõige agiilsetes tarkvaraarendustes. Metoodika on väga paindlik kiirelt arenevate või muudatuste rohkete süsteemide puhul.

5.1.1 Modelleerimise praktikad

Piisavalt head lahendused (*Just barely good enough (JBGE) artifacts*) – võimalikult lihtsad lahendused, lähtudes hetke-olukorrast.

Süsteemi arhitektuurile ja mudelitele keskenduv – mudelite loomise käigus selgub, mida on vaja infosüsteemi puhul, ning mis on sellest kõige olulisem. Tegemist on esmajärgse tarkvara planeerimise meetodiga, mille raames kaardistatakse ka tarkvara riske ja nõrkuseid. Toote omanikku kaasatakse sellesse protsessi kohe alguses.

Teiste kaasamine protsessi – agiilne modelleerimine ja dokumentatsioon eeldab pidevat kommunikatsiooni toote omanikuga ning meeskonnakaaslastega. Pidev tagasisidestamine ning koostöö on antud mõõdiku märksõnadeks.

Nõuded ja prioritseerimine – nõuded mõeldakse ajas ette, et panna paika projekti esialgne skoop ning luuakse esmane nõuete prioriteetide nimekiri. Nõudeid prioritseeritakse pidevalt,

et luua rohkem väärtust kui terviklahendust. Nõudeid kirjutatakse pigem testidena kui korrektselt vormistatud nõuetena.

5.1.2 Dokumenteerimise praktikad

Pidev dokumenteerimine – kogu dokumentatsioon luuakse terve projekti elutsükli vältel, paralleelselt lahenduse loomisega. Dokumenteeritakse kõike.

Hiline dokumenteerimine – kogu dokumentatsioon luuakse võimalikult hiliselt projekti lõpus (vastandina pidevale dokumenteerimisele). Eesmärk on vältida võimalikke „häid“ ideid, mis muudavad infosüsteemi lahendust ja struktuuri.

Ühtne dokumentatsiooni hoiustamine – dokumentatsioon on hoiustatud ühes, kõigile kättesaadavas kohas. Dokumentatsioon peab olema alati ajakohane.

5.2 Kerge dokumentatsioon (*Lightweight Documentation*)

Lightweight Documentation on pigem idee kui meetod ning käsitleb küsimust kuidas muuta traditsioonilist nõuete dokumenteerimist kergeks (*lightweight*) protsessiks [23]. Samas peavad nõuded olema lõppkasutajatele arusaadaval kujul ning nõudeid peab olema võimalik korrektselt analüüsida ja hinnata. Idee keskendub peamiselt kolmele dokumenteerimise variatsioonile – nõuded; välised dokumendid; tarkvara objektid (*artifacts*).

5.2.1 Mõõdikud

Spetsifikatsioon – nõuete dokumentatsioon on kestav protsess mille täpsustused ning detailid on võimalik üle vaadata ning hinnata vastavalt vajadusele igal ajahektel.

Taasesitus – nõuete esitamine prototüüpide või töötava tarkvara näol, muutes nõuded arusaadavamaks.

Kokkulepped ja tagasiside – õigeaegne ning mitmekülgne tagasiside toetab tarkvara evolutsiooni kasutajate individuaalsest vaatest ja loob aluse ühistele otsustele.

5.3 *Lean* printsiibid

Lean keskendub efektiivsuse optimeerimisele ning ebaolulise vähendamisele erinevate printsiipide läbi [24]. *Lean* tarkvaraarendus meetod viitab, et tarkvara peamised kvaliteedi probleemid põhinevad motivatsioonist ning ekspertide poolsetest hinnangutest ning seab eesmärgiks meeskonna motiveerimise pideva tagasisidestamise näol ja anda kindlat hinnangut meeskonnaliikme panusele [25].

5.3.1 Mõõdikud

Ebaolulise vähendamine – protsessid, dokumentatsioon, ülesanded, vead, poolikud tööd, lisaväärtused – *lean* keskendub kõige ebaolulise vähendamisele mis ei lisa otsest väärtust kliendi poolt tellitud tarkvarale.

Õppimiskõver – dokumentatsiooni või detailide planeerimise asemel on mõistlik keskenduda erinevate lahenduste loomisele ning mitmete ideede katsetamistele.

Terviku nägemine – tarkvara arendamisel liigendatakse enamasti suuremad funktsionaalsused väiksemateks ülesanneteks. Sellisel kujul lähenemine tähendab omakorda, et on äärmiselt oluline teada ja jälgida tervikut, vastasel korral muutuvad väiksemad ülesanded või funktsionaalsused omaette tarkvaraks.

6 Alternatiivsete lahenduste hinnang

Antud peatükis hindan alternatiivsete lahenduste vastavust valitud mõõdikutele, mis on kirjeldatud eelmises peatükis. Hinnangute saamiseks on kasutatud musta kasti testimise (*black-box testing*) meetodit [26] mis hõlmab endas uurimuslikku ja spetsifikatsioonidel põhinevat testimist.

Vastavuse hinnangud on antud viiepallisüsteemis. Hinnangute värvikoodide tähendusi ning selgitusi kirjeldab Tabel 4. Hinnangud võtab üldisemalt ülevaatliselt kokku Tabel 5.

Tabel 4: Alternatiivsete lahenduste hinnangute värvikoodid ning selgitused.

„ Suurepärane “ – lahenduse funktsionaalsus vastab täielikul määral vastava mõõdiku nõuetele.
„ Väga hea “ – lahenduse funktsionaalsus vastab suuremal määral vastava mõõdiku nõuetele ning vajab osalist kokkuleppelist mugandamist (k.a. lisatööd), et täita soovitud eesmärgid.
„ Hea “ – lahenduse funktsionaalsus vastab vastava mõõdiku nõuetele ning vajab keskmiselt rohkem kokkuleppelist mugandamist (k.a. lisatööd), et täita soovitud eesmärgid.
„ Rahuldav “ – lahenduse funktsionaalsus vastab vähesel määral vastava mõõdiku nõuetele ning vajab mahukat kokkuleppelist mugandamist (k.a. lisatööd), et täita soovitud eesmärgid.
„ Puudulik “ – lahenduse funktsionaalsus ei vasta vastava mõõdiku nõuetele või on funktsionaalsus puudulik. Kokkuleppelise mugandamise (k.a. lisatöö) tulemusel pole võimalik saavutada soovitud eesmäärke.

Tabel 5: Alternatiivsete lahenduste üldised hinnangud.

Mõõdiku nimetus	Hinnang lahendusele (<i>lahenduse nimetus</i>)		
	Atlassian Confluence	Axure RP	Targetprocess
Agiilne modelleerimine ja dokumenteerimine			
1. Piisavalt head lahendused			
2. Süsteemi arhitektuurile ja mudelitele keskenduv			
3. Teiste kaasamine protsessi			
4. Nõuded ja prioritseerimine			
5. Pidev dokumenteerimine			
6. Hiline dokumenteerimine			
7. Ühtne dokumentatsiooni			

hoiustamine			
Kerge dokumentatsioon (<i>lightweight documentation</i>)			
8. Spetsifikatsioon			
9. Taasesitus			
10. Kokkulepped ja tagasiside			
Lean printsiibid			
11. Ebaolulise vähendamine			
12. Õppimiskõver			
13. Terviku nägemine			
Aritmeetiline keskmine (<i>max – 5.000</i>)	4,154	3,539	4,154

6.1 Hinnangute selgitused

6.1.1 Atlassian Confluence

Piisavalt head lahendused (suurepärase) – viki lehtedena dokumentatsiooni kirjutamine loob võimaluse tarkvara komponentide dokumentatsiooni kirja panna võimalikult minimalistlikult andes siiski edasi piisavalt väärtust.

Negatiivse mõjuna võib sellise dokumenteerimise mõjuna tekkida palju dokumentatsiooni prahti, mis enam ei ole määrava väärtusega. Dokumentatsiooni haldamise lihtsusest tulenevalt, pannakse see dokumentatsiooni praht kõrvale. Pikema aja mõjuna hakkavad need dokumendid segama tervikpildi kujunemist.

Süsteemi arhitektuurile ja mudelitele keskenduv (väga hea) – viki lehtede vormingud tuleb mugandada vastavalt süsteemi arhitektuuri mudelile. Eelanalüüsi dokumenteerimise vormid tuleb kasutajal eraldi luua. Reaalsete mudelite loomine eeldab kolmanda tarkvaralise osapoole olemasolu.

Teiste kaasamine protsessi (suurepärase) – toote omanikel ja ka projekti meeskonnal on võimalik kõiki viki lehel olevaid tekste, pilte, graafikuid eraldiseisvalt kommenteerida lehel olevaid komponente aktiivseks tehes. Sellisel kujul on kerge teostada 4-silma kontrolli ning märkemeid autorile jätta. Confluence võimaldab töötada mitmel inimesel ühe viki lehe või nõude kallal, ilma, et tekiks suuremaid konflikte.

Nõuded ja prioritseerimine (hea) – nõuete teadmuspankade loomine eeldab projekti meeskonna kokkulepetest, kuidas neid nõudeid kirja pannakse või märgitakse. Eeldab

pakutavate vormide mugandamist. Süsteemi loomisel on vaja seda pidevalt hallata, et säilitada tervikpilt.

Pidev dokumenteerimine (suurepärane) – tegemist on terviklahendusega ning kogu meeskond on võimeline oma tegevusi pidevalt dokumenteerima vastavalt vajadusele.

Hiline dokumenteerimine (väga hea) – sarnaselt pidevale dokumenteerimisele on võimalik kogu tegevust dokumenteerida võimalikult hiliselt projekti elutsükli jooksul. Võrreldes Confluence hinnaliselt ülalpidamise kulusid, ei ole antud lähenemine selle lahenduse puhul soovitatav.

Ühtne dokumentatsiooni hoiustamine (suurepärane) – dokumentatsioon on ligipääsetav ühtses kohas, vajadusel tervele meeskonnale. Ligipääsetavus eeldab, et kogu dokumentatsioon esitatakse Confluence baasil.

Turvalisuse perspektiivist ei ole tegemist kõige parema lahendusega, kuna dokumentatsiooni ei hoiustata majasiseselt ning turvaliselt (nii palju kui see on võimalik).

Spetsifikatsioon (suurepärane) – lahendus võimaldab sisuliselt detailselt otsida kogu dokumentatsiooni kogumit. Dokumentatsiooni on võimalik igal ahahetkel täiendada vastavalt vajadusele.

Taasesitus (rahuldav) – prototüübid tuleb luua eraldiseisvatena väljaspool Confluence lahendust ning prototüübid on võimalik hiljem üles laadida piltide või muu meediafailide näol.

Kokkulepped ja tagasiside (suurepärane) – nagu teiste kaasamise juures on mainitud võimaldab lahendus toote omanikel ja ka projekti meeskonnal kõiki viki lehel olevaid tekste, pilte, graafikuid eraldiseisvalt kommenteerida lehel olevaid komponente aktiivseks tehes. Veelenam on projekti meeskonnal võimalik kogu dokumentatsiooni muuta.

Confluence versioneerib kõiki viki lehti, seega eelnevate kokkulepete tulemused alati säilivad. Dokumentatsiooni versioone on võimalik kerge vaevaga taastada.

Ebaolulise vähendamine (väga hea) – suuremal määral oleneb see analüütiku dokumentatsiooni haldamise oskusest, ning kuidas ta oma dokumentatsiooni üles märgib. Dokumentatsiooni on võimalik staatustega märkida valmis või mitte valmis olekusse, aga ka prioritseerida.

Õppimiskõver (hea) – viki lehtede puhul muutuvad vanad lehed, millegi uue õppimisel pigem dokumentatsiooni prahiks, mida enam uute asjade juurde tekkimisega üle ei uuendata. Confluence toetab ainult viki lehtede põhiste lähenemist, mistõttu on kasutaja limiteeritud kindlate võimaluste piiresse. Mitme (dokumenteermise) lahenduse proovimisel tuleb dokumentatsiooni vorme pidevalt mugandada.

Terviku nägemine (väga hea) – kui informatiivseid tarkvara projekti viki lehti pidevalt kaasajastakse ning need on muudatustele olnud vastuvõtlikud, siis on antud lahenduse põhjal tervikpildi nägemine väga hea.

6.1.2 Axure RP

Piisavalt head lahendused (suurepärase) – prototüüpimise põhisel lahendusel on piiramatud võimalused kuidas pakutavat lahendust piirata – alates lihtsamast piltlikust rakendusest kuni funktsionaalse rakenduseni välja.

Süsteemi arhitektuurile ja mudelitele keskendumine (suurepärase) – prototüübi loomisel, luuakse elementaarsel kujul süsteemi arhitektuur ning mudelite asemel tekib mingi tasemeline funktsionaalne rakendus. Prototüübist kujunevad välja infosüsteemi võimalikud nõrkused või tarkvaralised vead.

Teiste kaasamine protsessi (väga hea) – prototüüpimisse on võimalik kaasata kõiki kes on projektiga seotud, k.a. toote omanik. Oluline on jälgida, et igale meeskonna liikmele tuleb prototüüpimisega läheneda erinvalt, olenevalt nende oskustest ning kogemustest antud valdkonnas.

Nõuded ja prioritiseerimine (puudulik) – lahendus võimaldab prototüübi komponente nõuetega kirjeldada, kuid puuduvad head paindlikud võimalused neid väga hästi prioritiseerida või hallata.

Pidev dokumenteermine (suurepärase) – Axure RP võimaldab luua prototüübi loomise käigus väga detailset spetsifikatsiooni ning igat komponenti on võimalik pidevalt, võimalusterohkelt dokumenteermida. Samas saab lugeda prototüübi loomist kui pidevat dokumenteermist.

Hiline dokumenteermine (puudulik) – hiline dokumenteermine prototüübi vaatest eeldab kogu dokumentatsiooni olemasolu, mille alusel peab hakkama looma prototüüpi. Hiline prototüüp on oma olemuselt valmis rakendus.

Ühtne dokumentatsiooni hoiustamine (väga hea) – Axure RP võimaldab dokumentatsiooni ühtses kohas hoiustada. Sarnaneb SVN tüüpi lahendustele, kus dokumentatsiooni tuleb pidevalt uuendada, et see oleks ajakohane ning teistele ajakohaselt kättesaadav.

Spetsifikatsioon (väga hea) – prototüübi kõiki komponente saab väga detailselt dokumenteerida – alates üldisest kirjeldusest tehnilise funktsionaalsuse kirjelduseni välja. Axure RP võimaldab kogu prototüübi tehnilist kirjeldust väljastada Word'i või PDF dokumendina.

Taasesitus (suurepärase) – kõik nõuded esitatakse prototüübi või töötava rakenduse näol, tulenevalt rakenduse olemusest.

Kokkulepped ja tagasiside (hea) – tulenevalt dokumentatsiooni hoiustamise miinustest, peab tagasiside jaoks kogu dokumentatsiooni pidevalt edasi-tagasi saatma. Tagasisidet ning kokkuleppeid on parem teostada koosoleku vormis, kui kirjalikult.

Ebaolulise vähendamine (rahuldav) – ebaolulise vähendamisel tuleb oma prototüübi dokumentatsiooni ja lahendust suurepäraselt tunda. Vähendamisel eemaldatakse või kustutatakse prototüübist ebaolulised lahendused. Kuna puudub rakenduse poolne võimalus oma lahendusi versioneerida, siis kustutamisel pole võimalik lahendusi taastada (tuleb uuesti dokumenteerida ning prototüüpida).

Õppimiskõver (suurepärase) – prototüübi loomisel on analüütikutel võimalus läbi proovida erinevaid lahendusi ning oma ideed piltlikult valideerida.

Terviku nägemine (hea) – prototüübi näol on tegemist *mock-up* lahendusega lõplikust lahendusest. See annab tervikust piltliku nägemuse.

6.1.3 Targetprocess

Piisavalt head lahendused (suurepärase) – oma Scrum pileтите põhises ülesehituses on Targetprocess piisavalt heade lahenduste dokumenteerimiseks väga hästi üles ehitatud. Funktsionaalsused on jaotatud piletitteks ning iga funktsionaalsuse kohta on võimalikult minimalistlik kirjeldus.

Süsteemi arhitektuurile ja mudelitele keskendumine (rahuldav) – mudelid tuleb luua kolmanda osapoole lahenduste näol. Arhitektuurile ei ole võimalik palju rõhku panna, kui ei ole kokku lepitud ühtset vormingut arhitektuuri esitamiseks.

Teiste kaasamine protsessi (suurepärase) – toote omanikel ja ka projekti meeskonnal on võimalik dokumentatsiooni pileteid eraldiseisvalt kommenteerida iga pileti alla kommentaare jättes. Sellisel kujul on samuti kerge teostada 4-silma kontrolli ning autorile märkmeid jätta. Targetprocess võimaldab töötada mitmel inimesel ühe pileti kallal, ilma, et tekiks suuremaid konflikte. Tulenevalt projekti haldamise põhisest ülesehitusest on kõik muudatused Targetprocessi poolt dokumenteeritud.

Nõuded ja prioritseerimine (suurepärase) – Scrum põhimõttel toimivad vaated võimaldavad nõudeid hõlpsalt prioritseerida ning luua mitmekülgeid teadmuspanke vastavalt vajadusele.

Pidev dokumenteerimine (suurepärase) – tegemist on terviklahendusega ning kogu meeskond on võimeline oma tegevusi pidevalt dokumenteerima vastavalt vajadusele.

Hiline dokumenteerimine (hea) – sarnaselt pidevale dokumenteerimisele on võimalik kogu tegevust dokumenteerida võimalikult hiliselt projekti elutsükli jooksul. Hilise dokumenteerimise puhul, pole Targetprocessi funktsionaalsust kogu projekti arendustsükli käigus kasutada.

Ühtne dokumentatsiooni hoiustamine (suurepärase) - dokumentatsioon on ligipääsetav ühtses kohas, vajadusel tervele meeskonnale. Ligipääsetavus eeldab, et kogu dokumentatsioon esitatakse Targetprocess baasil.

Spetsifikatsioon (suurepärase) – *Scrum board* võimaldab detailseid ülevaateid kogu tarkvara projekti käekäigust ning tervest dokumentatsioonist.

Taasesitus (rahuldav) – prototüübid tuleb luua eraldiseisvatena väljaspool Targetprocess lahendust ning prototüübid on võimalik hiljem üles laadida piltide või muude meediafailide näol.

Kokkulepped ja tagasiside (suurepärase) – Targetprocess lahendus võimaldab toote omanikul osaleda analüüsi protsessis, kirjeldades esmaseid funktsionaalsusi, mille põhjal luuakse detailanalüüs ja dokumentatsioon. Kogu dokumentatsiooni on võimalik prioritseerida ja kommenteerida juba kirjeldamise ajal.

Ebaolulise vähendamine (väga hea) – suuremal määral oleneb see analüütiku dokumentatsiooni haldamise oskusest, ning kuidas ta oma dokumentatsiooni üles märgib.

Dokumentatsiooni on võimalik staatustega märkida valmis või mitte valmis olekusse, aga ka prioritseerida.

Õppimiskõver (hea) – dokumentatsiooni loomisel on võimalik kasutada erinevaid lahenduse viise, kui need jäävad rangelt Scrum raamidesse.

Terviku nägemine (suurepärane) – erinevate vaadete loomisest tulenevalt on tervikpildi loomine lihtne. Tervikut on võimalik personaliseerida või üles ehitada vastavalt vajadustele.

7 Pakutav lahendus

Antud skoobi raames on minimalistliku prototüübi tasandil realiseeritud ning kirjeldatud lahenduse kolm olulisemat ning peamist funktsionaalsust – nõuded, tellimine, taasesitus ja dokumenteerimine. Funktsionaalsused on kirjeldatud kujul, et tööga tutvujal on võimalik kogu funktsionaalsus taastada prototüübina.

Nõuded – töös on kirjeldatud nõuete vaatamine, loomine ja muutmine; kõik nõuete liigid ning nendega kaasnevad andmeväljad ja staatused.

Tellimine – kirjeldatud on tellimuste esitamine toote omaniku vaatest ning selle võimalikud variandid. Tellimuste esitamist on samuti laiendatud ärianalüütikutele.

Taasesitus ja dokumenteerimine – kirjeldatud on nõuete põhise tehnilise spetsifikatsiooni loomine ning nõuete taasesitamine prototüübi näol.

Funktsionaalsused on valitud lähtudes hetkeseisu kirjeldusest tulenevatest vajadustest, mis on kirjeldatud antud töö peatükis 0.

7.1 Nõuded

Nõuded on infosüsteemi dokumenteerimisel kõige olulisemad komponendid, ilma milleta pole võimalik lugeda infosüsteemi dokumentatsiooni korrektseks. Nõuded loovad alused programmeerijatele infosüsteemi funktsionaalsuste realiseerimiseks. Pakutud lahenduses kasutan Targetprocess'ile sarnast piletite põhise nõuete ülesmärkimist, toetades osaliselt nii *Scrum*'i pooldavaid meetodikaid.

Järgnevas peatükis annan detailse ärilise ning tehnilise kirjelduse, mille alusel nõuded ja nende loogika toimib pakutud lahenduses.




7.1.1 Andmeväljad

Tabel 6, Tabel 7 ja Tabel 8 kirjeldavad kõiki vormi modaalidel olevaid andmevälju. Andmeväljad on vastavalt grupeeritud kas metaandmeteks või sisestatavateks andmeteks. Ühtlasi on ära kirjeldatud kõik vormi juures olevad nupud.

Tabel 6: Nõude vaatamise modaali peamised andmeväljad.

Grupeering	Välja nimetus	Välja kirjeldus
Üldine info	Projekti nimi	Kirjeldab projekti, mille nõudega on tegemist.

Grupeering	Välja nimetus	Välja kirjeldus
	Nõude number	Ühe projekti raames unikaalne nõude number.
	Nõude versioon	Nõude muudatuste ajaloo määraja.
	Nõude omanik	Inimene, kes on nõude loonud.
	Loomise kuupäev	Kellaaja täpsusega kuupäev, millal nõue on loodud.
	Nõude valideerija	Inimene, kes on teostanud nõudele 4-silma kontrolli ning nõude valideerinud (versiooni põhiselt).
	Nõude staatus	Automaatselt määratud nõude staatus. Staatused on kirjeldatud eraldi Tabel 10.
	Nõude viimane muutja	Inimene, kes loonud nõudest viimase versiooni .
	Muutmise kuupäev	Kellaaja täpsusega kuupäev, millal nõuet on viimati muudetud.
	Nõude kinnitaja	Inimene, kes on nõude kinnitanud.
Nõude kirjeldus	Kinnitamise kuupäev	Kellaaja täpsusega kuupäev, millal nõue on kinnitatud.
	Nõude pealkiri	Lühike arusaadav pealkiri nõude paremaks mõistmiseks ning leidmiseks.
	Nõude prioriteet	Klassifikaatori põhine nimistu prioriteetidest. Mõjuvad nõude arhiveerimisele ning määravad ära tööde prioriteetide nimekirja.
	Nõude lühikirjeldus	Lühike ülevaade nõudest ning mida see reguleerib.
	Nõude eesmärk	Lühike põhjendus, miks kirjeldatud nõuet on vaja.
	Nõude kirjeldus	Detailne kirjeldus ning ülevaade nõudest ning selle omadustest.
	Manustatud dokumendid	Nõudele juurde lisatud nõuet täpsustavad dokumendid, pildid,

Grupeering	Välja nimetus	Välja kirjeldus
		vormid, mudelid jne.
Tagasiside ja märkused	Kommentaaride sisestamise lahter	Nõude kommenteerimise ning tagasiside andmise võimalus.
Nõude seosed	Seotud (liik)nõuded	Nõude kõik kirjeldatud seosed.
Nupud		Nupp määrab nupu vajutaja nõude valideerijaks.
		Nupp avab nupu vajutajale nõude muutmise modaali.
		Nupu vajutamisel kinnitub nõue lõplikult. Peale kinnitamist pole nõuet võimalik muuta ega valideerida.

Tabel 7: Nõude lisamise/muutmise modaali peamised andmeväljad.

Grupeering	Välja nimetus	Välja kirjeldus	Märkused
Üldine info	Projekti nimi	Klassifikaatoriline nimistu aktiivsetest projektidest.	Väli on kasutusel vaid nõude lisamise modaalis.
	Nõude liik	Klassifikaatoriline nimistu aktiivsetest nõuete liikidest.	
Nõude kirjeldus	Nõude pealkiri	Lühike arusaadav pealkiri nõude paremaks mõistmiseks ning leidmiseks.	Väli on kasutusel vaid muutmise modaalis.
	Nõude prioriteet	Klassifikaatori põhine nimistu prioriteetidest. Mõjuvad nõude arhiveerimisele ning määravad ära tööde	

Grupeering	Välja nimetus	Välja kirjeldus	Märkused
		prioriteetide nimekirja.	
	Nõude lühikirjeldus	Lühike ülevaade nõudest ning mida see reguleerib.	
	Nõude eesmärk	Lühike põhjendus, miks kirjeldatud nõuet on vaja.	
	Manusta faile	Detailne kirjeldus ning ülevaade nõudest ning selle omadustest.	
Nõude seosed	Otsi nõudeid	Seotud nõuete otsingu ja lisamise lahter.	

Tabel 8: Nõude muutmise/lisamise muud andmeväljad.

Kasutus (modaali nimetus)	Välja nimetus	Välja kirjeldus
Teenus	Andmeväljad	Andmebaasi tasandil kirjeldatud andmeväljad baasi parameetri täpsusega.
Komponent		
Komponent	Komponendi liik	HTML atribuudi põhine välja liik.

7.1.2 Nõude liigid

Nõuded on jaotatud liikideks, mille igal liigil on oma lühend ning tähendus. Nõuete vormid on liigiti erinevad (peamised andmeväljad on läbivalt samad), erinevused esinevad liikide omapärasuste tõttu.

Tabel 9 annab ülevaate kõikidest süsteemis kasutusel olevatest nõuete liikidest.

Tabel 9: Nõude liigid ja selgitused.

Nõude liik	Nõude liigi lühend	Nõude liigi kirjeldus
------------	--------------------	-----------------------

Nõude liik	Nõude liigi lühend	Nõude liigi kirjeldus
Üldine nõue	GR (<i>General Requirement</i>)	Nõuded, mida projekt peab järgima üle terve tarkvara või konkreetse funktsionaalsuse. Reguleerivad näiteks erinevate vormide ühesuguseid käitumismustreid või disaini.
Äriline nõue	BR (<i>Business Requirement</i>)	Organisatsiooni (ettevõtte) jaoks olulised protsessid ja vajadused. Selgitavad organisatsiooni edukriitilisi eesmärke, mida organisatsioon soovib infosüsteemi raames saavutada.
Süsteemne nõue (funktsionaalne)	FR (<i>Functional Requirement</i>)	Nõue, mis määrab infosüsteemi omaniku vajadused sisendinfo muutmiseks välisinfoks. See puudutab nii sisend- ja väljundinfo andmetüüpe, struktuuri, omavahelisi seoseid kui ka infosüsteemis toimuvaid arvutusi ning muudatusi struktuuris kõigi infosüsteemi liideste lõikes. [27]
Süsteemne nõue (mitte funktsionaalne)	NFR (<i>Non - Functional Requirement</i>)	Täpsustab tarkvara ja nõuete projekteerimises kriteeriume, mille alusel hinnatakse süsteemi töökäiku, mitte kindlaid omadusi. [27]
Funktsionaalsus	FN (<i>Functionality</i>)	Kirjeldab infosüsteemi mõnda suuremat tööprotsessi, mille tulemusel tagatakse infosüsteemi terviklahenduse toimimine. Antud liigiga nõuded sisestatakse eraldi modaalilt.
Teenus	S (<i>Service</i>)	Kirjeldab andmete liigutamisi kahe erineva või sama infosüsteemi piires andmeväljade täpsusega.

Nõude liik	Nõude liigi lühend	Nõude liigi kirjeldus
		Antud liigiga nõuded sisestatakse eraldi modaallilt.
Komponent	<i>C (Component)</i>	Kirjeldab süsteemiarhitektuurist tulenevaid infosüsteemi osasid (nupud, väljad, otsingud). Antud liigiga nõuded sisestatakse eraldi modaallilt.
Vorm	<i>F (Form)</i>	Kirjeldab infosüsteemis kasutajatele kuvatavaid vorme andmeväljade täpsusega. Võimalusel tuleks vormid kirjeldada koos prototüüpidega ja komponentidega. Antud liigiga nõuded sisestatakse eraldi modaallilt.

7.1.2.1 Nõude liikide omavahelised seosed ning arhitektuur

Nõudeid seostatakse omavahel mõjuanalüüsi käigus. Olenevalt infosüsteemi arhitektuurist või nõuete iseloomust võivad seosed olla hierarhiliselt lihtsad järjestikused (*waterfall*) (1) või komplekselt keerulised, kus olenemata nõude liigist on seosed kahepoolsed (2). Pakutav lahendus võimaldab kahepoolset seostamise ideed, vähendades sellega mõjuanalüüsile kuluvat ressursi ja aega.

1. Hierarhiliselt järjestikune seoste näide ühe projekti raames:

- Funktsionaalsus
 - Üldine nõue
 - Äriline nõue
 - Süsteemne nõue
- Teenus
 - Üldine nõue
 - Äriline nõue
 - Süsteemne nõue
 - Komponent

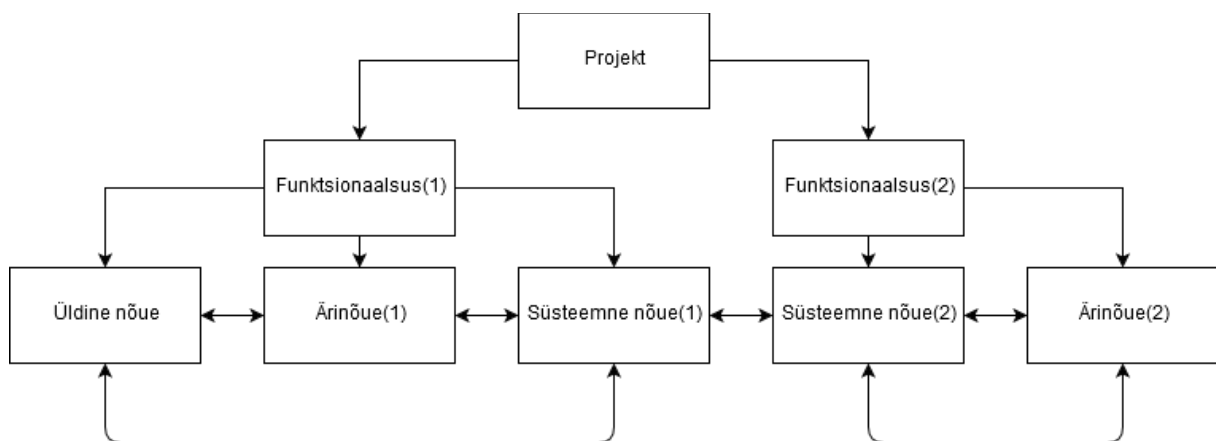
- Üldine nõue
 - Äriline nõue
 - Süsteemne nõue
- Vorm
 - Üldine nõue
 - Süsteemne nõue
 - Komponent

2. Kahepoolsete seoste näide ühe projekti raames:

Näidisprojektis on kaks funktsionaalsust – Funktsionaalsus(1) ja Funktsionaalsus(2). Funktsionaalsus(1) raames on ära kirjeldatud kolm nõuet – Üldine nõue, Ärinõue(1) ja Süsteemne nõue(1) ning Funktsionaalsus(2) raames kaks nõuet – Ärinõue(2) ja Süsteemne nõue(2).

Süsteemne nõue(1) on seotud mõlemapoolselt Ärinõude(1), Süsteemse nõude(2) ja Üldise nõudega. See tähendab, et nõuete kirjeldamisel on kasutatud viitamist teisele nõudele ning nad sõltuvad teineteisest. Sellest järeldatuna nõuete muutmisel tuleb jälgida seoseid teiste nõuetega (pidev ülevaade) ning kas nõude muutmisel on oluline muuta ka seonduvaid nõudeid.

Näitliku ülevaate seoste kohta annab Joonis 7.



Joonis 7: Projekti, funktsionaalsuse ning nõuete omavahelised seosed.

Pakutav lahendus muudab nõuete seostamise ning mõjuanalüüsi kasutajale lihtsamaks kasutades otsingupõhist nõuete seostamist. Seosed luuakse nõuete vahel kahepoolset (s.t. puudub vajadus siduda nõudeid omavahel mitu korda). Seotud nõudeid kuvatakse kasutajale

ülevaatlikult tabeli kujul koos nõude üldise informatsiooniga. Seosed filtreeritakse erinevatesse tabelitesse vastavalt nõude liigile, mis on kirjeldatud Tabel 9.

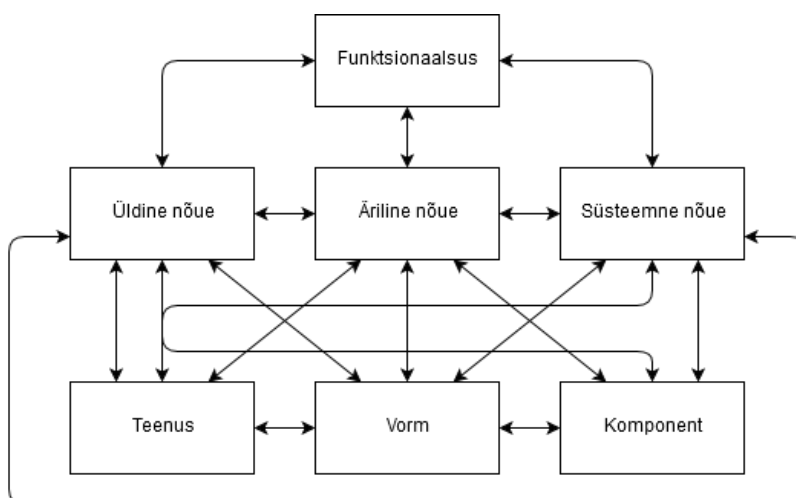
Seoste loomisega keskendub lahendus olemasolevate kirjelduste taaskasutamisele, mis tähendab, et ühtegi nõuet ega selles olevat infot ei tohiks sisestada dubleerivalt (parim võimalik olukord).

Joonis 8 kirjeldab seoste jaotuvust ühel nõudel. Seosed on jaotatud eraldi paneelidele vastavalt liigile ning on liigiti eraldi avatavad-suletavad (kerimise vähendamiseks). Tabeli nimekiri väljastab vaid olulisemat üldist informatsiooni nõuete numbrite, pealkirja staatuse ning omaniku kohta.

Nõude seosed			
Seotud ärinõuded			
Siin on nimekiri kõikidest seotud ärinõuetest!			
Nõude nimetus	Nõude pealkiri	Nõude staatus	Nõude omanik
TestProject1-BR-1001-13	Siin on mingisugune test pealkiri1.	Mustand	Asko Oruste
TestProject1-BR-1002-12	Siin on mingisugune test pealkiri2.	Analüüsimisel	Asko Oruste
TestProject1-BR-1003-11	Siin on mingisugune test pealkiri2.	Kinnitatud	Asko Oruste

Joonis 8: Nõude seoste vormiline näidis.

Joonis 9 kirjeldab nõuete seoste arhitektuuri. Vastavalt nõuete seostele Joonis 9 alusel tekitatakse vormil nõude seoste paneelis liigipõhised paneelkujul seotud nõuete filtreeringud.



Joonis 9: Nõuete seoste arhitektuur.

7.1.3 Nõude staatused

Igal nõudel on kindlal ajahetkel mingi staatus. Tulenevalt hetkeseisu kirjeldusest äri- ja süsteemianalüüsi osas peavad olema kaetud vähemalt elementaarsed pakutud; kinnitatud; töös ja tühistatud staatused.

Tabel 10 on kirjeldatud pakutud lahenduse kõik olemasolevad staatused ning nende hierarhilised ülemstaatused.

Tabel 10: Nõude staatused ja selgitused

Nõude staatus	Ülem staatus	Staatuse kirjeldus
Mustand	-	Esimene versioon nõudest on loodud. Staatus n.-ö. pakutud nõudele.
Analüüsimisel	Mustand	Nõude tingimusi ning täpsustusi analüüsitakse analüütiku poolt toote omanikuga koostöös. Peale igat muutmist on nõue uuesti selles staatuses.
Valideeritud	Analüüsimisel	Nõude versioon on üle vaadatud teise analüütiku poolt. Iga uue versiooniga tuleb seda korrata (vabatahtlikult).
Saadetud kinnitamisele	Analüüsimisel Valideeritud	Nõue on edastatud toote omanikule kinnitamiseks. Selles staatuses on võimalik nõudesse veel muudatusi sisse viia.
Kinnitatud	Saadetud kinnitamisele	Nõue on lõplikult valmis ning toote omaniku poolt õiguspäraselt kinnitatud. Nõuet pole võimalik enam muuta. Nõuet saab tühistada või arhiveerida.
Arhiveeritud	-	Nõue on realiseeritud, kuid nõuetega seotud funktsionaalsust on niivõrd palju muudetud, et nõue enam ei päde. Nõue on varasemas staatuses kui „Kinnitatud“ ning pikemat aega muutumatul kujul – nõue arhiveeritakse.

Nõude staatus	Ülem staatus	Staatuse kirjeldus
Tühistatud	-	Projekti eesmärk on muutunud nii suures mastaabis, et nõudel puudub vajadus. Arendusmeeskond on otsustanud nõuet mitte realiseerida.

7.1.3.1 Äri- ja funktsionaalsed nõuded

Nõude identifikaator	NõudeStaatuse-001
Sisu	
	<ol style="list-style-type: none"> 1. Kõikide nõuete esimene staatus on „Mustand“. <ol style="list-style-type: none"> a. Vajadusest lähtuvalt on see staatus võrdväärne staatusega „Pakutud“. 2. Esimene staatus määratakse nõuetele nõude loomise hetkel. 3. Staatuses „Mustand“ nõuded on alati versiooniga 0, ehk esialgne versioon.

Nõude identifikaator	NõudeStaatuse-002
Sisu	
	<ol style="list-style-type: none"> 1. Nõue võetakse töösse peale esimese muudatuse salvestamist. 2. Nõue muutub staatusest „Mustand“ staatusesse „Analüüsimisel“. 3. „Analüüsimisel“ staatuses olevad nõuded on alati nullist erineva versiooni numbriga. 4. Peale igat muutmist taastub nõude staatus staatusesse „Analüüsimisel“.

Nõude identifikaator	NõudeStaatuse-003
Sisu	
	<ol style="list-style-type: none"> 1. Igat töös olevat nõuet saab igal ajahetkel nõude omanikust erinev analüütik valideerida. <ol style="list-style-type: none"> a. Teostades põhjalikku 4-silma kontrolli ning vajutades „Valideeri“ nuppu. 2. Peale „Valideeri“ nupu vajutamist on nõude staatuseks „Valideeritud“. 3. Nõude igat versiooni on vaja eraldi valideerida.

Nõude identifikaator	NõudeStaatuse-004
Sisu	

1. Kõik nõuded, mis on analüütikul valmis kirjeldatud ning analüüsitud tuleb toote omanikul kinnitada.
2. Analüütik edastab nõuded edasi toote omanikule kinnitamisele.
 - a. Nõue märgitakse staatusesse „Saadetud kinnitamisele“.
3. Nõuet edastatakse kinnitamisele juhul kui see on staatustes „Analüüsimisel“ või „Valideeritud“.
4. Nõuet pole võimalik valideerida „Saadetud kinnitamisele“ staatuses.

Nõude identifikaator NõudeStaatuse-005

Sisu

1. Toote omanik kinnitab nõude peale temale kinnitamisele saatmist.
2. Nõude staatus muutub „Kinnitatud“ olekusse.
3. Nõuet pole võimalik antud staatuses enam muuta ega valideerida.

Nõude identifikaator NõudeStaatuse-006

Sisu

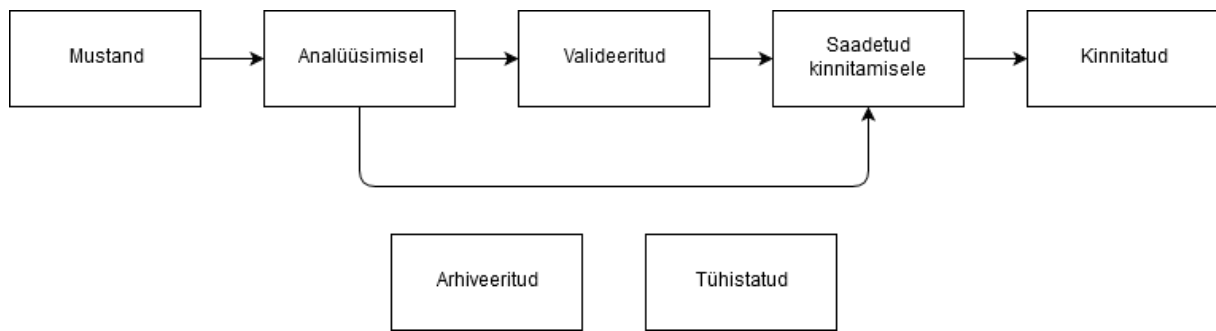
1. Nõuded arhiveeritakse peale mitte-aktiivset muutmist (viimasest muutmisest on möödunud kokkulepitud aeg).
2. Nõuetele märgitakse staatuseks „Arhiveeritud“.
3. Süsteem arhiveerib vaid nõudeid, mis on staatuses „Analüüsimisel“ või „Valideeritud“.
4. Nõude omanikule saadetakse hoiatus arhiveerimisest.
5. Arhiveeritud nõudeid on võimalik taastada nõuet muutes.

Nõude identifikaator NõudeStaatuse-007

Sisu

1. Nõue märgitakse staatusesse „Tühistatud“ peale nõudest loobumist.
2. Nõuet on võimalik tühistada igas nõude staatuses.
3. Tühistamisel nõue visuaalselt „kustutatakse“ (s.t. kasutaja enam nõuet taastada ei saa, kuid vajadusel on andmeid võimalik taastada hiljem andmebaasi kaudu).

Nõude staatuste struktuuri kujutab piltlikult Joonis 10.



Joonis 10: Nõude staatuste struktuur

7.1.4 Vormid ja prototüübid

Joonis 11 kirjeldab nõude vaatamise vormi (ärilise nõude näitel). Vaatamise vorm on jaotatud nelja peamisesse vaatesse – üldine info, nõude kirjeldus, tagasiside ja märkused, nõude seosed. Üldise info vaates on ära kirjeldatud kõik informatiivsed andmeväljad, mida rakendus muudab ise. Nõude kirjelduses on ära kirjeldatud kõik andmeväljad, mida nõude puhul on võimalik muuta või kirjeldada „Muuda“ nupule vajutades (avades muutmisvormi mis on kirjeldatud Joonis 13). Tagasiside ja märkuste funktsionaalsus on samasugune nagu Joonis 12 kujutatud. Nõude seoste vaade on samasugune nagu Joonis 8 kirjeldatud näidisel.

TestProject1-BR-1000-123

Üldine info

Projekti nimi:	Test Project 1	Nõude omanik:	Asko Oruste
Nõude number:	BR-1000	Loomise kuupäev:	01.01.2017 12:00:00
Nõude versioon:	123	Nõude valideerija:	Keegi Teine

Nõude staatus:	Mustand	Muutmise kuupäev:	02.01.2017 15:05:08
Nõude viimane muutja:	Asko Oruste	Kinnitamise kuupäev:	-
Nõude kinnitaja:	-		

Nõude kirjeldus

Nõude pealkiri:	Ärireegli vaatamise modaal
Nõude lühikirjeldus:	Vaade, mis annab informatiivse ülevaate ärireeglist.
Nõude eesmärk:	Vajalik prototüübi paremaks toimimiseks.

Nõude Kirjeldus:

Siin on detailne kirjeldus nõudest, võib olla pikem tekst. "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Manustatud dokumendid: [MingiDokument.pdf](#); [MingiDokument.docx](#); [MingiPilt.png](#)

Tagasiside ja märkused

Nõude seosed

[Valideeri](#) [Muuda](#) [Kinnita](#)

Joonis 11: Ärinõude vaatamise vormi näidis.

Joonis 12 kirjeldab nõude vormi tagasiside ning märkuste vaadet. Vaates on võimalik nõudele jätta kommentaaridena tagasisidet või märkuseid. Kõiki kommentaare on võimalik vastu võtta või tagasi lükata (kustutada) vastavast nupust.

The screenshot shows a web interface for a request titled "TestProject1-FR-900-1". It features several sections: "Üldine info", "Nõude kirjeldus", and "Tagasiside ja märkused". Under "Tagasiside ja märkused", there are three entries: a comment from "Asko Oruste" dated 01.01.2017 19:05:00 with a checkmark and an 'x' icon; a response from "Asko Oruste (Vastuvõetud!)" dated 01.01.2017 19:05:00 with an 'x' icon; and another comment from "Keegi Teine" dated 01.01.2017 20:05:00 with a checkmark and an 'x' icon. Below the comments is a text input field and a "Kommenteeri" button. At the bottom right, there are three buttons: "Valideeri", "Muuda", and "Kinnita".

Joonis 12: Nõude tagasiside vormi näidis.

Joonis 13 kirjeldab nõude muutmise vormi (funktsionaalse nõude näitel). Muutmise vorm on jaotatud kahte peamisesse vaatesse – nõude kirjeldus ja nõude seosed. Nõude kirjelduses on ära kirjeldatud kõik andmeväljad, mida nõude juures on võimalik muuta. Nõuete seoste vaade võimaldab sisu otsida ning siduda nõuet teiste nõuete või tellimustega. Andmete salvestamisel luuakse nõudest uus versioon.

TestProject1-FR-900-1 muutmine ✕

Nõude kirjeldus

Nõude pealkiri:

Nõude lühikirjeldus:

Nõude eesmärk:

Nõude kirjeldus:

Siin on detailne kirjeldus nõudest, võib olla pikem tekst. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Manusta faile: No file chosen
MingiDokument.pdf ✕;
MingiDokument.docx ✕;
MingiPilt.png ✕;

Nõude seosed

Otsi nõudeid:

Joonis 13: Nõude muutmise modaal.

Joonis 14 kirjeldab uue nõude lisamise vormi. Lisamise vorm on jaotatud kolme peamisesse vaatesse – üldine info, nõude kirjeldus, nõude seosed. Üldise info vaates on ära kirjeldatud kõik informatiivsed andmeväljad, mida uue nõude lisamisel tuleb käsitsi valida. Nõude kirjelduses on ära kirjeldatud kõik andmeväljad, mida tellijal on võimalik kirjeldada. Nõude seoste vaade allub funktsionaalsusele, mis on kirjeldatud Joonis 13 juures.

Uue nõude lisamine ✕

Üldine info

Projekti nimi:

Nõude liik:

Nõude kirjeldus

Nõude pealkiri:

Nõude lühikirjeldus:

Nõude eesmärk:

Nõude kirjeldus:

Manusta faile: No file chosen

Nõude seosed

Otsi nõudeid:

Joonis 14: Uue nõude lisamise modaal.

7.2 Tellimine

Tellimus on dokumendi kujul korrektselt vormistatud sisend arendusmeeskonnale. Oma pakutavas lahenduses keskendun rohkem toote omanikule, kes on infosüsteemi tellija ja ärianalüütikule, kes kirjeldab arendustellimusi. Süsteemianalüütiku koostatud arendustellimused on mõeldud rohkem programmeerijatele ning on koostatud programmeerimisülesande kujul. Viimane on antud lõputöö raames skoobiväline ning jäetud kirjeldamata. Seda põhjusel, et funktsionaalsus eeldab programmeerimisülesannete loomise võimalust ning nende olemasolu, mis on pigem antud töö jaoks edasiarendus arendajate jaoks.

7.2.1 Tellimuse koostamine ja esitamine

Tellimuste koostamine ning esitamine teostatakse võrreldes nõuete esitamisega erinevatelt vormidelt. Vormide erinevus tuleneb sellest, kas tegemist on sisulise sisendiga ärianalüütikutele või arendustellimus (funktsionaalsused, mida on vaja realiseerida) süsteemianalüütikutele ning arendusmeeskonnale.

7.2.1.1 Toote omaniku vaade

Toote omanik on isik, kes on infosüsteemi tellija ning sisuline konsultant, ehk inimene, kelle visiooni hakkatakse tehniliselt lahendama. Seetõttu on oluline teada minimaalselt mida on vaja ning mis väärtust see annab. Hetkel saadakse esmased vajadused kaardistatud eelanalüüsi käigus ning kogu projekti vältel saavutatakse kogu teadmine.

Tabel 11 kirjeldab tellimuse esitamise ja muutmise vormi kõiki andmevälju.

Tabel 11: Tellimuse koostamise ja esitamise andmeväljad.

Välja nimetus	Välja kirjeldus	Märkused
Funktsionaalsuse nimetus	Lühike nimetus funktsionaalsusele, mille järgi seda identifitseerida.	
Sihtgrupp	Kes on kasutajad, kes hakkavad antud funktsionaalsust enim kasutama.	Sihtgrupi defineerimine aitab kaasa infosüsteemi kasutajamugavuse loomisele. Vastab küsimusele, kes on kasutajad ning millised on nende kasutusharjumused.
Eesmärk	Millist lisandväärtust annab tellitav funktsionaalsus infosüsteemile ning miks on antud funktsionaalsus tellijale oluline.	
Kirjeldus	Tellija poolne detailne kirjeldus vastava funktsionaalsuse loomiseks.	Siin tuleb kirjeldada võimalikult täpselt, milline on vajadus.
Prioriteet	Klassifikaatoriline väärtus, mis	Prioriteet on kasulik nii

Välja nimetus	Välja kirjeldus	Märkused
	määrab tellija silmis töö olulisuse.	projektijuhile kui ka tellimuse töösse võtvale ärianalüütikule. Prioriteedi määramisel keskendutatakse rohkem väärtuse loomisele.
Lisa faile	Nupp, mille vajutamisel on tellijal võimalik üles laadida lisamaterjale piltide, dokumentide või näidis mallide näol.	
Loo prototüüp	Nupp, mis suunab kasutaja juturobot stiilis prototüüpimise keskkonda.	Prototüüp seostatakse automaaselt loodud tellimusega.
Informatiivsed väljad		
Projekti nimi	Projekt, mille raames tellimus on loodud.	
Tellijä nimi	Tellimuse loonud isiku nimi.	
Tellimuse kuupäev	Tellimuse loomise kuupäev.	
Tellimuse staatus	Klassifikaatoriline väärtus, mis määrab tellimuse staatuse.	Tellimuse staatused on kirjeldatud eraldi. Tellimuse staatusele märgitakse juurde isiku nimi, kes tellimuse staatuse määras.
Tellimuse muutja	Tellimust viimati muutnud isiku nimi.	Tellimuse muutja võib olla nii tellimuse looja, teine tellija või ärianalüütik.
Muutmise kuupäev	Tellimuse viimane muutmise kuupäev.	

Tabel 12 kirjeldab kõiki tellimuse prioriteetide väärtuseid.

Tabel 12: Tellimuse prioriteetide kirjeldused.

Tellimuse prioriteet	Prioriteedi kirjeldus
1 – kriitiline	Tellimus vajab kõrgendatud tähelepanu ning kiiremat reageerimist.
2 – väga oluline	Tellimused, mis on vajalik realiseerida eelisjärjekorras.
3 – oluline	Tavaväärtus, tellimused, mis on vaja realiseerida.
4 – vähem oluline	Tellimused, mis pole infosüsteemi jaoks niivõrd olulised, kuid võimalusel peaksid olema realiseeritud.
5 – mitte oluline	Lisand väärtus infosüsteemile, mis võetakse plaani kõige viimasena. Selliste tellimuste juures tuleb jälgida, kas neid on üldse vaja.

Tabel 13 kirjeldab kõiki võimalike tellimuse staatuseid ning nende struktuuri.

Tabel 13: Tellimuse staatuste kirjeldused.

Tellimuse staatus	Ülemstaatus	Staatuse kirjeldus
Ootel	-	Tellimus on esitatud ning on ootel staatuses.
Vastu võetud	Ootel	Ärianalüütik on tellimuse vastu võtnud ning koostab sellele ärianalüüsi.
Tagasi lükatud	Ootel	Ärianalüütik on tellimuse põhjendusega tagasi lükanud.
Tühistatud	Ootel	Tellijal on tellimuse tühistanud.
Analüüsitud	Vastu võetud	Tellimus on kaetud äriliste nõuetega.
Realiseeritud	Analüüsitud	Vastav tellitud funktsionaalsus on infosüsteemis realiseeritud.

7.2.1.1.1 Äri- ja funktsionaalsed nõuded

Nõude identifikaator TellimuseStaatus-0001

Sisu

Loodud tellimusele märgitakse alati esimeseks staatuseks „Ootel“.

Nõude identifikaator TellimuseStaatus-0002

Sisu

1. Ärianalüütik võtab tellimuse vastu või lükkab tagasi vajutades vastavat nuppu.
 - Vastu võtmisel muutub tellimuse staatus „Vastu võetud“. Tellimust pole võimalik enam muuta.
 - Tagasi lükkamisel muutub tellimuse staatus „Tagasi lükatud“. Tellimust on võimalik tellijal täiendada vastavalt tagasisidele.

Nõude identifikaator TellimuseStaatus-0003

Sisu

Tellija märgib tellimuse tühistatuks juhul, kui Ta on otsustanud tellitud funktsionaalsusest loobuda.

Nõude identifikaator TellimuseStaatus-0004

Sisu

Tellimus on staatuses „Analüüsitud“ kui see on kaetud tellija poolt kinnitatud nõuetega (S.t. kõik seotud ärilised nõuded on „Kinnitatud“ staatuses).

Nõude identifikaator TellimuseStaatus-0005

Sisu

Tellimus on staatuses „Realiseeritud“ kui see on kaetud süsteemsete funktsionaalsete kinnitatud nõuetega ning funktsionaalsus on valmis arendatud.

7.2.1.1.2 Vormid ja prototüübid

Joonis 15 kirjeldab toote omaniku poolt tehtava tellimuse vaatamise vormi. Vaatamise vorm on jaotatud nelja peamisesse vaatesse – üldine info, tellimuse kirjeldus, tagasiside ja märkused, tellimuse seosed. Üldise info vaates on ära kirjeldatud kõik informatiivsed andmeväljad, mida rakendus muudab ise. Tellimuse kirjelduses on ära kirjeldatud kõik

andmeväljad, mida tellijal on võimalik muuta või kirjeldada. Tagasiside ja märkuste funktsionaalsus on samasugune nagu Joonis 12 kujutatud. Tellimuste seoste vaade on samasugune nagu Joonis 8 kirjeldatud näidisel.

Tellimuse nimi test1

Üldine info

Projekti nimi:	Test Project 1	Tellimuse staatus:	Ootel (Toote Omanik)
Tellijä nimi:	Toote Omanik	Tellimuse kuupäev:	01.01.2017 12:00:00
Tellimuse muutja:	Toote Omanik	Muutmise kuupäev:	01.01.2017 12:00:00

Tellimuse kirjeldus

Funktsionaalsuse nimetus:	Tellimuse vaatamise modaal
Sihtgrupp:	Nupukad ja nutikad ärianalüütikud ja toote omanikud.
Eesmärk:	Vajalik prototüübi paremaks toimimiseks.
Kirjeldus:	Siin on detailne kirjeldus tellimusest, võib olla pikem tekst. "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."
Prioriteet:	1 - Kriitiline
Manustatud dokumendid:	MingiDokument.pdf ; MingiDokument.docx ; MingiPilt.png
Loodud prototüübid:	MingiVorm.html ; MingiLeht.html

Tagasiside ja märkused

Tellimuse seosed

Loo prototüüp Muuda Võta vastu Lükka tagasi Tühista

Joonis 15: Tellimuse vaatamise vorm.

7.2.1.2 Ärianalüütiku vaade

Ärianalüütik on nii tehniline kui sisuline vahendaja (tõlk) toote omaniku ning arendusmeeskonna vahel. Ärianalüütiku ülesanne tellimuste funktsionaalsuses on luua arendustellimusi vastavalt toote omaniku sisendi ning kirjeldatud äriliste nõuete põhjal. Arendustellimus on alus töö alustamiseks süsteemianalüütikule.

7.2.1.2.1 Äri- ja funktsionaalsed nõuded

Nõude identifikaator Arendustellimus-0001

Sisu

1. Ärianalüütik koostab arendustellimuse vajutades „Uus arendustellimus“ nuppu.
2. Kasutajale kuvatakse arendustellimuse lisamise vorm.

Nõude identifikaator Arendustellimus-0002

Sisu

Ärianalüütik otsib ning valib lisamise vormil välja soovitud funktsionaalsused või individuaalsed ärinõuded, mida on võimalik kirjeldada süsteemsete nõuetena.

Nõude identifikaator Arendustellimus-0003

Sisu

1. Nõuete ning funktsionaalsuste otsingud otsivad ainult toote omaniku poolt kinnitatud nõuete seast.
 - a. Funktsionaalsuste otsing kaasab kõik funktsionaalsuega seotud ning kinnitatud ärinõuded.

Nõude identifikaator Arendustellimus-0004

Sisu

Arendustellimus nummerdatakse automaatselt süsteemi loogika alusel.

Nõude identifikaator Arendustellimus-0005

Sisu

1. Süsteemianalüütikule edastatakse teavitus soovitud arendustellimusest.
 - a. Süsteemianalüütikul on võimalus tellimusest keelduda või see vastu võtta.

7.2.1.2.2 Vormid ja prototüübid

Joonis 16 kirjeldab uue arendustellimuse lisamise vormi. Vorm jaotub kahte peamisesse vaatesse – üldine info ning nõuete ja funktsionaalsuste lisamine. Üldine info tuleb tellimuse loojal täita käsitsi. Nõuete ja funktsionaalsuste vaade sarnaneb Joonis 13 juures kirjeldatud nõuete otsingu süsteemile.

Uue arendustellimuse lisamine

Üldine info

Projekti nimi:

Nõuete ja funktsionaalsuste lisamine

Otsi funktsionaalsust:

Otsi ärinõudeid:

Siin on nimekiri kõikidest lisatud funktsionaalsustest ja ärinõuetest!

Nõude nimetus	Nõude pealkiri	Nõude staatus	Nõude omanik
TestProject1-BR-1001-13	Siin on mingisugune test pealkiri1.	Kinnitatud	Asko Oruste
TestProject1-BR-1002-12	Siin on mingisugune test pealkiri2.	Kinnitatud	Asko Oruste
TestProject1-BR-1003-11	Siin on mingisugune test pealkiri2.	Kinnitatud	Asko Oruste

Loo tellimus

Joonis 16: Arendustellimuse lisamise vorm.

7.3 Taasesitus ja dokumenteerimine

Kõikidest pakutava lahenduse kirjeldatud funktsionaalsustest, pole antud funktsionaalsus prototüübi tasemel realiseeritud.

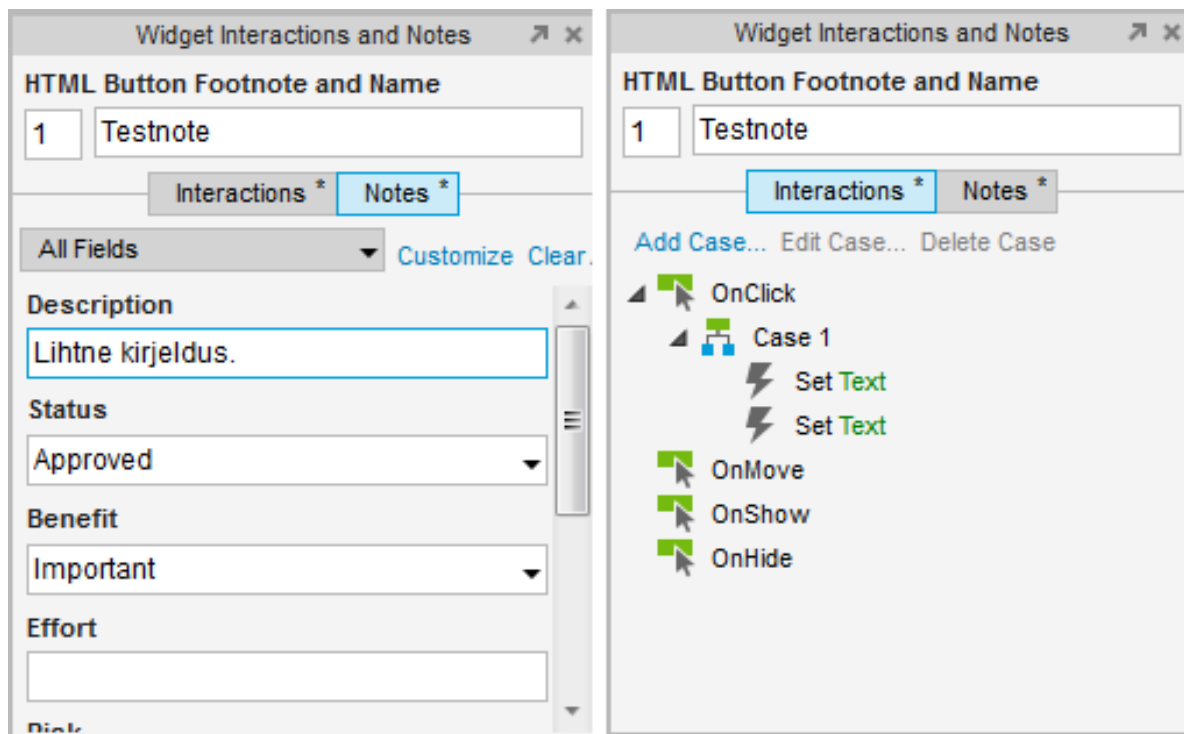
7.3.1 Nõuete taasesitus

Nõuete juures olen välja toonud, et kasutajal on võimalus kirjutada dokumentatsiooni infosüsteemi komponentidele ja vormidele. Antud liiki nõuete korrektsel vormistamise tulemusel pakub lahendus võimalust luua esialgne, vormide põhine protoüüp (dokumentatsioon-enne lahenduskäik). Analüütikul on võimalus lisada prototüübid vormide

nõuetele juurde, omamata seejuures kogemust prototüübi loomises, kuna rakendus genereerib prototüübi vastavalt analüütiku kirjeldustele.

Vastupidiselt dokumentatsioon-enne prototüübi loomisele, pakub lahendus ka vastupidist lähenemisviisi ehk prototüüp-enne lahenduskaiku. Antud lahendus jaotub suures osas kaheks:

1. Ärianalüütik loob prototüübi ning süsteem genereerib prototüübile nõuded liigiga „Vorm“ ja „Komponent“. Sarnaselt Axure RP lahendusele on prototüübi loomise ajal analüütikul võimalus kõiki komponente täiendada märksõnade või nõuetega (Joonis 17). Nimetatud märksõnad ja nõuded genereerib süsteem ümber infosüsteemi komponentide ja vormi nõueteks. Prototüüpi loov analüütik peab äri- ja funktsionaalsete nõuete seosed hiljem ise looma.



Joonis 17: Axure RP vormi komponentide kirjeldamine.

2. Prototüübi loomise ajal kontrollib prototüüp pidevalt olemasolevaid vorme ning komponente taaskasutamise eesmärgil. Analüütikul on võimalus kinnitada, et tegemist on sama komponendiga mis on varasemalt kirjeldatud või on tegu uue komponendiga, mille tulemusel luuakse uus nõue. Säilitades nii lahenduse põhimõtte sisestada kõik asjad ühekordselt.

Lahendusega kaasnevad positiivsed omadused:

- Prototüübi loomine on kasutajatele muudetud kergemaks ning rohkem automatiseeritud.
- Prototüüp on kaetud nõuetega.
- Antud kujul prototüüp on informatiivsem ja ülevaatlikum.

Lahendusega kaasnevad riskid:

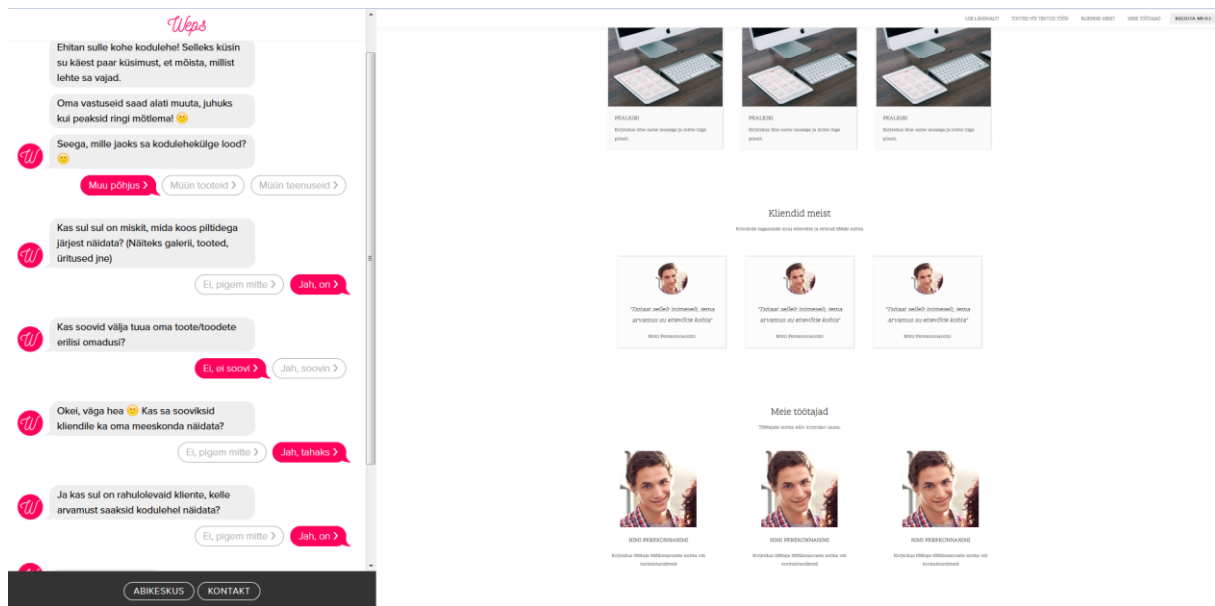
- Kasutaja keskendub pigem prototüübi loomisele, kui nõuete kirjeldamisele.
- Seotud ja genereeritud nõuded luuakse tühjadena.

7.3.2 Tellimuste prototüüpimine

Ärianalüütikule tellimuste vormistamise juures on toote omanikul võimalus luua prototüüpe. Antud prototüüpimise variandi eesmärgid on anda toote omanikule kohest tagasisidet ilma analüütikute sekkumiseta ning luua visuaalne pilt toote omaniku nägemusest.

Prototüübi loomine ei nõua programmeerimise oskuseid või teadmisi arendusest. Idee põhineb *chatbot*'i abiga veebilehtede loomise lahendusel, mille on välja töötanud Weps [28]. Sarnaselt lahenduse ideele peab toote omanik vastama erinevatele küsimustele, mille tulemusel luuakse reaalses kõrvale rakenduslik visuaalne näidis prototüüp.

Joonis 18 kuvab Weps rakenduse töötamise põhimõtet. Kasutaja vastab vasakul olevale *chatbot*'i küsimustele ning paremale kuvab reaalses loodud tulemust. Küsimuste vastuseid on võimalik igal hetkel muuta ning prototüüp uuendab end vastavalt vastustele.



Joonis 18: Weps rakenduse näide.

Lahendusega kaasnevad positiivsed omadused:

- Tellija mõtleb oma sisendid paremini läbi.
- Tellija suudab kommunikeerida oma visiooni paremini.
- Tellija on kaasatud rohkem analüüsi ning arenduse protsessi.
- Kohene tagasiside.

Lahendusega kaasnevad riskid:

- Tellija loodud prototüüpe hakatakse tegema üks-ühele ümber infosüsteemiks.
- Tellijast saab ärianalüütik. Tellijal puudub kompetents ja kogemus kommunikeerida oma ideid tehnilise taustaga inimestele.
- Tellija poolsed pakutud lahendused on tellija meelest kõige optimaalsemad ning on vaja selliselt realiseerida nagu kirjeldatud.
- Piirangud (näiteks täpne paigutus vormil) küsimuste kommunikeerimisel ning rakenduse prototüüpimisel.

7.4 Rollid ja õigused

Süsteemil on selgelt kirjeldatud kolm rolli – tellija/toote omanik, ärianalüütik, süsteemianalüütik. Tabel 14 on välja toodud kõikide rollide selgitused.

Tabel 14: Rollide selgitused.

Rolli nimetus	Rolli selgitus
Tellija	Sisuteadlik inimene, kellel on kindel visioon loodavast infosüsteemist. Tellija on kursis kõikide ärilist sisu puudutavate teemadega.
Ärianalüütik	Sisu-tehniline inimene, kelle ülesanne on kaardistada kõik tellija poolsed ärilised vajadused. Samuti olla vahendaja tellija ning süsteemianalüütiku vahel. Omab osalisi või täielike teadmisi infosüsteemi teema valdkonnas.
Süsteemianalüütik	Tehniline inimene, kelle ülesanne on välja pakkuda tehnilised lahendused vastavalt kaardistatud äriliste vajadustele.

Tabel 15 kirjeldab kõikide rollide õiguseid erinevate vormide lõikes CRUD (loomine (*create*); lugemine (*read*); muutmine (*update*); kustutamine (*delete*)) maatriksi näol. Roheline värv määrab õiguse olemasolu, punane õiguse puudumise.

Tabel 15: Õiguste kirjeldused CRUD maatriksina.

Vormi liikide nimetused	Rolli nimetus											
	Tellija				Ärianalüütik				Süs.analüütik			
	C	R	U	D	C	R	U	D	C	R	U	D
Üldise nõude vorm												
Ärilise nõude vorm												
Süsteemse nõude (funktsionaalne) vorm												
Süsteemse nõude (mitte funktsionaalne) vorm												
Funktsionaalsuse vorm												
Teenuse vorm												
Komponendi vorm												
Vormi vorm												
Tellimuse vorm												
Arendustellimuse vorm												

8 Pakutava lahenduse hinnang

Hinnangu andmisel kasutan sama meetodikat nagu alternatiivsete lahenduste hindamisel.

Tabel 16: Pakutava lahenduse üldised hinnangud.

Mõõdiku nimetus	Hinnang lahendusele
Agiilne modelleerimine ja dokumenteerimine	
1. Piisavalt head lahendused	
2. Süsteemi arhitektuurile ja mudelitele keskenduv	
3. Teiste kaasamine protsessi	
4. Nõuded ja prioritseerimine	
5. Pidev dokumenteerimine	
6. Hiline dokumenteerimine	
7. Ühtne dokumentatsiooni hoiustamine	
Kerge dokumentatsioon (<i>lightweight documentation</i>)	
8. Spetsifikatsioon	
9. Taasesitus	
10. Kokkulepped ja tagasiside	
Lean printsiibid	
11. Ebaolulise vähendamine	
12. Õppimiskõver	
13. Terviku nägemine	
Aritmeetiline keskmine (<i>max – 5.000</i>)	4,769

Piisavalt head lahendused (suurepärase) – sarnaselt Targetprocess'le on pakutud lahendus üles ehitatud piletite põhiste nõuete süsteemile, mis on ära jaotatud vastavalt nõude liigile. Eksisteerib võimalus luua piisavalt häid lahendusi, kuna nõude piletid vajavad minimalistlike kirjeldusi.

Süsteemi arhitektuurile ja mudelitele keskenduv (väga hea) – tellimuste vormistamine eelanalüüsi raames soosib arhitektuurilist mõtlemist. Samuti pidev poolautomaatne seoste loomine mõjuanalüüsi näol on nõuded kirjeldatud süsteemi arhitektuuri silmas pidades.

Erinevaid mudeleid (näiteks baasimudeleid) ning UML skeeme antud lahendus ei toeta.

Teiste kaasamine protsessi (suurepärase) – lahendus kaasab oma olemuselt rohkem toote omanikku kui tellijat analüüsi protsessi, muutes tellija osaliselt ärianalüütikuks. Lahendus võimaldab igat nõuet eraldi kommenteerida ning tagasisidestada. Nõuete ülesehitus soosib pidevat 4-silma kontrolli meeskonnakaaslaste vahel.

Nõuded ja prioritseerimine (suurepärase) – loodud nõuded alluvad vastavalt nõude liigile ühtsele ette antud mallile. Nõudeid on võimalik tellimuste kaudu prioritseerida. Kõiki nõudeid on kirjeldatud mudelile võimalik omavahel siduda ning pidevalt taaskasutada.

Pidev dokumenteerimine (suurepärase) – kogu meeskond on võimaldatud oma tegevusi dokumenteerima terve projekti elutsükli vältel.

Hiline dokumenteerimine (suurepärase) – sarnaselt pidevale dokumenteerimisele on võimalik kogu tegevust dokumenteerida võimalikult hiliselt projekti elutsükli jooksul.

Ühtne dokumentatsiooni hoiustamine (suurepärase) – dokumentatsioon on ligipääsetav ühtses kohas, vajadusel tervele meeskonnale.

Spetsifikatsioon (suurepärase) – lahendus võimaldab sisuliselt detailselt otsida kogu dokumentatsiooni kogumit. Dokumentatsiooni on võimalik igal ajahetkel täiendada vastavalt vajadusele.

Taasesitus (suurepärase) – kõik nõuded on võimalik siduda prototüüpidega, sidumine on kasutajasõbralik. Prototüüpe pole keeruline luua ning sellega saab hakkama ka programmeerimisega mitte tegelev inimene.

Kokkulepped ja tagasiside (väga hea) – lahendus toetab 4-silma kontrolli ning nõuete individuaalset tagasisidestamist ja kommenteerimist.

Tagasisidestamise eelduseks on, et kasutajal on pidev ülevaade enda kirjeldatud nõuetest.

Ebaolulise vähendamine (suurepärase) – nõuetele on lisatud arhiveerimine, mis eemaldab pikalt, ebaaktiivselt seisvad mittekinnitatud nõuded. Dokumentatsiooni nõuded on kaetud staatustega ning tellimuste kaudu on võimalik kõike prioritseerida.

Õppimiskõver (*suurepärane*) – mitmekülgne prototüüpimise lahendus võimaldab luua erinevaid lahendusi ning katsetada nende sobivust, õppides jooksvalt kogemustest. Dokumentatsiooni võib luua peale prototüüpe.

Terviku nägemine (*suurepärane*) – lahendus võimaldab näha tervikut väga mitmel tasemel, tulenevalt nõuete seotuse funktsionaalsustest.

9 Kokkuvõte

Antud töö peamiseks eesmärgiks on luua ühtne sobilik standardne tööprotsessi mall infosüsteemi dokumentatsiooni loomiseks, samuti töötada välja metoodika tööprotsessi läbiviimiseks, mis toetab loodud tööprotsessi malli.

Töö käigus on selgitatud kuidas antud eesmärke saavutada. Esmalt on uuritud ning analüüsitud kuidas näevad välja rakendatavas ettevõttes praegused kasutusel olevad tööprotsessid ja vormid. Sellega seoses on pandud paika esmavajadused pakutavale lahendusele ning alternatiivsetele lahendustele. Seejärel on alternatiivsete lahendustena valitud kolm kolmanda osapoolse tarkvara (Atlassian Confluence, Axure RP ja Targetprocess). Analüüsitud on nende vastavust esmavajadustele ning hinnatud erinevate mõõdikute vastu. Mõõdikud on autori poolt valitud ning kokku pandud lähtudes ettevõtte strateegiast ning lähenemisest töö teema valdkonnale.

Töö tulemusena on valminud analüüsiv ning hindav kirjeldus olemasolevate lahenduste lähenemisest sarnasele probleemile. Samuti on töö tulemusena valminud süsteemianalüüs süsteemile, mis muudab analüüsi dokumenteerimise protsessi ühtlasemaks ning lõppkasutajale kiiremaks ja ülevaatlikumaks. Töö viimase tulemuse valideerimiseks on loodud minimaalse funktsionaalsusega vormidel põhinev veebipõhine näiterakendus. Vastavalt analüüsile on näiterakendust võimalik taas-luua.

10 Kirjanduse loetelu

- [1] Registrate ja Infosüsteemide Keskus. [Online]. <http://www.rik.ee/asutusest>
- [2] Registrate ja Infosüsteemide Keskuse põhimäärus (18.04.2007), RT I, 28.03.2015, 11.
- [3] Arendusega ja arendamisega seotud kokkulepitud mõisted. [Asutussisene dokument].
- [4] P. Saartee. Ärianalüüsi koostamise hea tava. [Asutussisene dokument].
- [5] Majandus- ja Kommunikatsiooniministeerium; Riigi Infosüsteemi Amet. Kasutajasõbralike e-teenuste disainimine Maanteeameti näitel. [Online]. https://www.ria.ee/public/publikatsioonid/E-teenuste_disainimise_kasiraamat.pdf
- [6] O. Serrat. (2009) Asian Development Bank. [Online]. <https://www.adb.org/sites/default/files/publication/27641/five-whys-technique.pdf>
- [7] RIK Arendustellimuse koostamise juhend. [Asutussisene dokument].
- [8] Nõuete kirjeldamise ja haldamise põhimõtted ja reeglid (olemasoleva lahenduse põhiselt). [Asutussisene dokument].
- [9] How to document product requirements in Confluence. [Online]. <https://confluence.atlassian.com/doc/blog/2015/08/how-to-document-product-requirements-in-confluence>
- [10] Süsteemianalüüsi põhimõtted. [Asutussisene dokument].
- [11] Head ja halvad analüüsi praktikad. [Asutussisene dokument].
- [12] Detailanalüüsi miimumnõuded nõuetele/reeglitele. [Asutussisene dokument].
- [13] Atlassian Confluence. [Online]. <https://www.atlassian.com/software/confluence>
- [14] Atlassian Confluence pricing. [Online]. <https://www.atlassian.com/software/confluence/pricing?tab=cloud>
- [15] Atlassian software. [Online]. <https://www.atlassian.com/software>
- [16] Axure. [Online]. <https://www.axure.com/>
- [17] Axure RP pricing. [Online]. <https://www.axure.com/buy>
- [18] Targetprocess. [Online]. <https://www.targetprocess.com/>
- [19] Targetprocess pricing. [Online]. <https://www.targetprocess.com/pricing/>

- [20] IIBA, *A Guide To The Business Analysis Body of Knowledge*.: International Institute of Business Analysis, 2015.
- [21] C S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*.: Wiley Series in Systems Engineering and Management, 2005.
- [22] Agile Modeling. [Online]. <http://www.agilemodeling.com/>
- [23] Z. Zhang et al., "Towards Lightweight Requirements Documentation," *Journal of Software Engineering and Applications*, vol. 3, no. 9, p. 882, 2010.
- [24] T. Poppendieck and M. Poppendieck, *Lean Software Development: An Agile Toolkit*.: Addison-Wesley Professional, 2003.
- [25] C. Hibbs, S. Jewett, and M. Sullivan, *The Art of Lean Software Development: A Practical and Incremental Approach*.: O'Reilly Media, Inc., 2009.
- [26] Black-box testing. [Online]. <http://softwaretestingfundamentals.com/black-box-testing/>
- [27] IREB - International Requirements Engineering Board. [Online]. https://www.ireb.org/content/downloads/2-syllabus-foundation-level/ireb_cpre_syllabus_fl_en_v22.pdf
- [28] Weps. [Online]. <https://www.getweps.com/>