

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogiateaduskond

Liis Harjurand 178032IABM

TARKVARAPROJEKTI PLANEERIMINE TEENINDUSETTEVÕTTE X NÄITEL

Magistritöö

Juhendaja: Epp Immato
MSc

Tallinn 2019

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: Liis Harjurand

03.05.2019

Annotatsioon

Magistritöö on kirjutatud tarkvara projekti planeerimisest. Magistritöö käsitleb tarkvara projekti erinevate etappide tegevusi ning toob välja erinevad meetodikad projekti juhtimiseks ning arenduse teostamiseks. Magistritöö on koostatud reaalse ettevõtte põhjal ning analüüsi tulemusi ja soovitusi kasutatakse projekti teostamisel. Magistritöö raames on teostatud küsitlus, intervjuud osakonnajuhtidega ning tehtud juhtumianalüüs.

Magistritöö tulemuseks on erinevate teadusallikate põhjal projekti läbiviimist ning arendust toetavate praktikate pakkumine ning näidisprojektile sobiva meetodika valiku tegemine. Magistritöö raames on kaardistatud hetkeolukord tarkvaraga rahulolu osas ning juhtumianalüüsi käigus on hinnatud teostatud tarkvara projekti põhjal protsessi efektiivsust. Samuti on kaardistatud esialgsed nõuded tarkvarale.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 65 leheküljel, 6 peatükki, 16 joonist, 9 tabelit

Abstract

Software project planning on a basis of service organization x

The master thesis is written about software project planning. The thesis is based on real organization who is considering changing current software. The thesis consists of software project activities and brings out different methodologies to manage project and development. The results of the thesis are used in project planning. With this thesis a collection of templates is created for the organization to improve software procurement process.

Master thesis incorporates the result of end-user survey, case-study of previous software procurement process and interviews with department managers and management member. The result of the survey showed that although the satisfaction is relatively good, there are areas in functionality that need improvement. Further interviews with department managers and management member gave more detailed information about required software features. With this thesis it was concluded that main functionality that was deficient in current software is tools for analyse.

The result of this thesis is a collection of suggestions that are based on scientific sources. With this master thesis current situation is mapped, and initial requirements are analyzed. Also, a suggestion for suitable development methodology is offered based on interviews and previous experience. The result of case-study of the software procurement process is also evaluated.

The thesis is in estonian languages and contains 65 pages of text, 6 chapters, 16 figures, 9 tables

Lühendid ja mõisted

Agiilne metoodika	(inglise keeles <i>Agile method</i>) iteratiivne tarkvara arendusmetoodika, kus tarkvara toodetakse väikeste etappidena ning tulemust testitakse peale igat etappi. Eeldab tellija ja tarnija väga tihedat koostööd.
Kasutuslugu	(inglise keeles <i>Use case</i>) süsteemi funktsionaalsuse esitamine skeemina või kirjeldusena
Koskmudel	(inglise keeles <i>Waterfall method</i>) traditsiooniline arendusmeetod, kus arendus teostatakse lineaarsete etappidena
KPI	(inglise keeles <i>key performance indicator</i>) võtmetulemuslikkuse näitajad soovitud eesmärgi saavutamisel
PMBOK	(inglise keeles <i>Project Managment Body of Knowledge</i>) projektijuhtimise metoodika
PRINCE2	(inglise keeles <i>PRojects IN Controlled Environments</i>) projektijuhtimise metoodika

Sisukord

1 Sissejuhatus	10
1.1 Taust ja probleem	10
1.2 Ülesande püstitus	11
1.3 Metoodika	11
1.4 Ülevaade tööst	12
2 Ülevaade tarkvara arendusprotsessist.....	13
2.1 Nõuete analüüs	14
2.1.1 Funktsionaalsed ja mittefunktsionaalsed nõuded	16
2.2 Skoop	17
2.3 Lahenduse teostus	18
2.4 Riskide juhtimine.....	18
2.5 Kasutuselevõtt	20
3 Ülevaade projektijuhtimise metoodikatest.....	22
3.1 PRINCE2	22
3.2 Etapid ja otsustuskohad	23
3.3 PMBOK	25
3.4 Etapid ja otsustuskohad	25
3.5 Metoodikate võrdlus	26
4 Arendusmetoodika valik	28
4.1 Agiilsed meetodid.....	28
4.2 Koskmudel.....	29
4.3 Meetodite kasutus	29
5 Projektimeskonna planeerimine.....	32
5.1 Sotsiaalsed protsessid tarkvara arendusprojektides	33
5.2 Meeskonnaliikmete mõju kuludele.....	35
5.3 Vastutus ja rollid.....	35
6 Näidisprojekt ettevõttes X	37

6.1 Juhtumianalüüs	37
6.1.1 Kokkuvõtted intervjuust	38
6.2 Projektiplaan ja meeskond projekti juhtimiseks	40
6.2.1 Ärilise vajaduse põhjendus	40
6.2.2 Projekti läbiviimise meetodika	40
6.2.3 Hetkeolukord ja rahuolu tarkvaraga	41
6.3 Skoop	45
6.4 Ajakava	46
6.5 Nõuete kaardistus	46
6.6 Meeskond	50
6.7 Riskide kaardistus projektis	51
6.8 Tulemite vastuvõtmine	52
6.9 Järeldused ja valideerimine	52
Kokkuvõte	55
Summary	57
Kasutatud kirjandus	59
Lisa 1 Küsimustik	63
Lisa 2 Intervjuu	64
Lisa 3 Juhtumianalüüsi intervjuu	65

Jooniste loetelu

Joonis 1. Nõuete jälgitavus (Kannenberga ja Saiediani 2010, lk 4).....	15
Joonis 2. Jälgitavust mõjutavad tegurid (Gotel, Finkelstein 1994, lk 98).....	15
Joonis 3. Riski juhtimise mudel (Dey <i>et al.</i> 2007, lk 290).....	19
Joonis 4. PRINCE2 protsessimudel Siegelaubi põhjal (Siegelaub 2004, lk 3).....	24
Joonis 5. PMBOK Etapid (Project management fundamentals...2019).....	26
Joonis 6. Koskmeetodi ja Agiilsete meetodide etappide võrdlus (Huo <i>et al.</i> 2004).....	30
Joonis 7. Projekti etapid.....	41
Joonis 8. Rahulolu olemasoleva tarkvaraga (Küsitlus 2019).....	42
Joonis 9. Rahulolu tarkvaraga, keskmine ja jaotus (Küsitlus 2019).....	43
Joonis 10. Funktsionaalsuse piisavus kasutajate arvates (Küsitlus 2019).....	43
Joonis 11. Tarkvarast info kättesaadavusega rahulolu (Küsitlus 2019).....	44
Joonis 12. Korrelatsioon info kättesaadavuse ja rahulolu tarkvaraga vahel (Küsitlus 2019).....	45
Joonis 13. Skoobi määratlemise ettevõtte X.....	45
Joonis 14. Ostuarve protsess.....	48
Joonis 15. Näide ettevõtte ühest põhiprotsessist.....	49
Joonis 16. Näidisprojektile sobilik struktuur.....	51

Tabelite loetelu

Tabel 1. Tugevad ja pehmed kriteeriumid (Albert <i>et al.</i> 2017 lk 809).....	20
Tabel 2. Neli etappi projektijuhtimises (Maylor 2003, lk 28).	22
Tabel 3. PMBOK ja PRINCE 2 võrdlus Siegelaub põhjal (Siegelaub 2004, lk 2)	26
Tabel 4. Projektijuhtimise rollid (Zwikael ja Meredith 2018, lk 485)	35
Tabel 5. Projekti eeldatav ajakava	46
Tabel 6. Mall kasutuslugudeks	47
Tabel 7. Rollide jaotus näidisprojektis	50
Tabel 8. Mall riskide kaardistuseks projektis X	51
Tabel 9. Tulemite vastuvõtmise mall	52

1 Sissejuhatus

Magistritöö on kirjutatud teenindusettevõttest, kus töötab üle 2000 töötaja ning mis teenindab ligi tuhandet äriklienti üle Baltikumi. Ettevõttel on kasutusel mitu tarkvara, mis omavahel ei ole integreeritud, vaid täidavad ainult ühte kindlat eesmärki või tööloiku. Ettevõttes ei ole eraldi IT osakonda või töötajat, kes tegeleb ainult IT projektidega. Ettevõtte on küll varasemalt tarkvara hankeid teinud, kuid puudu on olnud raamistik mismoodi projekti tehakse. Sellest tulenevalt on tekkinud probleeme nii rollide, vastutuse kui ka hilisema kasutuse osas.

Takistuseks võib ka see tulla, et tihtipeale jääb oskustest või teadmistest puudu, et olla nõ hea tarkvara tellija. Tellitakse teatav funktsionaalsus, aga hiljem selgub, et seda pole vaja või ei ole võimalik rakendada nii, kuidas algselt mõeldi. Teadmisi, kuidas ohjes hoida projekti on eriti vaja, kui ei osteta valmis lahendust, vaid tellitakse süsteemile lisafunktsionaalsust ehk tegemist on sisuliselt rätsepatööga. Seega tuleks suure projekti puhul otsustada, millised kompetentsid ostetakse väljast ja millised kompetentsid on ettevõttes olemas.

1.1 Taust ja probleem

Ettevõttes on eraldi tarkvara tööajaarvestuseks ja objektide halduseks ning kliendiga suhtlemiseks. Sisuliselt peab kasutaja sisse logima mitmesse keskkonda, et andmed kätte saada. Teine probleem tekib sellest, et kuna tarkvarasid on mitu, siis sisestatakse ühe kliendi või objekti kohta andmeid tihtipeale 1- 4 korda. See võib kaasa tuua lisaks topelt koormusele ja ajaressursi raiskamisele ka andmekvaliteedi probleemid, kuna sisestajad on erinevad ja ka süsteemid on erinevad. Ettevõtte kaalub võimalust vahetada välja olemasolev süsteem ning integreerida eraldiseisvad tarkvarad raamatupidamistarkvaraga.

1.2 Ülesande püstitus

Ettevõtte on varasemalt tarkvara hankinud nii valmistoote kui ka eritellimusena, kuid suuremahulise investeeringu jaoks ei piisa pakkumiste võrdlusest ning demode vaatamisest. Soovitud tulemuse saavutamiseks tuleks antud IT süsteemi hankimine teostada projektina. Magistritöö raames on kavas:

- kaardistada hetkeseis ja rahulolu olemasoleva tarkvaraga
- koostada raamistik projekti läbiviimiseks X ettevõttes
- kirjeldada edutegureid projekti läbiviimiseks
- selgitada välja esialgsed nõuded süsteemile

1.3 Metoodika

Magistritöö metoodikaks on intervjuud ja lõppkasutajate küsitlus. Intervjuude ja küsitluse käigus kaardistatakse ootused projekti osas ja süsteemi nõuded. Intervjuu üheks osaks on ka varasema praktika õppetundide analüüsimine ning nendest järelduste tegemine uue raamistiku jaoks. Intervjuu osalisteks valiti osakonnajuhid ja ettevõtte juht just selle tõttu, et saaks erinevate osakondade vajadus ning samas ka tervik kaardistatud.

Intervjuud koosnevad kahest osast :

1. Intervjuu osakonnajuhtide ja ettevõtte juhiga teemal IT süsteemi nõuete ja ootuste kaardistamine. Projekti läbiviimise raamistiku valimine. Arvamused protsessist ja hetkeolukorrast.
2. Intervjuu varasema tarkvara hanke osapoolega. Metoodikaks on juhtumianalüüs.

Intervjuud on plaanis läbi viia osakonnajuhtide ning ettevõtte juhiga. Juhtumianalüüsi jaoks tehakse intervjuu ka varasema tarkvaraprojekti osapoolega. Töö autor on ettevõttes läbi viinud ka ühe tarkvara hanke ning seetõttu on autor lähedalt näinud kitsaskohti tarkvara hankimisprotsessis. Küsitluse valimi moodustavad ettevõtte administratiivtöötajad.

Magistritöö teooria osas on kasutatud viitamisstiilina kogu töö ulatuses nimeviidet.

1.4 Ülevaade tööst

Sissejuhatuses antakse ülevaade taustast ja probleemist ning metoodikast. Peatüki eesmärk on kirjeldada probleemi, mida soovitakse lahendada. Teises peatükis antakse ülevaade tarkvara arendusprotsessist. Peatükis on kirjeldatud tegurid, mida arvestada nõuete määratlemisel.

Kolmas peatükk kirjeldab tarkvara projekti juhtimiseks sobivaid metoodikaid. Vaadeldud on nii PMBOK kui ka PRINCE2 metoodikat ning toodud võrdlus. Neljas peatükk on tarkvara arendusmetoodikate võrdlus. Mõlema peatüki eesmärgiks on välja selgitada milline metoodika oleks sobilikum organisatsiooni ja projekti kontekstis ja mil määral saaks neid rakendada ning kohandada ettevõttele sobivaks.

Viies peatükk on projektimeeskonna planeerimisest. Peatükis kirjeldatakse, milline võiks olla meeskond ja millega tuleks arvestada meeskonna valikul ja suurusel. Kuuendas peatükis on koostatud raamistik, mida kasutada projekti juhtimiseks. Peatükis on analüüsitud intervjuude põhjal metoodikate sobivust organisatsiooni ja projekti kontekstis. Lisadena on toodud küsitluse, juhtumianalüüsi ja intervjuude küsimustik.

2 Ülevaade tarkvara arendusprotsessist

Üha enam ettevõtteid tunnistab, et tõhus tarkvara arendamine - alates ideest kuni kasutuselevõtuni - on konkurentsivõime saavutamiseks otsustava tähtsusega. Kiirendatud tarkvara tarnimise suutlikkuse loomine võib aidata ettevõtetel turul eristuda. (Lesser, Ban 2016, lk 44) Organisatsiooni edu kasumi mõistes, ei korreleeru tingimata sellega, kui palju vara on investeeritud infosüsteemidesse. Põhjuseks on see, et raha, mis on investeeritud, ei too alati loodetud kasutegurit. Väärtusepõhine nõueteanalüüs edendab toote, projekti ja äriotsuste vastavusse viimist ja mitme huvigrupi perspektiivi kaasamist toote, projekti ja ettevõtte väärtuse loomisesse. (Barney *et al.* 2008, lk 577) Seega võib ettevõtte infosüsteem luua konkurentsieelise, kui suudetakse arendada tarkvara, mis loob ka reaalselt ettevõttele väärtust. Tarkvara välja vahetamiseks või lisaarenduste tellimiseks peab olema äriiline vajadus, mis on läbimõeldud.

Alljärgnevalt on Pass ja Ronen kirjeldanud tüüpilisi probleeme (Pass, Ronen 2014, lk 81):

- nõuded ei ole piisavalt defineeritud;
- kulud lähevad liiga suureks;
- väärtuse loomist ei hoita fookuses;
- sisemisi kliente ei kaasata piisavalt;
- ebaefektiivne IT ressursside kasutamine;
- arendatud tarkvara lahendustest loobutakse.

Kõik nimetatud probleemid on autori arvates ka üksikuna piisavad, mille realiseerumisel võib lugeda projekti ebaõnnestunuks. Need probleemid on ka üksteisest sõltuvad. Nõuete ebamäärasel defineerimisel võib projektis ette tulla vigade parandust, mis viib kulude tõusuni. Miks jäävad nõuded defineerimata võib tuleneda ka sellest, et ei kaasata tarkvara lõpptarbijaid. Projekti mõistes kõige ebasoovitavam olukord on, kui loobutakse üldse valminud arendusest.

Zarndt on toonud välja kolm olulist tegurit, miks projekt ei pruugi õnnestuda (Zarndt 2011, lk 171-173):

1. ebapiisav planeerimine;
2. puudulik kommunikatsioon meeskonnaliikmete vahel;
3. halvasti defineeritud vastuvõtu kriteeriumid.

Kindlasti ei tohiks alahinnata sotsiaalseid protsesse meeskonnas ning selle mõju projektile. Seetõttu on ka antud magistritöö raames uuritud meeskonna mõju tarkvara projektile.

Igas arendusprojektis seisavad tarkvaraarenduse juhid silmitsi probleemiga, kuidas tagada, et ressursid oleksid tõhusalt ja tulemuslikult rakendatud, et täita organisatsioonilised vajadused. Kui nõuded on tuvastatud, peab järgmine kaalutus olema disaini kvaliteet, et minimeerida vigade parandamise kulusid. (Morgan 1994, lk 14) Kokkuvõtvalt järeltab autor, et nõuete määratlemiseks peab olema õige sisend ning arusaam, millist probleemi lahendatakse ja kuidas tarkvara seda teha aitab.

2.1 Nõuete analüüs

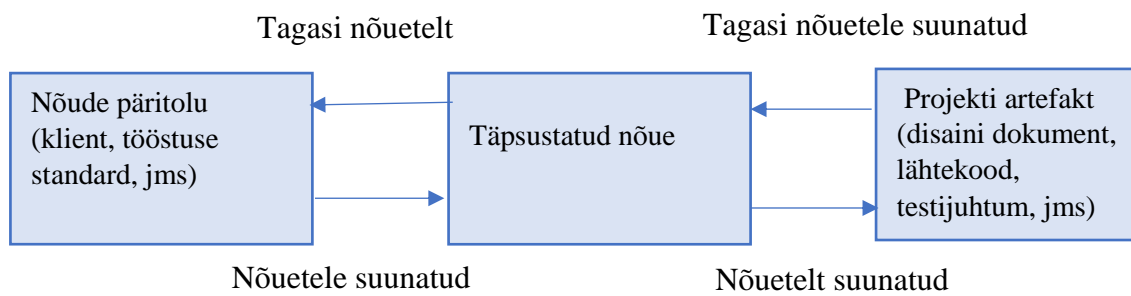
Tarkvara nõuded on projektides ilmnevate probleemide olulised panustajad, sest projektide ebaõnnestumisi võib sageli seostada nõuete muutumisega projekti edenedes. Nõuete täpsele kindlaks tegemisele kulutatud aeg võib oluliselt parandada kogu projekti edu. (White 2013, lk 424) Seega ei tohi seda etappi kiirustades teha, äripoolel peab olema selge arusaam, mida soovitakse infosüsteemilt ja miks. Antud alapeatüki eesmärgiks on välja tuua, millele tuleks tähelepanu pöörata projekti selles etapis.

Hall, Beecham ja Rainer küsitlesid 12 ettevõtet, et uurida milliseid probleeme esineb seoses nõuete kirjeldusega. Uuringus tuli välja, et kõige rohkem (25 %) vastanutest pidas suurimaks probleemiks häguseid nõudeid. 24 % küsitletustest pidas probleemiks ka defineerimata nõuete protsessi ning 23% pidas probleemiks nõuete kasvu. Küsitluse käigus oli ka üldisem küsimus, millega sooviti teada saada arendusprotsessi suuremaid probleeme. Nõuetega seotud probleeme täheldas tervelt 48 % vastajaid. (Hall, Beecham, Rainer 2002, lk 153-155) Autor järeltab, et kui nõuded on häguselt kirjeldatud tekitab see arenduses palju probleeme ning lõppkokkuvõttes ei ole kumbki osapool rahul.

Nõuete tehnilised ülevaated tehakse tavaliselt puuduvate andmete, vastuoluliste või ebarealistlike nõuete tuvastamiseks. See võib olla väga aeganõudev ja kulukas protsess. Mudelite kasutamine hõlbustab kliendi ja analüütikute vahelist koostööd. (Lopez-Lorca *et al.* 2016, lk 610) Funktsionaalsete nõuete kirja panemiseks võib kasutada kasutuslugusid. Kasutuslugude lühikirjelduses peaks olema ära toodud üldine kirjeldus,

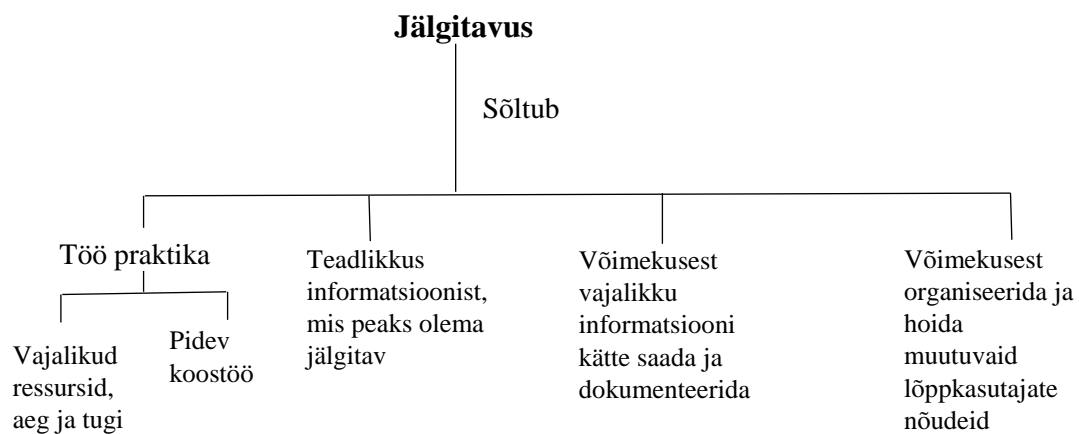
peastsenaariumi kokkuvõte, alternatiivsed stsenaariumid ning eel- ja järeltingimused. (Matveus 2009, lk 29)

Nõude jälgitavust (*traceability*) saab määratleda, kui võimet kirjeldada ja jälgida nõude kestvust. Nõude jälgitavus on oluline just projekti juhtimise ja protsessi nähtavuse seisukohalt. (Kannenberg, Saiedian 2010, lk 4)



Joonis 1. Nõuete jälgitavus (Kannenberg ja Saiediani 2010, lk 4)

Joonisel on näha, et nõudele annab sisendi klient või mõni standard ning väljundiks oleva lähtekoodi aluseks on sama sisend. Järgneval joonisel (Joonis 2) on näha, millest jälgitavus sõltub.



Joonis 2. Jälgitavust mõjutavad tegurid (Gotel, Finkelstein 1994, lk 98)

Hattoni kirjeldatud uuringu tulemused näitasid, et projekti varajases staadiumis on suurim osa nõuete suurenemisest tingitud kliendi otsustest. Tõenäoliselt on see sellepärast, et mõeldakse vaid selle üle, mida süsteem nende jaoks võiks teha. Iteratiivsed meetodid eeldavad, et kliendil ei ole selget ettekujutust täpselt sellest, mida nad projekti alguses tahavad. Kui kliendil on võimalik vahepeal näha, mida toodetak ja ka arendajatega arutleda, saavad nad rohkem ülevaate sellest, mida projektiga on võimalik luua. (Hatton 2008, lk 522) Hatton jaotas nõuete muudatused kolme kategooriasse (Hatton 2008, lk 520):

- kliendi poolt lisatud uued nõuded;
- nõuded mis on varem tähelepanuta jäänud;
- täpsustatud nõuded.

Nõuete kogumiseks on mitu viisi. Üks viis nõuete kogumiseks on intervjuude tegemine. Intervjuul võib küsida, kas lahtiseid või kinniseid küsimusi või lasta lihtsalt kirjeldada ootusi. Lisaks on võimalik kirjeldada ka kasutuslugusid ja stsenaariume. Kolmas viis on kasutaja igapäevase töö vaatlemine ning protsessi selgeks saamine. Kasutada võib ka fookusgruppi, kuhu on kaasatud erineva taustaga inimesed ning koguda kokku nende nägemus, milline funktsionaalsus peaks olema tagatud. (Paetch, *et. al* 2003, lk 1-2)

Kuna üks meetod ei välista teist, on võimalik projekti raames kõiki meetodeid kasutada või kombineerida. Näiteks annab vaatluse ja intervjuude tulemuste võrdlemine parema ülevaate, kui lihtsalt ühe meetodi kasutamine. Samuti on võimalik läbi teise pilgu protsessis parendusi leida. Lisaks eelpoolt kirjeldatud meetodile on võimalus ka funktsionaalsus ette kirjutada ja lõppkasutajad saavad teha valiku. Samas on töö autoril kogemus sellise meetodika kasutamisel, et valitakse põhimõtteliselt kõik variandid ja tegemist ei oleks läbimõeldud otsustega.

2.1.1 Funktsionaalsed ja mittefunktsionaalsed nõuded

Tarkvara töökindlust saab mõõta kahest vaatenurgast: esimene on tehniline ja teine on lõppkasutaja vaatenurk. Tehniline vaatenurk tähendab seda, kas tarkvara kood toimib nii nagu on kavandatud. Kasutaja vaatenurgast lähtumine tähendab uurimist, kas tarkvara vastab kasutaja ootustele ja nõuetele. (Bowen *et al.* 2002, lk 187-188)

Süsteemi mittefunktsionaalsed aspektid, nagu käideldavus ja töökindlus, jäetakse sageli esmase süsteemi kavandamise käigus tähelepanuta. Neid kirjeldatakse eraldi mudelites. Käideldavusega seotud aspektidega peaks tegelema varases süsteemi disaini arendusetapis. (Hassine 2013, lk 1455) Analüüsi teostamine nõuete tasemel toetab vigaste disainilahenduste avastamist tarkvaraarenduse elutsükli algstaadiumis ja aitab vähendada hilisema ümberkujundamisega seotud tegevuste kulusid. Selle eesmärgi saavutamiseks tuleb tarkvaranõudluse faasis lisada mittefunktsionaalsed aspektid. (Hassine 2015, lk 182)

Kokkuvõtvalt järeltab autor, et funktsionaalseid ja mittefunktsionaalseid nõudeid peaks koos üle vaatama. Kui tarkvaras on funktsionaalsus, mis teostab mingit suurt päringut, siis peaks vaatama, kuidas see tarkvara koormab ja mis juhtub jõudlusega. Süsteemi funktsionaalsusest ei ole kasu, kui tarkvara funktsionaalsuse tulemusena süsteem hangub ja lõppeesmärk – mingisuguse toimingu teostamine jääb saavutamata.

Üks aspekt, millega tuleb süsteemi disainimisel kindlasti arvestada on turvalisus, see hõlmab järgmisi samme Jones ja Rastogi põhjal (Jones, Rastogi 2004, lk 33-34):

1. intervjuude läbiviimine, et saada aru ettevõtte turvanõuetest;
2. organisatsiooni turvanõuete kaasamise tagamine;
3. parimate tavade ja standardite kasutamine;
4. reguleerivate privaatsus- ja turvalisuse nõuete, seaduste, poliitika, standardite, välismõjude ja piirangute kindlakstegemine.

2.2 Skoop

Skoobi määratlemine on projekti ja toote üksikasjaliku kirjelduse väljatöötamise protsess. Selle protsessi peamiseks eeliseks on see, et see kirjeldab projekti, teenuse või tulemuse piire, määratledes, millised kogutud nõuded lisatakse projekti skoopi ja millised sellest välja jäetakse. (A Guide to the..., 2013, lk 120) Skoobi valideerimine on lõpule viidud projekti tulemuste heakskiitmise protsess. Selle protsessi eeliseks on see, et see toob aktsepteerimisprotsessile objektiivsuse ja suurendab lõpptoote, teenuse või tulemuste vastuvõtmise võimalust, valideerides iga tulemuse. (A Guide to the..., 2013 lk 133)

Projektijuht on vastutav planeerimis- ja kontrollisüsteemi osana muudatuste süsteemi loomise eest. Projekti alguses tuleks kõigil osapooltel kokku leppida muudatuste

juhtimise protsessis. Määratleda tuleks ka volitatud isikud, kes võivad muudatusi heaks kiita. (Burke 2013, lk 169)

2.3 Lahenduse teostus

Nii nagu ka eelpool kirjeldatud on tarkvara projekti eesmärgiks ettevõttele väärtuse loomine. Organisatsioon peab saama reaalselt kasu arendusest.

Subramanian, Klein, Jiang ja Chan tegid uuringu, kus küsitlesid infosüsteemi spetsialiste, projektijuhte ja IT juhte, et teada saada nende hinnangut 4 faktori prioriteetsuse osas projekti läbiviimisel, projekti teostusele (Subramanian et al. 2009, lk 119). Nendeks faktoriteks ja iseloomustavateks tunnusteks oli (Subramanian *et al.* 2009, lk 119) :

- õppimine – organisatsiooni teadmised tehnoloogia kasutusest, arendustehnoloogiast ja äri funktsionaalsusest;
- kontroll – kontroll projektide kulude, ajakava ja standardite üle;
- efektiivsus – usaldusväärne tarkvara, kiire reageerimisaeg, optimaalsed tegevuskulud;
- paindlikkus – muudatustega efektiivne kohanemine, süsteemi hoolduse efektiivne kulu.

Tulemused näitasid, et IT juhid ja projektijuhid arvasid suure ülekaaluga, et kõige olulisem faktor on kontroll kulude ja ajakava üle. IT professionaalid, kes osalevad projektides, hindasid kontrolli kõrval sama oluliseks tarkvara usaldusväarsust. (Subramanian *et al.* 2009, lk 119-120) Tulemus on autori arvates igati loogiline, sest projekti arendusetapis on suurem vastus ja töö IT juhtidel või projektijuhtidel, kellel on omad eesmärgid.

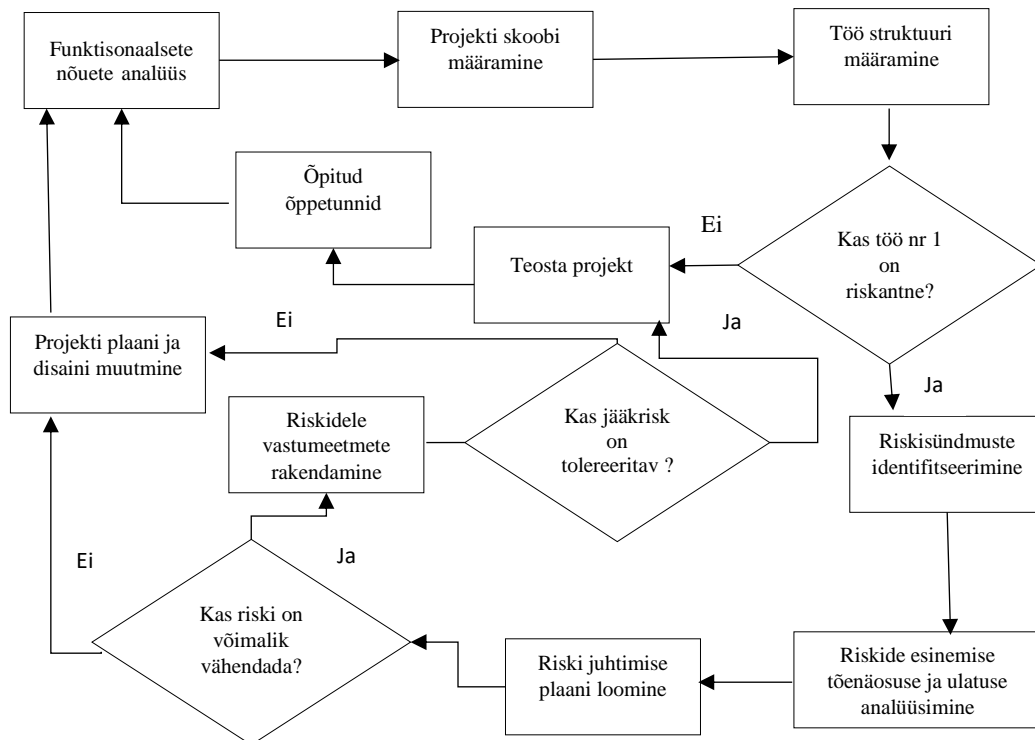
2.4 Riskide juhtimine

Riskid tulenevad sellest, et ei ole võimalik täpselt teada projekti kulgu. Projekti alustades loodetakse, et sellega kaasneb oodatud tulu ning kulud, ei ületa seda, mis oli plaanitud. (Webb 2003, lk 27-28) Riskide juhtimine koosneb riskide kindlaks tegemisest, hindamisest ja prioriseerimisest, mille abil saab organisatsioon juhtida sündmusi, mis võivad projekti edukust mõjutada (Vatsa, Kumar 2016, lk 8).

Dey, Kinch, Ogunlana on riski juhtimise raamistikku kirjeldanud läbi seitsme sammu (Dey *et al.* 2007, lk 290-291):

1. funktsionaalsete nõuete analüüsimine;
2. tarkvaraarendusprojekti skoobi loomine ja tööjaotuse struktuuri arendamine;
3. riskantsete kohtade kaardistamine;
4. riskisündmuste identifitseerimine;
5. riskide analüüsimine;
6. riskijuhtimise plaani koostamine;
7. riski kontrollimine.

Riskide juhtimisel on autori arvates oluline, et plaan ei ole ühekordne vaid kontrollitakse riskide realiseerumise tõenäosust kogu projekti vältel. Kõike riske ei ole võimalik samas ette näha. Väga raske on ette näha meeskonnast tulenevaid riske nagu näiteks projektiliikmete vahetumine või võtmetöötaja lahkumine organisatsioonist. Samuti on sellise riski realiseerumisel ka mõju keeruline ette näha.



Joonis 3. Riski juhtimise mudel (Dey *et al.* 2007, lk 290)

PMBOK-i järgi on riskide identifitseerimine üks põhiprotsesse projektis. Riskide identifitseerimine on iteratiivne protsess, kuna uued riskid tulevad esile samal ajal, kui projekt edeneb. Riskidega on võimalik tegeleda kasutades erinevaid viise. Virine ja

Trumper on toonud välja, et PMBOK järgi on võimalik neljal viisil võimalike ohtudega tegeleda. Esimene võimalus on ära hoida – risk täielikult elimineerida. Teine võimalus on üle kanda – risk kanda üle kolmandale osapoolle. Kolmas võimalus on leevendada, mis tähendab siis riski esinemise tõenäosuse või selle mõju vähendamist. Neljas võimalus on aktsepteerida – kui riski tõenäosus ja mõju on tähtsusetu, siis võib valida strateegia sellega mitte tegeleda. (Virine, Trumper 2008: 119, 123-124)

2.5 Kasutuselevõtt

Ainus viis teada saada, mida osapooled peavad projekti eduks on küsida seda. Eeldused on tihti peale eksitavad. Samas on paljudel inimestel raske kokku võtta selgelt eesmärke, seega tuleks neid aidata projekti edu defineerimisel. (Grey 1995, lk xiii)

Projekti edu saab hinnata pehmete või tugevate kriteeriumite mõjul. Tugevad kriteeriumid on näiteks aeg ja raha, mida on lihtne mõõta. Pehmed kriteeriumid on näiteks rahulolu, mida on keerulisem mõõta, kuna see põhineb subjektiivsel arvamusel. Oluline on vahet teha ka toote edukusel ja projekti juhtimise edukusel. Toode võib hea olla kuid, projektijuhtimine selle loomisel võis olla ebakvaliteetne ning vastupidi. (Albert *et al.* 2017 lk 799, 809)

Tabel 1. Tugevad ja pehmed kriteeriumid (Albert *et al.* 2017 lk 809)

Tugevad kriteeriumid	Pehmed kriteeriumid
Aeg	Ettevõtte rahulolu
Kulu	Juhtide rahulolu
Teostus	Projektiliikmete rahulolu
Kvaliteet	Klientide rahulolu
Majanduslik edu	Lõppkasutajate rahulolu
	Hankijate rahulolu

Millised on edukriteeriumid, peaks autori arvates olema juba projektiplaanis kirjeldatud. Samuti peaks olema määratletud, millised on prioriteedid ja mil määral ja kas ollakse nõus eelarvet või ajalimiiti ületama, et tagada soovitud funktsionaalsus. Kasutuselevõtu juures peab arvestama sellega, et kõigile kasutajatele ei pruugi muudatused meeldida. Muudatuste juhtimine, eriti kui see hõlmab suurt osa organisatsiooni töötajatest, võib olla keeruline, kuna hoitakse veel vanast kinni. Autoril on kogemus organisatsioonis, kui

muudeti süsteemi, kuidas kasutaja oma failidele ligi pääseb. Muudatus oli keerulisem mitte tehnilise teostuse pärast vaid kasutajate ümberhäälestamise tõttu. Üleminek oli aeglasem, kui plaanitud ning ajaline ressurss, mis kulus selgitamisele oli plaanitud pikem.

Tihti peale ei piisa lihtsalt informeerimisest ja juhendist, kuidas edasi toimetada. Kasutajatele tuleb muudatused ka nii öelda maha müüa. Kasutaja vastupanu uue tarkvaraga kohanemisel võib viia projekti ebaõnnestumisele, seega tuleks koolituse osa hoolikalt planeerida.

3 Ülevaade projektijuhtimise metoodikatest

Antud töö raames vaadeldakse kahte tuntud metoodikat PMBOK ja PRINCE2. Ettevõtte, kus projekti hakatakse teostama ei ole kummagi metoodikaga praktilist kogemust, küll aga on töö autoril, kes on ühtlasi ka projekti meeskonnaliige, teoreetilised teadmised metoodikate sisust. Eesmärk ei ole ühtegi metoodikat projektis üks ühele kasutada ega ka metoodikat peensusteni tunda, vaid pigem on toetavaks materjaliks raamistiku loomisel. Enne metoodikate ülevaatus on allpool olevas tabelis kirjeldatud projektijuhtimise nelja etappi ja nende põhiküsimusi.

Tabel 2. Neli etappi projektijuhtimises (Maylor 2003, lk 28).

Faas	Põhipunktid	Fundamentaalsed küsimused
Defineeri projekt	Projekti ja organisatsiooni strateegia, eesmärgi defineerimine	Mida on vaja teha? Miks on vaja teha?
Disani projekti protsess	Planeerimine, ressursside analüüs	Kuidas seda tehakse? Kes on osaline igas etapis? Millal alustada ja lõpetada?
Teosta projekt	Organisatsioon, kontroll, eestvedamine, otsuste tegemine, probleemide lahendamine	Kuidas peaks projekti igapäevaselt juhtima?
Arenda protsessi	Protsessi hindamine, projekti tulemite hindamine, muudatused tulevikuks	Kuidas saab protsessi pidevalt parandada?

3.1 PRINCE2

PRINCE2 (*Projects IN Controlled Environments*) tähendab kontrollitavas keskkonnas olevaid projekte ja seda kirjeldatakse kui struktureeritud meetodit tõhusa projektijuhtimise jaoks. Metoodika on kõikidele projektiliikidele, mitte ainult infosüsteemidele, kuigi selle mõju on metoodikas väga selge. PRINCE2 projekti elutsüklil ei käivitu algse vajadusega, lahenduste genereerimise ja teostatavusuuringutega - neid peetakse projekti elutsükli sisenditeks. Näiteks kirjeldab PRINCE2 toote eluiga, millel on viis etappi: kontseptsioon, teostatavus, rakendamine (või realiseerimine), toimimine ja lõpetamine, kuid neist ainult rakendamine hõlmab

PRINCE2- te. PRINCE2 on rakendamismetoodika, mitte kogu projektijuhtimise metoodika. (Wideman 2002, lk 1,3)

PRINCE2 printsiibid on Yeong põhjal (Yeong 2012):

1. äriiline põhjendatus;
2. õpi kogemusest;
3. defineeritud rollid ja vastutused;
4. etapi kaupa juhitud;
5. halda eranditult - PRINCE2 projektis on määratletud iga projekti eesmärgi suhtes lubatud hälbed;
6. keskendu tootele;
7. kohandage vastavalt projekti keskkonnale.

Autori arvates on PRINCE2 olulisemaks printsiibiks äriiline põhjendus, ehk miks üldse seda projekti tehakse. Kui vajadus ja põhjendus on hägune, siis võib ka eesmärk olla hägune.

3.2 Etapid ja otsustuskohad

PRINCE2 seitse protsessi/etappi on Yeong põhjal (Yeong 2012):

1. projekti algatamise otsustamine (*Starting up a project*)
projekti käivitamise eesmärk on tagada elujõulise ja kasuliku projekti alustamise eeldused;
2. projekti juhtimine (*Directing a project*)
projekti juhtimise protsessi eesmärk on võimaldada projekti juhatusel vastutada projekti edu eest, tehes olulisi otsuseid ja teostades üldist kontrolli, delegeerides samal ajal igapäevase juhtimise projektijuhile;
3. projekti algatamine (*Initiation a project*)
projekti algatamise protsessi eesmärk on luua projektile tugevad alused, mis võimaldavad organisatsioonil mõista tööd, mida on vaja teha projekti toodete tarnimiseks, enne oluliste kulutuste tegemist;
4. projekti faasi/etapi juhtimine (*Controlling a Stage*)
etapi protsessi kontrollimise eesmärk jälgida tehtud tööd, tegeleda probleemidega, teatada edusammudest projekti juhatusele ning rakendada parandusmeetmeid;

5. toote tarnimine (*Managing Product Delivery*)

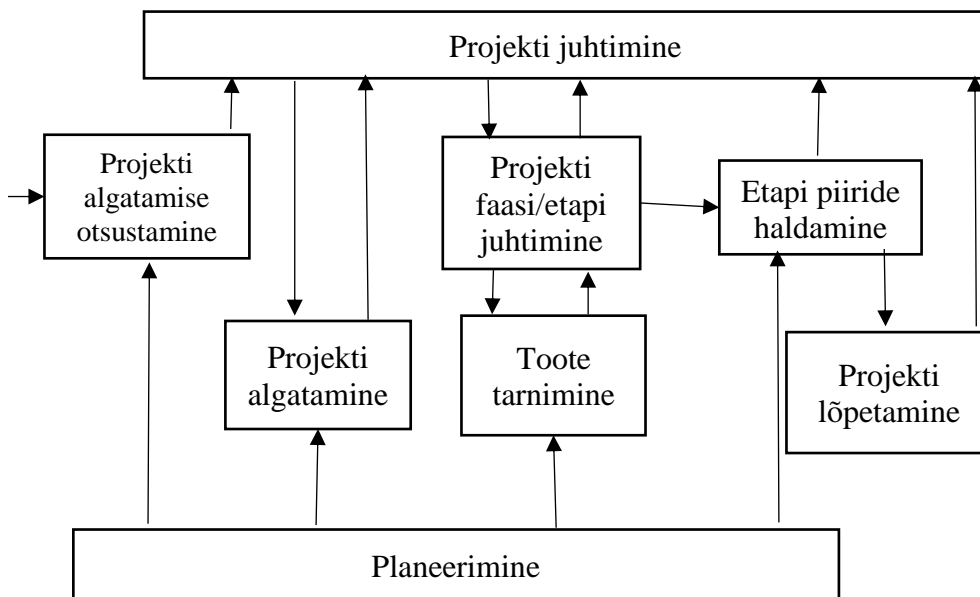
toote kohaletoimetamise protsessi juhtimise eesmärk on kontrollida projektijuhi ja meeskonnaliikmete vahelist koostööd, seades ametlikud nõuded projekti töö vastuvõtmisele, täitmisele ja edastamisele;

6. etapi piiride haldamine (*Managing a Stage Boundary*)

protsessi juhtimise eesmärk on võimaldada projektinõukogul anda projektijuhile piisavalt teavet, et ta saaks vaadata läbi praeguse etapi edu, kinnitada järgmise etapi kava, vaadata läbi uuendatud projekti plaani, kinnitada jätkuvat ärialast põhjendust ja riskide vastuvõetavust;

7. projekti lõpetamine (*Closing a project*)

projekti lõpetamise protsessi eesmärk on pakkuda kindlat punkti, kus projektitoote aktsepteerimine kinnitatakse ning tunnistada, et esialgses projekti algatamise dokumentatsioonis sätestatud eesmärgid on saavutatud (või on saavutatud heakskiidetud muudatused eesmärkides) või, et projektil pole midagi enam panustada.



Joonis 4. PRINCE2 protsessimudel Siegelaub põhjal (Siegelaub 2004, lk 3)

Jooniselt on näha, et projekti alustala on planeerimine. Planeerimisest saab alguse projekt ning planeerimise käigus määratletakse etapi piirid.

3.3 PMBOK

PMBOK on projektijuhtimise kutseala tunnustatud standard. Standard on ametlik dokument, mis kirjeldab kehtestatud norme, meetodeid, protsesse ja tavasid. PMBOK annab juhiseid üksikute projektide haldamiseks. See määratleb projektijuhtimise ja sellega seotud mõisted ning kirjeldab projektijuhtimise elutsükli ja sellega seotud protsesse. (Project Management Body of Knowledge...2019)

PMBOK võimaldab projektijuhtidel töötada ettevõtetes standardiseeritud süsteemiga. Keegi, kes töötab ettevõttes x, kes seejärel liigub ettevõttesse y, võib kasutada samu tavasid. (What is PMBOK.... 2019)

3.4 Etapid ja otsustuskohad

PMBOK- i etappe on kokku neli ja need jagunevad (What is PMBOK...2019):

1. algatamine – tööetappide loomine, meeskonna moodustamine esialgse skoobi ja eelarve paika panemine;
2. planeerimine – planeeritakse detailsemalt töökorraldus, skoop ja eelarve, määratakse tööplaan ja prioriteedid;
3. teostamine – selles etapis toimub projekti eesmärkide elluviimine; monitoorimine ja kontrollimine – tegeletakse muudatuste töötlemisega, käimasolevate eelarvekaalutluste käsitlemise ja ettenägematute olukordade lahendamisega, mis võivad mõjutada meeskonna võimet täita projekti esialgseid ootusi;
4. lõpetamine – selle etapi suurimaks väljakutseks on viia projekt edukalt lõpule, mis tähendab, et see viiakse lõpule õigeaegselt ja eraldatud eelarve piires.



Joonis 5. PMBOK Etapid (Project management fundamentals...2019)

Selline etapiline käsitlus on autori arvates hea juhend struktuuri loomiseks. Tarkvaraprojekti kontekstis tuleks alustada projekti üldkirjeldusega ning koostada ka hinnanguline eelarve projektile. Edasi saab liikuda detailsema kavaga, kus on paika pandud skoop, kohendatud eelarvet ja määratud sobilikud ressursid sealhulgas meeskond. Meeskonna moodustamist on kirjeldatud täpsemalt viiendas peatükis. Projekti teostades peavad toimima ka kontrollimehhanismid, mis hoolitseksid selle eest, et fookus hoitakse skoobil ja eesmärgil ning seejuures ei ületada eelarvet.

3.5 Metoodikate võrdlus

PRINCE2 on tugev protsessis ja dokumentatsioonis, kuid puudub keskendumine kommunikatsioonile, personalijuhtimisele, mida PMBOK suudab täiendada. Teisest küljest tugevdab PRINCE2 ärijuhtumipõhist lähenemisviisi projektijuhtimisele, milles PMBOK on nõrk. PMBOK on suunatud rohkem kliendikesksusele, rõhutades kvaliteetsete toodete tarnimist kliendile õigeaegselt projekti eelarve raames. PRINCE2 ja PMBOK- i ei tohiks vaadelda projektide haldamisel konkureerivatena. Nad ei tohiks olla üksteist välistavad. Kombineeritud lähenemisviisid võiksid pakkuda kliendikeskseid ja kvaliteetseid tooteid, mis vastavad organisatsiooni ärivajadustele või äritegevusele. (Yeong 2012)

Tabel 3. PMBOK ja PRINCE 2 võrdlus Siegelaub põhjal (Siegelaub 2004, lk 2)

PMBOK teadmiste valdkond	Võrreldavad PRINCE2 komponendid
---------------------------------	--

Integratsioon	Ühendatud protsessid ja komponendid, muudatuste kontroll
Skoop, aeg, kulu	Plaanid, ärijuhtum
Kvaliteet	Kvaliteet, konfiguratsiooni juhtimine
Risk	Risk
Kommunikatsioon	Kontroll
Personal	Organisatsioon
Hanked	Ei ole hõlmatud

Karaman ja Kurt on toonud välja veel 4 erinevust PMBOK ja PRINCE2 vahel (Karaman, Kurt 2015, lk 578):

- PMBOKil on tugevam mehhanism protsesside integreerimiseks;
- organisatsiooniliste protsesside varad ja keskkonnategurid on paremini integreeritud PMBOKi protsessidega;
- PMBOKil on tugevam kommunikatsiooni juhtimise mehhanism;
- PRINCE2 hõlmab toote tarnetegevuse juhtimist projekti meeskonna vaatenurgast.

Väikeste IT-projektide puhul on eelistatud PRINCE2. PMBOK on eelistatav IT-projektide puhul, millel on suured ja keerulised projektimeeskonnad ning kõrge allhanke tase. (Karaman, Kurt 2015, lk 578)

Hughes, Rana, Simintiras on artiklis välja toonud, et organisatsioonid ei saa eeldada, et sobiva projektijuhtimise meetodika või standardite kogumi valik toob kaasa eduka tulemuse. Kuidas teatud meetodikat rakendada on võtmetähtsusega, võttes arvesse organisatsiooni kultuuri ja kogemuste laiemaid aspekte. Meetodit tuleks kohandada vastavalt projektile ja organisatsioonile. (Hughes *et al.* 2017, lk 159)

4 Arendusmetoodika valik

Alljärgnevides alapeatükkides on kirjetatud nii agiilset arendusmetoodikat kui ka traditsioonilist koskmetoodikat. Viimane alapeatükk on metoodikate omavahelisest võrdlusest ning sobivusest erinevatest projektides.

4.1 Agiilsed meetodid

Mõiste agiilne tähistab kiiret liikumist. See tarkvaraarenduse metoodika põhineb tarkvaraarenduse iteratiivsel mudelil. Selle mudeli kõige olulisem põhimõte on klientide rahulolu, pakkudes tarkvara kiiresti ja pidevalt. Tarkvara kättetoimetamine toimub korrapäraste ajavahemike järel. Kuna tarkvara on välja töötatud väikestes partiides, saab tarkvara paindlikumalt muuta. (McCormick 2012, lk 7)

Paljud organisatsioonid on liikunud agiilsete metoodikate peale. See on tingitud vajadusest toota paremaid, kiiremaid ja kulutõhusamaid tarkvaralahendusi ning säilitada samal ajal töötajate tööga rahulolu kõrge tase. Taleb ja Hajjdiabi juhtumianalüüsis toodi välja, mis on agiilsete meetodite rakendamise väljakutsed. Üheks väljakutseks oli see, et igapäevased koosolekud pingestasid meeskonda, kuna töötati ranges ajagraafikus. Koosolekuid hakati harvem pidama ning see mõjutas õppimisprotsessi. See viis lõpuks ebaõnnestumise ja agiilse meetodi ebaõige rakendamiseni. (Hajjdiab, Taleb 2011, lk 2,7) Siit saab järeldada, et õppimisprotsess on agiilse meetodi üks olulisemaid aspekte ning kuigi koosolekud võtavad aega, siis ei saa neid ära jätta.

Tagasivaatamine on agiilne praktika, mis võimaldab meeskondadel kontrollida oma praegust olukorda, õppida kogemustest ja teha plaane tulevaste iteratsioonide kohandamiseks õppimise alusel. (Hoda *et al.* 2013, lk 97) Põhimõtteliselt hõlbustab peegeldava praktika juurutamine tarkvaraarenduses organisatsioonile õppesüsteemi, kus ükski konkreetne tarkvaraarendusmeetod ei ole nii oluline, kui võime kohandada ja arendada meeskonnas vajalikke teadmisi. (Hoda, *et al.* 2014, lk 56) Autor järeldab, et agiilse metoodika puhul peaks analüüsima, kas organisatsioon on selleks valmis.

4.2 Koskmudel

Koskmeetod on lineaarne protsess, mis koosneb järjestikustest arengufaasidest. Arengufaasid hõlmavad endas süsteemi teostatavusuuringut, nõudluse analüüsi ja projekti planeerimist, süsteemi kavandamist, kodeerimist, testimist ja integreerimist, paigaldamist ja hooldust. Iga etapp dokumenteeritakse aruandes, mis tuleb enne järgmise etapi juurde liikumist ametlikult kinnitada. Koskmeetod sobib pigem rutiinset tüüpi projektidele, kus nõuded on hästi määratletud. (Khalifa, Verner 2000, lk 360-361)

Alljärgnevalt on kirjeldatud koskmeetodi sobivust ja meetodi kitsaskohti VanderLeest, Buteri põhjal (VanderLeest, Buter 2009, lk 2):

- koskmeetod sobib vähese muutusega projektide, vähese uudsuse ja vähese keerukusega projektide jaoks;
- koskmeetod paneb suure riskiga keerulised elemendid projekti lõppu;
- koskmeetod suurendab keerukust;
- koskmeetod ei tule toime nõuete muudatustega;
- koskmeetod annab ebausaldusväärseid ajakava hinnanguid.

Metoodika valiku jaoks peaks seega olema arusaam ettevõtte X kontekstis, milliseks kujuneb projekti skoop. Keerukus sõltub sellest, kui palju saab valmistoodet kasutada ja kas ja kui palju on üldse vaja lisafunktsionaalsust tellida.

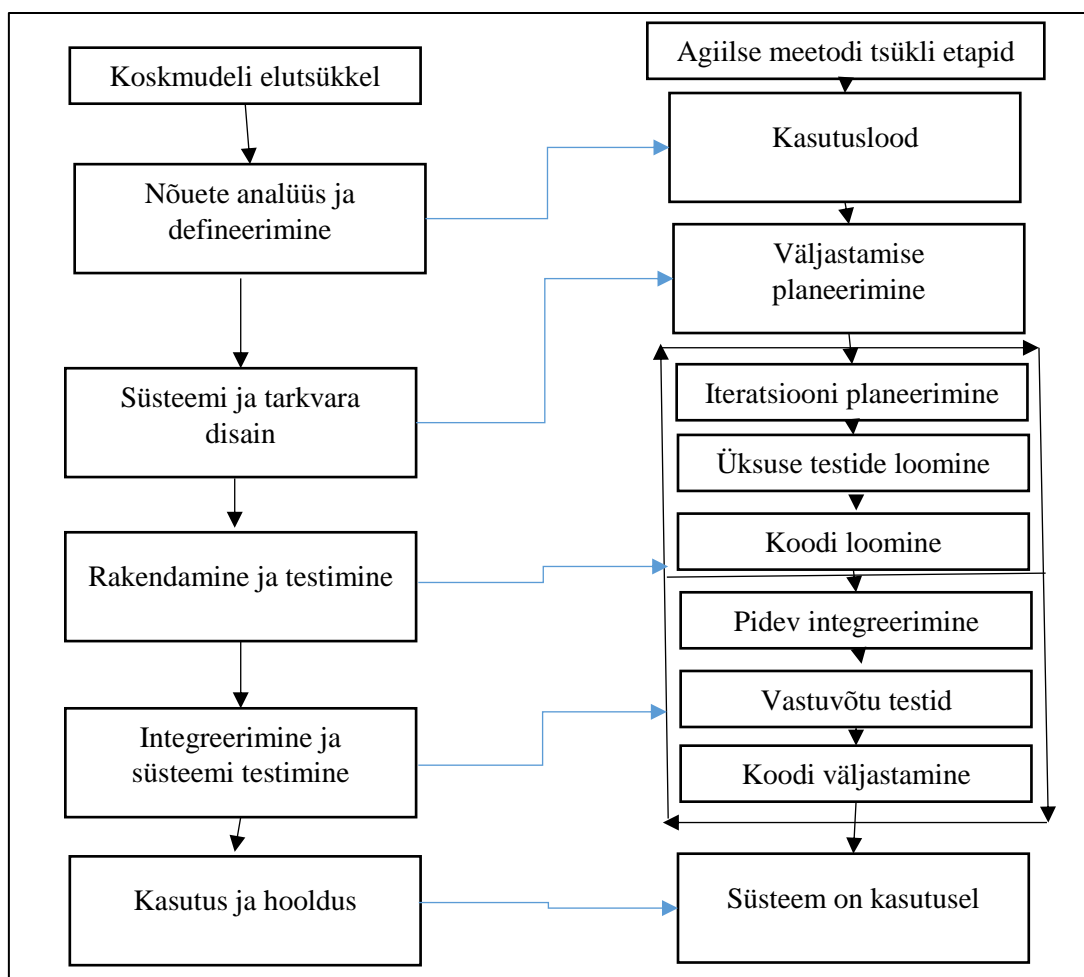
4.3 Meetodite kasutus

Koskmudeli arendusmetoodika rõhutab tavaliselt rohkem kontrolliaspekte ja tõhusaid tegevusi, samas kui agiilse arenduse pooldajad otsivad rohkem õppeprotsessi ja paindlikku toodet. Sõltumata sellest, millist arendusmetoodikat järgida, peavad projektijuhid süsteemi õigeaegselt, eelarve piires ja kasutaja nõudeid täites edastama. (Subramanian *et al.* 2009, lk 118)

Koskmudeli metoodikas osalevad kliendid tavaliselt nõuete määratlemisel, kuid nad ei osale nii palju kui agiilse meetodi puhul. Agiilse meetodi puhul on klientide kaasamine palju suurem, kui koskmeetodi puhul, sest arvestatakse pideva tagasisidega. Koskmeetodi puhul teostatakse ka vastuvõtuteste, kuid neid ei tehta nii tihti kui agiilse meetodi puhul.

Varajane kliendi tagasiside on agiilse meetodi üks väärtuslikumaid omadusi. (Huo *et al.* 2004)

Võrreldes traditsiooniliste meetoditega (nagu näiteks koskmeetod) on agiilsetes meetodites oluliselt vähem dokumente. Dokumentide vähesus muudab lihtsamaks neid ajakohasena hoida. Dokumentide ja muude selgesõnaliste teadmiste vähendamise kompenseerimiseks ergutavad agiilsed meetodid otsest ja sagedast suhtlemist ning koostööd. Agiilsete meetodite rõhuasetus on inimestele, suhtlusele ja koostööle. Samuti edendavad nad teadmiste jagamise meeskonda, vastastikust usaldust ja hoolt. (Chau *et al.* 2003)



Joonis 6. Koskmeetodi ja Agiilsete meetodide etappide võrdlus (Huo *et al.* 2004)

Balaji ja Murugaiyan tegid võrdluse erinevate meetodite vahel ning nad jõudsid järeldusele, et kui on suur projekt, mille nõuded on selged, siis valik on koskmeetod. Kui nõuded muutuvad tihti ja projekt on väiksem, siis jääb valikuks agiilne meetod. (Balaji, Murugaiyan 2012, lk 29) Agiilse ja koskmeetodi valiku kriteeriumil võib arvestada ka aja

faktoriga. Kui ajaraamistik on piisavalt pikk võib koskmeetodi valida, kui on vaja kiirelt tarkvara, siis on agiilne meetod parem. (McCormick 2012, lk 4)

5 Projektimeeskonna planeerimine

Meeskonna moodustamisel tuleb arvestada, et valitakse töötajaid, kes lisaks ettevõtte äriprotsessidele teavad ka levinud parimaid praktikaid (Carton *et al.* 2008, lk 121). Altiner ja Ayhan tegid uuringu, mille eesmärgiks oli aru saada, kuidas meeskonna mitmekesisus mõjutab projekti edu. Altiner ja Ayhan valisid neli tegurit mõju hindamisel: vanus, sugu, haridus ja kogemus. Kõige rohkem positiivset mõju efektiivsusele oli erineva haridusliku taustaga inimeste kaasamisel. Projekti meeskonna juhid peaksid meeskondade moodustamisel või uute töötajate töölevõtmisel kaaluma erinevaid mitmekesisusi, et suurendada meeskonna tõhusust. Lisaks peaksid juhid looma oma töötajatele erinevaid koolitusvõimalusi. Erineva haridustasemega töötajad saavad ettevõttele tuua uusi kompetentse. Kuigi tööjõu homogeensus võib tagada sujuvama töökeskkonna, suurendab meeskondade suurem mitmekesisus nende tõhusust. (Altiner, Ayhan 2018, lk 69-76)

Liang, Lin ja Lin analüüsisid teise vaatenurga alt meeskonna mitmekesisuse mõju tarkvara projekti edukusele. Uuriti teadmiste mitmekesisuse ja väärtuste mitmekesisuse ning sotsiaalse mitmekesisuse mõju projekti edukusele. Tulemused näitasid, et erinevad mitmekesisuse vormid tekitavad erinevaid konfliktivorme, mis omakorda mõjutavad meeskonna jõudlust. Teadmiste erisus mõjutab positiivselt meeskonna jõudlust, mõjutades selle mõju ülesannetele ja suhetele. Väärtuste erisus võib kahjustada meeskonna jõudlust suhetes. Sotsiaalne mitmekesisus mõjutab nii ülesande konflikti kui ka suhteid ja mõju võib olla nii positiivne kui ka negatiivne. (Liang *et al.* 2007, lk 649) Autor järeldeb, et negatiivne mõju projektile tuleb väärtuste konfliktist ning selle ületamine on keeruline, kuna väärtusi ei saa lihtsalt ümber õpetada.

Arusaamine meeskonna suuruse mõjust tarkvaraarenduse kuludele on oluline, sest enamik juhte, kellel on ranged tähtajad, on silmitsi kiusatusega suurendada meeskonda (Pendharkar, Rodger 2009, lk 141). Meeskonna suuruse suurenemise põhjuste hulka kuuluvad Pendharkar ja Rodgersi põhjal: (Pendharkar, Rodger 2009, lk 141)

- ajakava täitmine, kuna tehnoloogia konkurentsieelis aja jooksul väheneb;
- tööd on võimalik korraldada väikestes allprojektides, mis vähendab projekti valmimise aega.

Pendharkari ja Rodgersi uuringus selgus, et meeskonnaliikmete kasv ei suurenda lineaarselt kulude kasvu. Meeskonnaliikmete arvu valimisel peaks projektijuht hoolikalt vajaduse läbi mõtlema, kuna meeskonna suuruse mõju on raske hinnata. (Pendharkar, Rodger 2009, lk 144) Probleeme võib tekkida siis kui üks või mitu projektiliiget läheb ära. Telioglu ja Wagner kirjeldasid ühte tarkvara projekti, kus võtmeisik – pearendaja lahkus ettevõttest projekti keskel. Ühine nägemus, sealhulgas arusaam, et igauks õpib, oli sõltuvuses arendaja tugevast tahtest ja aeg, mil ta ära läks, see nägemus lagunes ka koost. (Telioglu, Wagner 1997, lk 259) Autor järeldeb, et meeskonna püsimine projekti vältel võib projekti kulgu oluliselt mõjutada.

Alati ei ole tarkvara arenduses projektijuhti ja tekib nii öelda enesekorralduslik meeskond. Enesekorralduslikud meeskonnad koosnevad üksikisikutest, kes juhivad oma töökoormust, vahetavad tööd omavahel tuginedes vajadusele. Iseorganiseeruvatel meeskondadel peab olema ühine fookus, vastastikune usaldus ja võime korduvalt uusi väljakutseid lahendada. (Hoda *et al.* 2013, lk 422) Ilma otsese juhita ja koordineerimata projektid sobiksid autori arvates pigem väiksema mastaabiga tarkvaraarenduste puhul. Juhul, kui on vaja luua ettevõttele terviklik infosüsteem, siis projektijuht on vajalik, et hoida silma peal skoobil, eelarvel, tähtajal ning ressurssidel.

5.1 Sotsiaalsed protsessid tarkvara arendusprojektides

Projektimeeskonnas peaks olema usaldus meeskonnaliikmete vahel. Igasuguse projekti tulemusi mõjutavad osapoolte esialgne kavatsus ja ootused. Kui suhe algab usalduseta kavatsusest, on projekti dünaamika selline, et projekt võib ebaõnnestuda. Ebaõnnestunud tulemus tekib, sest osapooled ei soovi liikuda ühise eesmärgi suunas, mis on usalduse loomulik tagajärg. (Brewer, Strahorn 2012, lk 293)

Meeskondadel võib olla erinevaid viise ja nüansse töö korraldamiseks ja kommunikatsiooniks. Sawyer' i järgi on kolm arhetüüpi, mis kirjeldab sotsiaalseid protsesse tarkvara arendusmeeskonnas (Sawyer 2004, lk 96) :

1. järjestus – tarkvaraarendus on tootmistegevus, mis põhineb lineaarsel ülesannete komplektil ning inimesed töötavad spetsialiseeritud funktsioonides ja neid hinnatakse nende oskuste eest;
2. grupp – tarkvaraarendust peetakse arendamise ja tootmise kombinatsiooniks, kus võidakse ülesandeid korrata, kuni toode on valmis ning arendajad on organiseeritud

üksteisest sõltuvasse rühmadesse ja neid hinnatakse nii nende erioskuste kui ka võimetest teiste meeskonna liikmetega koostööd teha;

3. võrgustik – tarkvaraarendust peetakse pideva arengu protsessiks, keskendudes eelkõige tulemusele/tootele, ülesanded ei ole järjestikused ning on seotud üksikisikutega või väikeste gruppidega ning grupi liikmeid hinnatakse selle eest, mida nad saavad toota.

Järjestuse arhetüüp kinnitab veendumust, et hea protsess toob kaasa hea toote. Tarkvaraarendust peetakse lineaarseks, ülesannetega struktureeritud pingutuseks. Tarkvaraarenduse järjestuse arhetüübi sotsiaalsed suhted põhinevad kontrolli mõistetil. Selline ülesandele orienteeritus viitab ka sellele, et kui tekib vajadus meeskonnaliikme vahetumiseks, siis see on võimalik, põhiline on see, et asendaja oleks samad funktsionaalsed oskused. Grupi arhetüübis põhineb tarkvaraarendus rida etteantud ülesannete lahendamisel, mis tuginevad grupi liikmete kollektiivsetele oskustele ja nõrkustele. Grupi arhetüübi sotsiaalsed struktuurid põhinevad koostööl. Ülesanded on järjestikused, kuid iteratiivsed ning rühma liikmed on selgelt pööranud tähelepanu protsessi parandamisele. Toode on võrgu arhetüübi keskmes; tootmisprotsessid on teisejärgulised. Arendustegevus kujuneb osalejate poolt välja töötatud võrguühenduste kaudu. Võrgu arhetüübi aluseks olev usk on see, et hea toode on pärit headest inimestest. See lähenemisviis tunnistab, et võrgu võtmeisikute asendamine on raske, kui mitte võimatu, sest nad esindavad olulisi sõlmpunkte. (Sawyer 2004, lk 96-97) Eelpool kirjeldatud kolm arhetüüpi on väga erinevad üksteisest ning meeskonda võib sattuda inimene, kellele ei sobi töökorralduse protsess või kes ei sulandu meeskonda, kuna isiku tööstiil on erinev.

Ruuska ja Vartiaineni juhtumianalüüsis tuli välja, et kommunikatsioon ei ole oluline mitte ainult projektimeeskonna sees, vaid ka organisatsiooni ja projektimeeskonna vahel. Ruuska ja Vartiainen toovad ka välja, et projektimeeskonna liikmed peavad nägema suurt pilti ning jagama ühist eesmärki. (Ruuska, Vartiainen 2003, lk 310-311)

Crawford ja Leonard tegid uuringu, mille eesmärgiks oli analüüsida erinevate tegurite mõju koosoleku efektiivsusele ja pärast koosolekut järgnevatele tegevustele. Uuringus oli 5 hüpoteesi, millest esimesed kolm oli meeskonna mitmekesisuse kohta (funktsionaalne mitmekesisus, kogemus ettevõttes, ettevõtte töötaja versus lepinguline). Neljas ja viies hüpotees oli vastavalt meeskonna liikmete arv, mis koosolekul oli ning eelmiste koosolekute mõju. Uuringus selgus, et positiivset mõju tegevustele omavad mitmekesine

kogemus ettevõtte sees ja eelmiste koosolekute toimumine. See on ka loogiline, kuna efektiivsuse saavutamine ja hea meeskonnatöö võtab aega. Aja jooksul võivad sama projektiga kohtumised suurendada tootlikkust, kuna grupi dünaamika muutub prognoositavamaks ja stabiilsemaks. (Crawford, Leonard 2012, lk 59-73) Crawfordi ja Leonardi uuringust võib järeldada, et regulaarsed koosolekud tagavad parema meeskonnatöö. Koosolekute ja meeskonnakohtumiste plaan võiks olla juba projekti alguses paigas.

5.2 Meeskonnaliikmete mõju kuludele

Tarkvaraarenduse algfaasis on projekti suurust raske ennustada, kuna nõuded on ebatäpsed ja mittetäielikud ning arendustsükli käigus võivad need ka muutuda. Agiilne tarkvaraarendus on muutumas üheks sagedamini kasutatavaks tarkvaraarendusvahendiks, mida laialdaselt kasutavad nii erinevad teadlased kui ka tarkvaraarenduse organisatsioonid. Kulude ja jõupingutuste täpne hindamine tarkvaraprojektis on projekti edukuse seisukohalt olulise tähtsusega. (Bilgaiyan *et al.* 2017, lk 59)

Krishnan (1998) tegi uuringu, et saada teada, mil moel meeskonna koosseis mõjutab kulusid. Analüüs näitab, et meeskonna personali võimekus on märkimisväärselt seotud defektide arvu vähenemisega ja tarkvaratoodete suurema tootlikkusega. Personali kasutamine, kellel on kõrgem kogemus konkreetse tarkvaravaldkonnaga võib tagada väiksema vigade arvu toodete tarnimisel. Siiski ei pruugi personali kasutamine, kellel on suurem arv aastaid kogemusi konkreetse programmeerimiskeeles, aidata kaasa kõrgema tootlikkuse saavutamisele või defektivaba toote tarnimisele. (Krishnan 1998, lk 32-33)

5.3 Vastutus ja rollid

Selles peatükis on kirjeldatud projektimeeskonna rolle ja vastutust. Zwikael ja Meredith jagasid projekti olemuse kaheks: rahastamine (*funding*) ja täitmine (*performing*). Nii rahastamise kui täitmise all oli viis rolli ja vastutust. Alljärgnevalt on esitatud tabelis põhiideed. (Zwikael, Meredith 2018, lk 485)

Tabel 4. Projektijuhtimise rollid (Zwikael ja Meredith 2018, lk 485)

Olemus	Projekti roll	Rolli definitsioon ja vastutus	Peamised tulemuslikkuse kriteeriumid
--------	---------------	--------------------------------	--------------------------------------

Rahastus	Rahastaja	Projekti rahastaja ja ressursside määraja	Kvaliteetne investeerimisotsus
	Meister	Ärijuhtumite (<i>business case</i>) välja selgitamine	Kvaliteetsed ärijuhtumid (business case)
	Projekti omanik	Kõrgem juht, kes vastutab ärijuhtumi realiseerimise eest	Ärijuhtumite realiseerimine
	Projekti juhtkomitee	Strateegiline üksus, mis tagab projekti edenemise vastavalt plaanile ja äritegevusele	Ärijuhtumite realiseerimine
	Kasutaja	Üksus, kes kasutab projekti tulemusi.	Tõhus väljundi kasutamine
Teostus	Projekti juht	Vastutab projekti tulemuste eest vastavalt heakskiidetud projekti kavale	Projekti plaani realiseerimine
	Projekti meeskond	Projekti väljundeid tootev meeskond	Projekti plaani realiseerimine
	Programmi juht	Isik, kes töötab mitmete projektidega. Ei ole vajalik kui tegu on eraldiseisva projektiga	Programmi plaani realiseerimine
	Sponsor	Kõrgem juht, kes toetab projekti ja pakub toetust projektijuhile ja nende meeskonnale	Projekti plaani realiseerimine
	Projektijuhtimise büroo	Grupp, mis tavaliselt standardiseerib projektijuhtimisprotsesse ja toetab sama osakonna, asutuse või äriüksuse projektijuhte sellistes valdkondades nagu projektijuhtimise meetodid, vahendid ja tehnikad.	Kõikide projekti plaanide realiseerimine

Autori arvates ei pea iga rolli jaoks olema eraldi inimene või inimesed. Kõik oleneb projekti suurusest ja organisatsioonist ja pigem tuleks tabelit kohandada vastavalt vajadustele. Oluline, mida siit tabelist võtta on vastutus ja ootused rollile. Ettevõttes planeeritava projekti jaoks ei ole tingimata vaja programmi juhti ega ka projektijuhtimise bürood.

Analüüsid eelnevalt viidatud autorite uurimusi, siis meeskond peaks koosnema erinevate kompetentsidega töötajatest, kellel on ühised arusaamad projekti toimimisest. Meeskonnatöö parandamiseks peaks toimuma regulaarsed koosolekud, kus arutatakse hetkeseisu ning seatakse uued eesmärgid järgnevateks etappideks.

6 Näidisprojekt ettevõttes X

Ettevõtte X on viimase nelja aasta jooksul hankinud kaks tarkvara ning lisaks arendusi olemasolevale raamatupidamissüsteemile. Kahest tarkvarast üks osteti valmislahendusena ning teine tarkvara hangiti nii, et põhifunktsionaalsus oli olemas aga aasta jooksul arendati veel ettevõttele X sobilikku lisafunktsionaalsust. Ettevõtte on loobunud ühest varasemast tarkvarast ning ka ühest hiljuti hangitud tarkvarast. Seega on protsessis, kuidas tarkvara hangiti, kindlasti parendusruumi, kuna tarkvara hankimine on ressursirikas, ning tihtipeale on just suuremad kulud (mida on raske mõõta) seotud koolituse ja juurutamisega.

Tarkvara on varasemalt hangitud sisuliselt osakonnapõhiselt. Osakonnajuht on kaardistanud, kas on vaja tarkvaralist muutust ja on vastavalt vajadusele ostnud valmistarkvara või tellinud arenduse olemasolevale. Selline korraldus on tekitanud olukorra, kus ettevõtte tarkvara on killustatud ning ühtset toimivat süsteemi ei ole. Uus tarkvara on lahendanud ainult teatud grupi huvid, kuid ettevõtte vajadusi tervikuna see ei toeta.

6.1 Juhtumianalüüs

Juhtumianalüüsiks on valitud tarkvara, mille eesmärk oli tööaja kaardistamine. Kuna ettevõttes on üle 2000 töötaja oli vaja rakendust, millega oleks võimalik tööaega fikseerida ning graafikuid valmis teha. Tarkvara oli mõeldud ka sisemiseks värbamiseks. (Juhtumianalüüsi intervjuu 2019) Alljärgnevalt on tehtud kokkuvõtte intervjuust ettevõtte juhiga, kes oli ka projekti juurutamisega seotud.

Ettevõtte juht nägi probleemi tarkvara hankeprotsessis, selles et tarkvara ei suudetud kasutajatele maha müüa ehk selle vajalikkust ei suudetud tõestada. Organisatsioonis ei olnud kedagi, kes tarkvara nii hästi tunneks ning majasisene tugi kasutajatele ei olnud piisav. Tarkvara projekti jaoks oli meeskond, kuid otsest vastutajat või tootemanikku ei olnud. Samuti koolitused küll tehti, kuid abistav kirjalik juhendmaterjal kasutajatele, tuli liiga hilja. Samas ei näinud ettevõtte juht selles, et korralik juhendmaterjal puudus, projekti läbikukkumises juurpõhjusest. Põhjus oli pigem see, et juurutamisel ja koolitusel tehti vigu ja seetõttu kasutajad olid ebakindlad tarkvara kasutamisel. Põhiline viga oli see,

et tarkvara üritati väänata ettevõtte protsesside jaoks mitte vastupidi. Protsessis oldi liiga kinni. (Juhtumianalüüsi intervjuu 2019)

Intervjuus küsisin kui palju tehti ka tehnilist eeltööd uurimaks, kas hangitavat tarkvara saaks liidestada ka raamatupidamistarkvaraga. Vastus oli, et eeltööd ei tehtud. (Juhtumianalüüsi intervjuu 2019) See on ka autori arvates üheks põhjuseks, miks projekt ei õnnestunud. Kasutajad pidi nüüd andmed panema kahte süsteemi. Mõlemasse tarkvarasse sisestati graafikuid, kuid uut tarkvara ei olnud võimalik liidestada raamatupidamistarkvaraga, seega jäi paralleelselt kaks tarkvara ning uue tarkvara puhul kasutati teisi funktsioone. See tekitas autori arvates kasutajates rahulolematust, kuna ei saadud tarkvara kasulikkusest aru ja kõik muu funktsionaalsus oli teisejärguline.

Juhtumianalüüsi intervjuu käigus küsisin, ka milline oli õppetund ja mida saaks teha teisiti ning ettevõtte juht arvas, et raskuskoht on planeerimisel, enne kui alustada peaks teadma miks ja kuidas. Oluline on, et ei oleks protsessis kinni. Nii palju standard lahendusi kui võimalik kasutada. (Juhtumianalüüsi intervjuu 2019)

6.1.1 Kokkuvõtted intervjuust

Osakonnajuhtidelt ja ettevõtte juhilt küsiti samasugused küsimused ning nende eesmärk oli hinnata praegust tarkvara ja varasemat tarkvara hanke protsessi, saada mõtteid, kuidas läbi viia uus tarkvara projekt ning kaardistada ka esialgsed nõuded. Alljärgnevalt on tehtud kokkuvõtte intervjuudest autori märkmete põhjal.

Esimese küsimusena oli küsitud kas küsitluse tulemusena välja tulnud pigem hea rahulolu mõjutab kuidagi tarkvara projekti otsust ning kõik vastasid, et ei mõjuta. Põhjenduseks toodi, et lõppkasutaja ei oska võibolla hinnata kõige paremini protsesside efektiivsust. Pigem fikseeris see praeguse seisuga. Intervjuust osavõtjad tundsid tarkvara puhul kõige rohkem puudust aruandlus ja analüüsi funktsioonist, samuti toodi miinuspoolena erinevate tarkvarade paljusus. Küsimusele millist väärtust tarkvara looks ja millist ärilist probleemi aitaks lahendada vastati, et kindlasti oleks võit ajas, sest äriotsusteks oleks võimalik kiiremini info kätte saada. (Intervjuud 2019)

Edasi olid küsimused projekti planeerimise osas. Vastajad arvasid, et põhinõuded võiksid olla enne projekti algust määratletud. Arvamus oli ka ühe vastaja poolt, et töötajatel ei ole ajalist ressursi iganädalaselt koosolekutel osaleda ja tulemit testida. Arvati, et

põhiprotsessid peaksid paigas olema enne arendust. Samas toodi välja ka paindlikkuse säilitamine. (Intervjuud 2019) Metoodika osas sobiks seega pigem traditsiooniline meetod

Töö autor lasi hinnata ka varasemat tarkvara hanke protsessi ning vastajad panid hindeks kas kahe või kolme viie punkti skaalal. Põhjendati seda sellega, et ei olnud vajaduspõhist tõestust, ega kirjeldatust miks seda vaja oli. Vähe oli kaasatud inimesi, kes igapäevaselt selle programmiga toimetavad. Vajaduste kaardistamine ja projektiplaneerimine ning protsessi juhtimine on olnud ettevõtte juhi sõnul olematud. Lisaprogrammid on juurde võetud reageerimispõhiselt mingi protsessi osa puudumise tõttu ja on keskendatud selle osa ära tegemisele, aga suurt pilti ei ole vaadatud. Selle tulemusena on näiteid, kus hangitud tarkvara tõttu on andmete sisestamist hoopis juurde tulnud, sest protsessi ei muudetud. (Intervjuud 2019)

Autor palus ka hinnata, kas on projektides mingis etapis olnud takistusi. Ning vastajad leidsid, et juurutamine on olnud kõige suurem takistus. Kõige keerulisemaks projekti etapiks peetakse planeerimist ja lõpetamist. Toodi välja, et lõpetamine on oluline just selle tõttu, et varasema kogemuse põhjal ei ole projekte korralikult lõpetatud, sest lõpetamise faasis hakatakse asju juurde soovima ning lõpetamine hakkab venima. (Intervjuud 2019) Autor järeldab, et tulemuste vastu võtmiseks peaks olema eraldi mõõdikud.

Meeskonna planeerimise osas oldi arvamusel, et sellise projekti puhul peaks kindlasti olema projektijuht ning projekti juhtrühm. Märgitigi ära ka, et lõppkasutajaid peaks võimalikult palju kaasama. Põhi meeskonna suuruseks võiks kujuneda 5-7 meeskonnaliiget. (Intervjuud 2019)

Intervjueeritavalt sai küsitud, mis on nende jaoks kõige tähtsam, kas kvaliteedi saavutamine või aja ja ressursside piires püsimine. Vastuseks oli, et kvaliteedi tagamine on kõige olulisem ning kvaliteedi parandamiseks ollakse nõus ka eelarvet suurendama. (Intervjuud 2019)

6.2 Projektiplaan ja meeskond projekti juhtimiseks

6.2.1 Ärilise vajaduse põhjendus

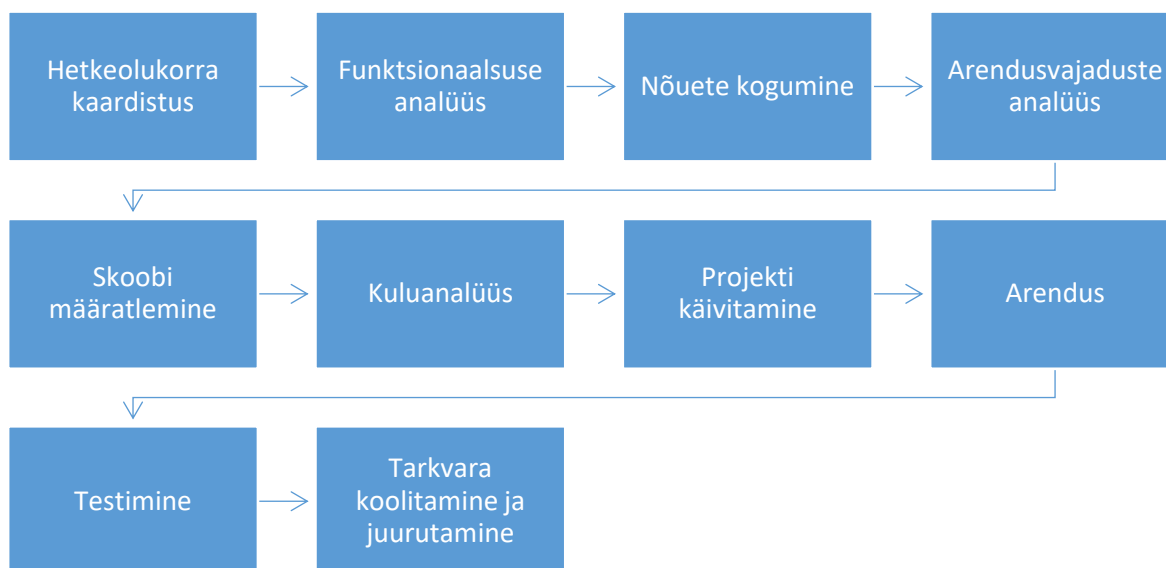
Teorias sai välja toodud, et ärilise vajaduse väljaselgitamine on projekti edukaks läbiviimiseks olulise tähtsusega. Ettevõttes X on ärilise vajaduse uurimiseks teinud intervjuud osakondade juhtidega ning lisaks on tehtud ka küsitlus lõppkasutajate seas.

Intervjuudes osakonnajuhtidega tuli välja, et suurim probleem on erinevate tarkvarade paljusus ning nende omavahelise liidestatuse puudumine. On protsessid, mis on tarkvaraga toetatud täies ulatuses ja ka protsess, kus info haldamine ja säilitamine on veel Excelis. Samuti toodi välja, et olemasolev tarkvarade süsteem ei ole kasutajasõbralik. (Intervjuud 2019)

6.2.2 Projekti läbiviimise metoodika

Teooria poole peal sai vaadeldud kahte erinevat metoodikat tarkvaraarenduse läbiviimiseks ning kui võtta arvesse ettevõtte valmisolek ja tööprotsessid, siis autor järeldab, et ettevõttele sobiks pigem traditsiooniline arendusmeetod nagu koskmeetod. Töö autor on seisukohal varasemate juhtumite põhjal, et kasutajad eelistavad pigem stabiilsust tarkvara osas ning ei tule muudatustega kohe kaasa. Kuna ka otsustajate ring on suur, siis võib arendus pikale venida, kui kasutada agiilset arendusmetoodikat, sest uuendused vajavad kinnitamist iga kord.

Teorias sai vaadeldud ka erinevaid projektijuhtimise metoodikaid ning varasemate kogemuste põhjal võib kindlasti öelda, et ettevõttel oleks metoodikat või raamistikku vaja projektide juhtimiseks, kuid ei peaks kopeerima üks ühele metoodikaid. Alljärgnevalt on skeem, milline võiks olla ettevõttes X tarkvara projekti läbiviimise etapid.



Joonis 7. Projekti etapid

Hetkeolukorra kaardistuse etapis tehakse olemasoleva süsteemi analüüs. Sisend saadakse kasutajatelt küsitluse põhjal ja autori poolt tehtud analüüsi käigus. Funktsionaalsuse analüüsis vaadeldakse, milline funktsionaalsus toetab äriprotsessi ning kas on vaja seda muuta. Nõuete kogumine on plaanis teha intervjuude ja vaatluste põhjal. Arendusvajaduste analüüsi põhjal saaks teha järeldusi, kas mingi oluline funktsionaalsus on puudu. Projekti käivitamine algatatakse, siis kui nõuded on paigas ja nii eelarve kui ka ajakava on kinnitatud. Testimine toimub lõppkasutajate poolt väiksemas testrühmas. Testjuhtude allikateks on kasutuslood. Tarkvara koolitusplaani teostatakse peale testimise lõppu.

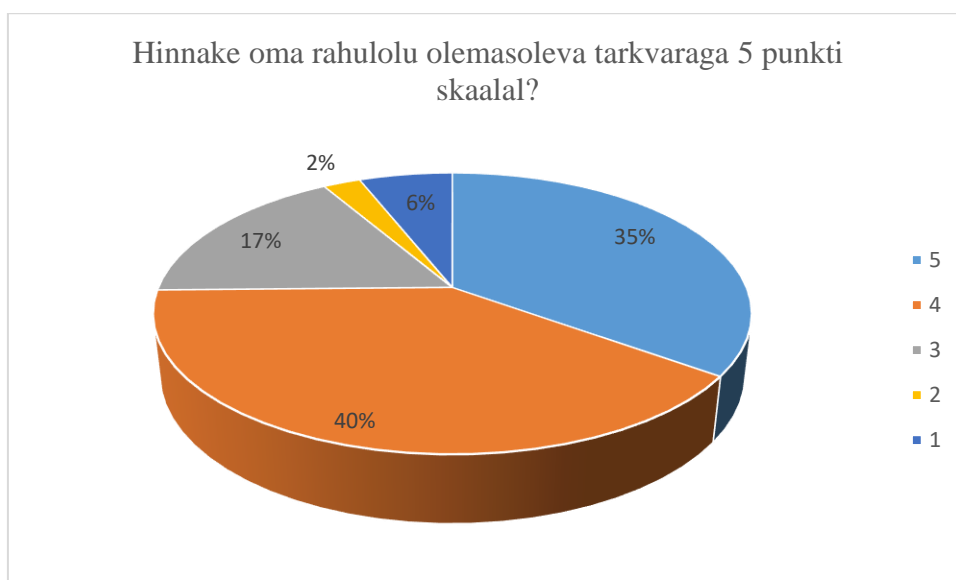
6.2.3 Hetkeolukord ja rahuolu tarkvaraga

Magistritöö üheks eesmärgiks oli kaardistada hetkeolukord ja rahulolu ettevõtte X põhjal. Selleks koostas autor küsitluse. Küsitluse valimiks oli 90 töötajat ning vastajaid oli 83. Küsitlus oli anonüümne ning viidi läbi elektroonselt. Küsitlusest vastanute protsent oli 92 %. Küsitlus koosnes seitsmest küsimusest. Küsimused, kus sooviti hinnangut punktiskaalal, tuli ka põhjendada arvulist hinnet. (Küsitlus 2019) Küsitluse küsimused on toodud magistritöö lisas. Küsitluse tulemusi on analüüsitud Excelis ja tarkvaraga PAST.

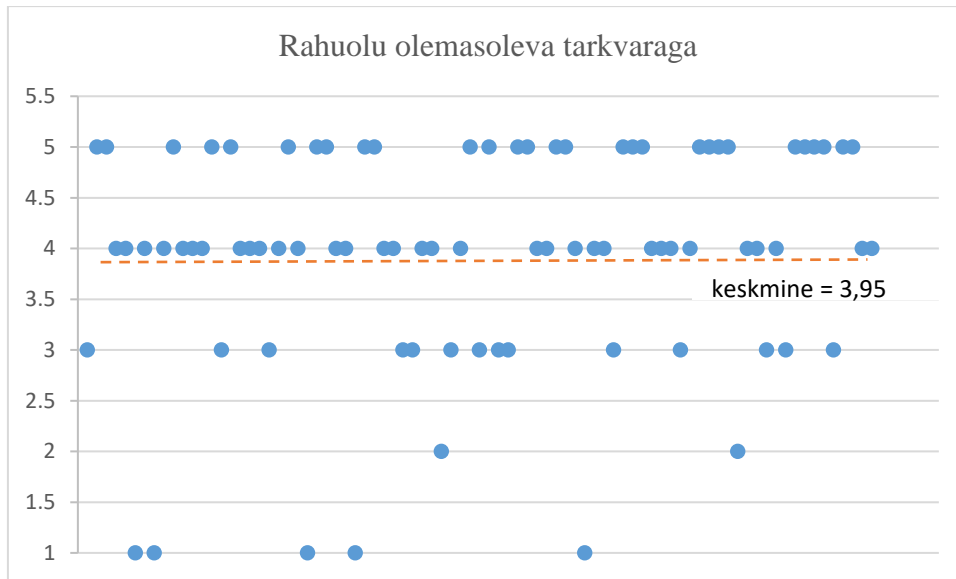
Küsitluses küsiti ka üldist rahulolu IT toe osas eesmärgiga teada saada, kas hetkel on kasutajatel tagatud piisav abi nii tarkvara kui ka arvuti üldprobleemide osas. Tulemuseks

oli, et viie punkti skaalal hinnati IT tuge keskmiselt 4,67 punkti vääriliselt. (Küsitlus 2019) Hea tulemus on autori arvates tingitud mitmest tegurist – arvutiparki on uuendatud, infrastruktuuris on muudatusi tehtud ning ka IT teenusepakkuja on vahetunud.

Asjaolu, et üldises joones ollakse IT toega rahul võib mõjutada ka rahulolu ettevõttes olevate tarkvaradega, kuna on parandatud süsteemide töökindlust. Selle hüpoteesi uurimiseks kasutab autor korrelatsioonanalüüsi. Korrelatsioonikordaja IT toe rahulolu ja tarkvaraga rahulolu osas on küsitlustulemuse põhjal 0,15 mis tähendab, et tunnused on omavahel nõrgalt seotud. Rahulolu ettevõttes X oleva tarkvaraga on 3,95. Keskmine on peaaegu punkti võrra kehvem kui IT toe oma. (Küsitlus 2019) Allpool olevatel joonistel on näha täpsemat jaotust.

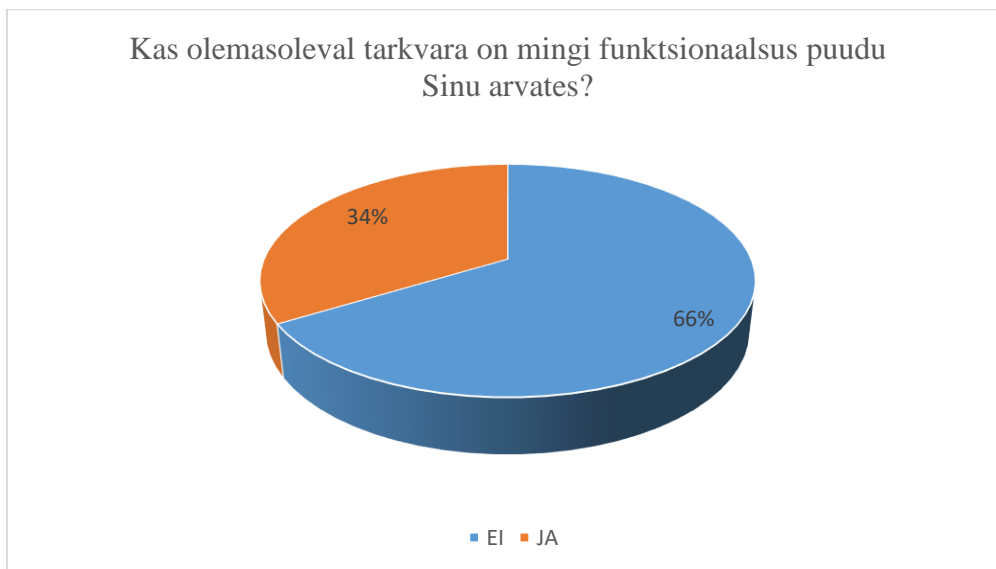


Joonis 8. Rahulolu olemasoleva tarkvaraga (Küsitlus 2019)



Joonis 9. Rahuolu tarkvaraga, keskmine ja jaotus (Küsitlus 2019)

Järgmisena oli küsitluses küsimus, kas olemasoleval tarkvaral on mingisugune oluline funktsionaalsus puudu. Küsimuse eesmärk oli teada saada kui suur osa kasutajatest sooviks tarkvarale lisafunktsionaalsust. (Küsitlus 2019) Tulemust on näha allpool oleval joonisel.



Joonis 10. Funktsionaalsuse piisavus kasutajate arvates (Küsitlus 2019)

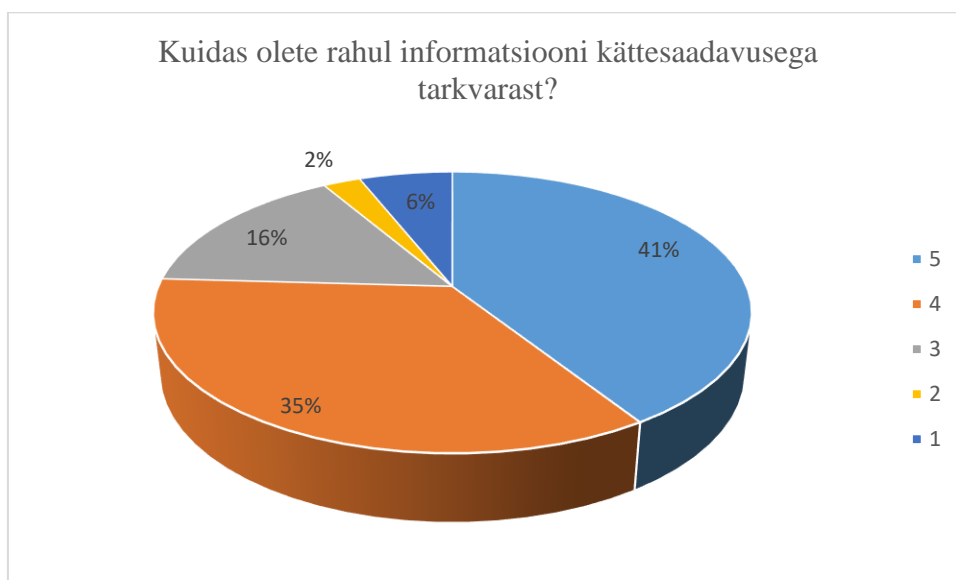
Kuigi rahulolu tarkvara osas ei olnud nii hea, kui toe osas, on enamus pidanud funktsionaalsust piisavaks (Küsitlus 2019). Antud tulemuse põhjal ei saa aga täielikku järeldust teha, sest vastajaid oli erinevatest osakondadest ja võib olla ongi mõne osakonna vajadused kaetud.

Lähemaks uurimiseks küsis autor, et vastust põhjendatakse ning lisaks ka, millist funktsionaalsust oleks juurde vaja (Küsitlus 2019). Autor jagas saadud vastused alljärgnevasse kategooriatesse ning detailsemaid ettepanekuid antud magistritöös ei kajastata. Ettepanekud funktsionaalsuse osas (Küsitlus 2019):

- analüüsivõimaluste lisamine;
- Iliendisuhte juhtimise mooduli/ osa lisandumine;
- personalihalduse funktsionaalsuse parendus;
- palgaarvestuse funktsionaalsuse parendus.

Järgmine küsimus oli, et kuidas kirjeldatud funkstionaalsus tööd aitaks paremini teha. Igasugune arendus on seotud kuludega ning tarkvara peab tooma lisandväärtust ning toetama äriprotsesse. Vastuste tulemusena saab järeldada, et muudatused aitaksid aega kokku hoida, vähendaksid vigade arvu ning võimaldaks lõpetada andmete dubleerimise. Samuti on toodud välja, et kui oleks ühtne infosüsteem lõppeks mitme programmi vahel pendeldamine ära. (Küsitlus 2019)

Küsitluses soovis autor ka teada, kuidas ollakse rahul informatsiooni kättesaadavusega tarkvarast. Küsitluse tulemused näitavad, et informatsiooni kättesaadavus on hea. Keskmine tulemus oli 4,02 ja täpsem jaotus on alloleval joonisel. (Küsitlus 2019)



Joonis 11. Tarkvarast info kättesaadavusega rahulolu (Küsitlus 2019)

Autor analüüsis ka tarkvaraga rahulolu ja informatsiooni kättesaadavuse rahulolu seost.

Correlation

	A	B
A		0.73675
B	0.73675	

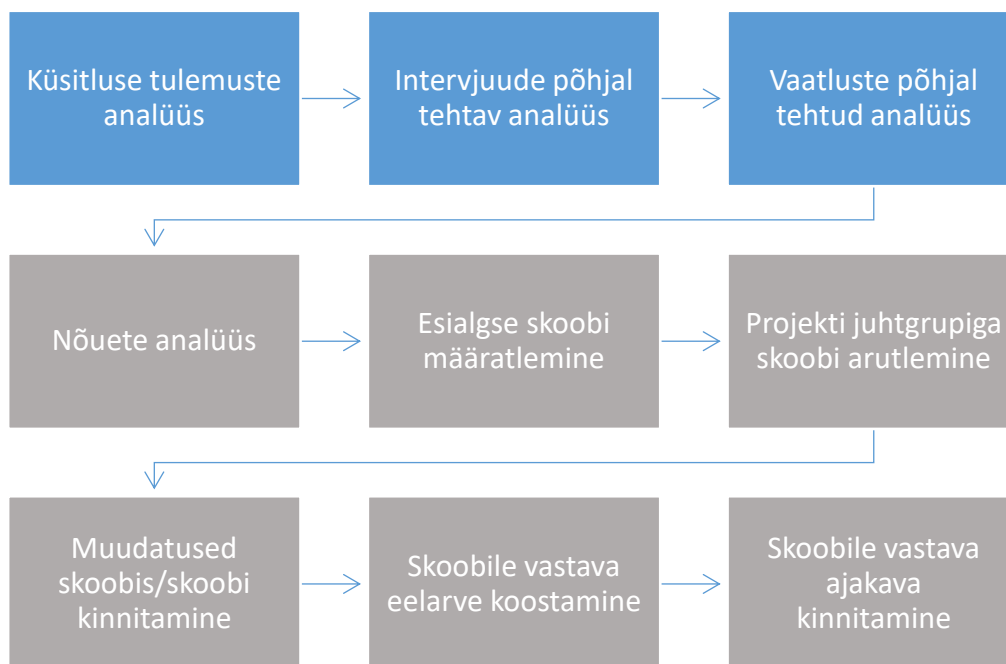
Joonis 12. Korrelatsioon info kättesaadavuse ja rahulolu tarkvaraga vahel (Küsitlus 2019)

Korrelatsioonikordaja on 0,74 mis tähendab, et tunnused on omavahel tugevalt seotud. Selline järeldus tuleb välja ka tekstiväljaga küsimustest. Nimelt oli märgitud probleemiks, et info on erinevates tarkvarades ja otsimine võtab aega. (Küsitlus 2019)

Kokkuvõttes järeldan küsitluse põhjal, et ettevõtte vajaks tarkvarale lisafunktsionaalsust ning äriiline vajadus on täiesti olemas. Ettevõttes X on autori arvates kõige kriitilisem etapp tarkvara juurutuse osa, sest muudatuste vajalikkus tuleb kasutajatele kõigepealt selgeks teha. Autori arvates aitab sellele protsessile kaasa lõppkasutajate kaasamine, mida sai ka antud küsitluse raames tehtud.

6.3 Skoop

Ettevõttes X määratakse projekti skoop peale esialgset kaardistamist. Skoobi määratlemiseks on kavas kasutada alljärgnevat skeemi.



Joonis 13. Skoobi määratlemise ettevõttes X

Autor arvas, et skoobi määramiseks on antud skeem parim, sest nõuete analüüsile eelnevad etapid – küsitlus, intervjuud ja vaatlus täiendavad üksteist ning lisaks annab küsitluse tegemine kasutajatele võimaluse ka ise kaasa rääkida ning panustada võimalikku projekti. Sinises kastis olevad etapid tehakse kindlasti projekti X raames, kuid väiksemate projektide puhul ei pruugi vaja minna nii küsitlust, intervjuusid või vaatlust. Olenevalt projekti suurusest võib ühest meetodist piisata.

6.4 Ajakava

Projekti kavandamisega on plaanis alustada 2019 aastal. Aasta esimeses pooles toimub täpse projektikava koostamine koos ajagraafikuga. Esimese poolaasta jooksul selgitatakse ka esialgsed nõuded tarkvarale.

Tabel 5. Projekti eeldatav ajakava

Projekti etapp	Algus	Lõpp
Hetkeolukorra analüüs	2019 I kvartal	2019 I kvartal
Funktsionaalsuse analüüs	2019 II kvartal	2019 II kvartal
Nõuete kogumine	2019 II kvartal	2019 III kvartal
Arendusvajaduse analüüs	2019 III kvartal	2019 III kvartal
Skoobi määratlemine	2019 III kvartal	2019 III kvartal
Kuluanalüüs	2019 III kvartal	2019 III kvartal
Projekti käivitamine	2019 IV kvartal	2019 IV kvartal
Arendus	2019 IV kvartal	2020 II-II kvartal
Testimine	2020 II kvartal	2020 II kvartal
Juurutamine/koolitus	2020 III kvartal	2020 IV kvartal

Nagu ajakavast on näha siis põhinõuded kirjeldatakse algfaasis, mistõttu on antud etapile ka pikem aeg planeeritud. Samas ei välista autor, et arenduse faasis selgub veel nõudeid, mida tuleks arendusse viia. Seega võib ka koskmudeli raames olla iteratiivsust ning see ei tähenda, et testimine toimub ainult projekti lõpus.

6.5 Nõuete kaardistus

Nõuete täpsemaks kaardistamiseks on autor valinud ettevõttes X küsitluse, intervjuude tegemise ning protsesside vaatluse. Selliste meetodite kasutamine koos võimaldab arvestada nii lõppkasutaja vajadusi kui ka samas näha protsessi kõrvalt pealt. Saadav info

vormistatakse kasutuslooks. Teooria poole peal sai kirjeldatud, millest kasutuslugu peaks koosnema, ning selle põhjal sai koostatud alljärgnev mall. Mall sisaldab kasutusloo lühikirjeldust, tegevusi, eeldusi ja resultaati (Matveus 2009, lk 29). Autor lisas ka kasutaja malli.

Tabel 6. Mall kasutuslugudeks

Kasutuslugu	X
Kirjeldus	kasutusloo lühike seletus
Tegevused	kasutusloo raames tehtavad toimingud
Kasutajad	kes on teostajad
Eeldused	teostamiseks vajalikud ressursid ja õigused
Resultaat	Kasutaja poolt sooritatud tegevuse tulemus

Selliste kasutuslugudega on plaanis katta ära ettevõtte põhiprotsessid ja toimingud. Magistritöös on näidisenä toodud ka protsess kasutusloo ja joonisena ning lisaks on protsessi täiendatud küsitlusest välja tulnud arendusvajadusega.

Alljärgnevalt on malli kasutades üks näidis ostuarve protsessist. Antud näidis on lihtsustatud ning sündmustel on ka alternatiive, mis on paremini kajastatud joonisel 14.

Kasutuslugu : Ostuarve protsess

Kirjeldus: Ostuarve sisestamine ja kinnitus läbi raamatupidamissüsteemi

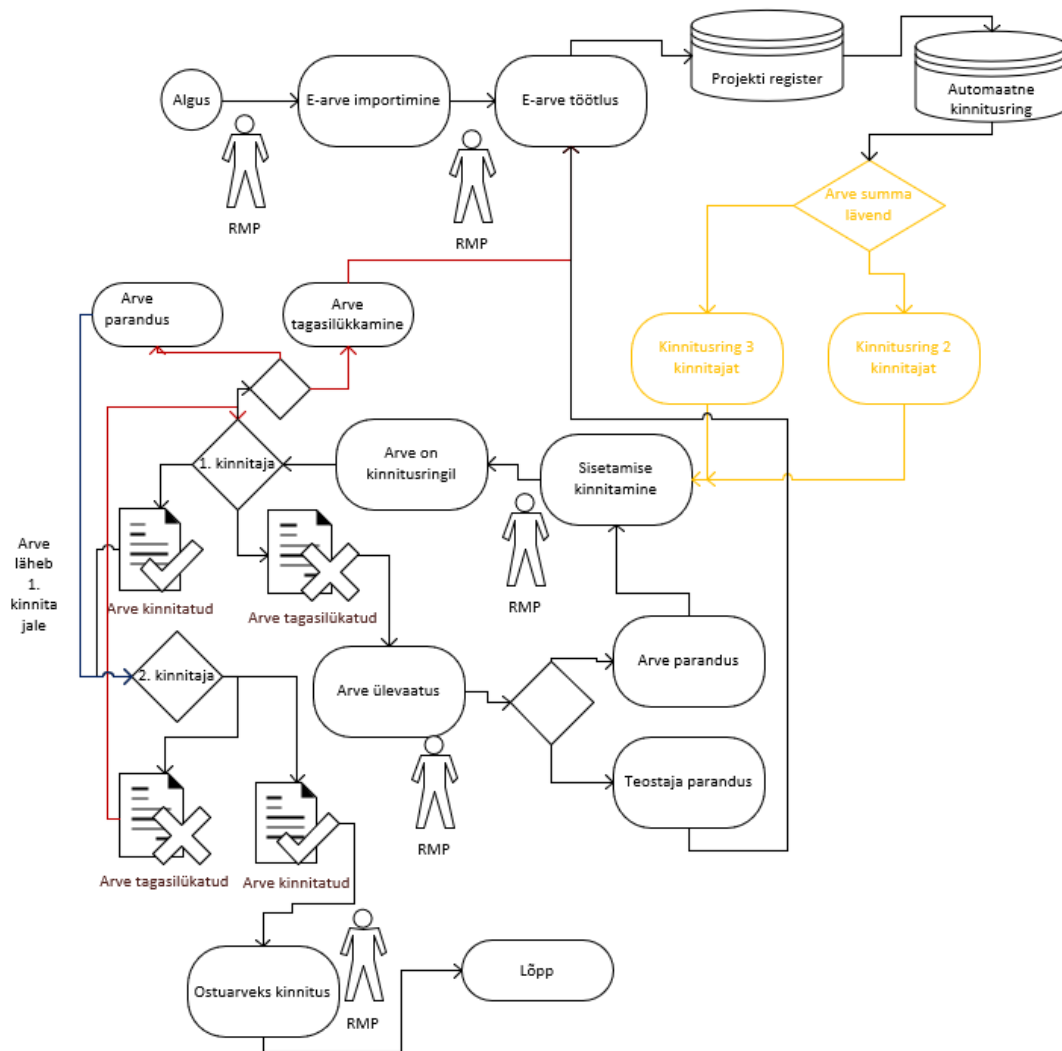
Tegevused:

- a) raamatupidaja impordib läbi e-arvekeskuse XML formaadis arve
- b) raamatupidaja töötleb arvet
- c) süsteem leiab arvele teostajad läbi projektide seoste
- d) arve suunatakse kinnitusringile
- e) arve kinnitatakse kõigi kinnitajate poolt
- f) arve muutub ostutellimusest ostuarveks

Kasutajad: raamatupidamine, ostu teostaja, vastavalt teostajale kinnitajad

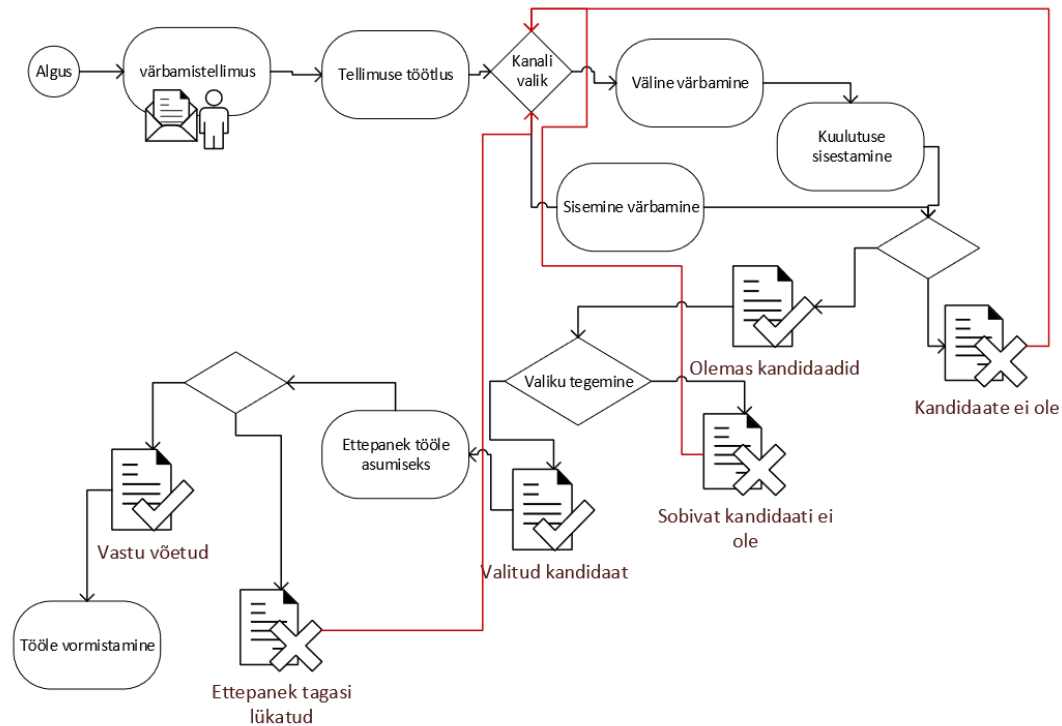
Eeldused: õigused raamatupidamissüsteemis

Resultaat: arve on kinnitatud ja ostuarveks vormistatud.



Joonis 14. Ostuarve protsess

Joonisel on kirjeldatud vaatluse põhjal ostuarve protsessi ettevõttes X. Kollasel taustal on märgitud üks funktsionaalsus, mis on puudu, kuid mida peaks kindlasti juurde arendama. Parandusi protsessis on vaja teha rohkem just kasutajate mugavuse poolel kinnitusringil. Põhjus, miks seda teha on vaja on see, et kolmandal kinnitajal on kõige rohkem arveid, mida vaja üle vaadata. Hetkel kinnitab kolmas kinnitaja kõiki arveid alates paari eurosest lõpetades mitme tuhande eurodega. Olenemata summast on aeg täpselt sama, mis arve kontrollimisele kulub. Arendusega oleks plaanis seada rahaline piir, mis määraks ära kinnitajate arvu.



Joonis 15. Näide ettevõtte ühest põhiprotsessist

Joonisel on näidatud ettevõtte ühte põhiprotsessi – värbamist. Hetkeseisuga ei toeta seda protsessi ükski tarkvara ning tööd tehakse sisuliselt Excelis. Värbajaid on ettevõttes kolm ning töö lihtsustamiseks oleks vaja tarkvara, mis suudaks hallata värbamisvajadusi, käimasolevaid värbamiskampaaniaid ning samuti potentsiaalsete töötajate sooviavaldusi. Samuti peaks olema võimalus kasutada tarkvara sisemiseks värbamiseks ehk olemasolevatele töötajatele oleks võimalus pakkuda näiteks lisatööd.

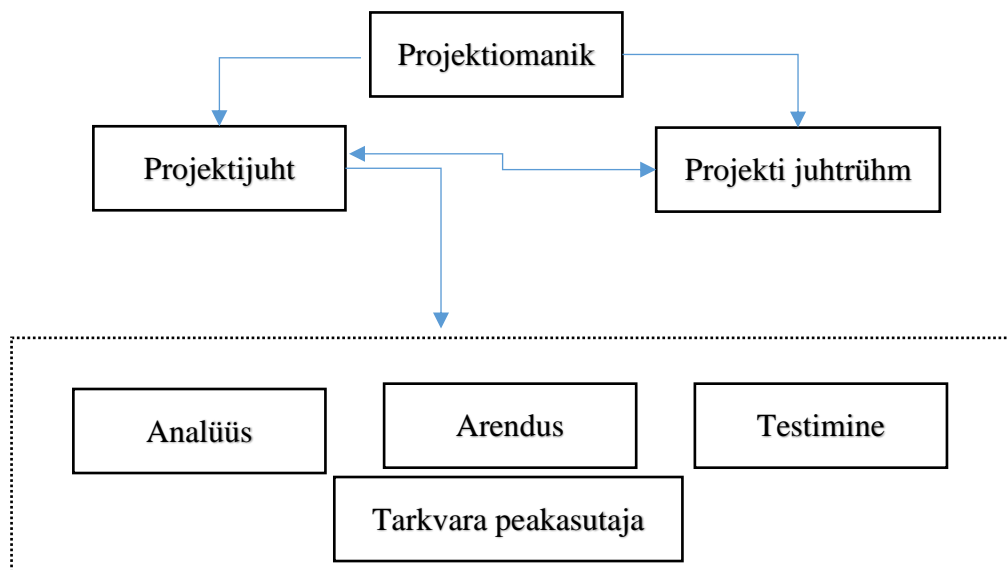
6.6 Meeskond

Teorias sai vaadeldud milliste teguritega peab arvestama meeskonna moodustamisel ja kokkuvõtvalt sai tehtud järeldus, et meeskond peaks olema võimalikult mitmekesine. Samuti on oluline rollide ja vastutuse määramine. Alljärgnevalt pakub autor välja projekti jaoks vajalikke rolle ja vastutusala. Rollid ja vastusala ning põhitegevused baseeruvad teooria osas kirjeldatud rollidele, mida autor on vastavalt projektile ja ettevõttele kohandanud. Rollid on projekti omanik, projektijuht, projekti juhtrühm, projekti meeskond (Zwikael, Meredith 2018, lk 485). Alljärgnevas tabelis on kirjeldatud rollid ja mõõdikud.

Tabel 7. Rollide jaotus näidisprojektis

Roll	Rolli vastutus	KPI
Projekti omanik	Projektile eelarve eraldamine, projekti eestvedamine	Investeeringisotsuse põhjendatus
Projektijuht	Projekti ohje, dokumentatsioon, kommunikatsiooni planeerimine, skoobi hoidmine, muudatus ettepanekute tegemine. Projekti meeskonna juhtimine	Projekti teostatakse eelarve piires. Projekt on ajagraafikus. Projekti muudatused on kõik kooskõlastatud.
Projekti juhtrühm	Vastutav projekti sisendi eest, muudatuste kinnitamine.	Kvaliteetsed sisendid ärijuhtumite näol.
Projekti meeskond	Analüüsi, arenduse ja testimise teostus	Projekti plaani realiseerimine

Tabelis on vähem rolle, kui teorias vaadeldud Zwikaeli ja Meredithi jaotuses (Zwikael, Meredith 2018, lk 485), kuid autor on seisukohal, et nende rollidega on kaetud kogu projekti vajadus. Samuti tuli see välja ka osakonnajuhtidega tehtud intervjuus, kus kõik vastajad pidasid oluliseks projektijuhti ja projekti omaniku olemasolu ning selget vastutust projektis (Intervjuud 2019).



Joonis 16. Näidisprojektile sobilik struktuur

Joonisel on struktuur rollide põhjal. Projekti omaniku ülesanne on projekti toetada, tagada eelarve ning meeskonda eest vedada. Projektijuht on vastutav projekti eelarve, aja ja ressursside eest. Kuna ettevõttes ei ole kogemust sellise projekti juhtimisel, tuleks see kompetents sisse osta, sest projektijuhil on kandev osa selles projektis. Projektijuhtrühm võtab tulemid vastu.

Samuti tuli intervjuudes välja, et uuel tarkvaral võiks olla nõ peakasutaja või siis tooteomanik, kes juurutuse ajal kui ka peale seda aitab kasutajaid tarkvara kasutamisel (Intervjuud 2019).

6.7 Riskide kaardistus projektis

Riskide hindamiseks on plaanis kasutada 5 punkti skaalat ning hinnata tuleb nii riski mõju kui ka selle tagajärge. Allpool olevas tabelis tuleks alustada riski kaardistamist määrates sellele allika, edasi tuleks hinnata esinemise tõenäosust ja mõju projektile. Teooria poole peal sai välja toodud, et PMBOK-i järgi on riskide toimetulemiseks neli meetodit - elimineeri, aktsepteeri, kanna üle, või leevenda ning riskide tegelemise protsess on iteratiivne (Virine, Trumper 2008: 119, 123-124). Need neli meetodit on ka allolevas tabelis toodud. Riskid kaardistab projekti juhtrühm.

Tabel 8. Mall riskide kaardistuseks projektis X

Riski nimetus	X
---------------	---

Riski allikas	Allikas võib tuleneda skoobist, ajagraafikust, nõuetest, meeskonnast jms
Riski esinemise tõenäosus	Hinnata, kui reaalne on riski realiseerumine
Riski mõju projektile	Hinnata, kuidas see mõjutab projekti edasist kulgu
Meetmed riskiga toimetulemiseks	Otsust kuidas riskiga tegeleda (elimineeri, aktsepteeri, kanna üle, või leevenda)

Riskide kaardistamist on plaanis teha enne projekti käivitamist. Kuna riskide hindamine on pidev protsess, siis kindlasti riski hindamise dokument täieneb projekti jooksul, sest riske võib lähtuvalt projektist juurde tulla.

6.8 Tulemite vastuvõtmine

Iga tulemi üleandmine peaks olema projektis fikseeritud. Tulemus võetakse vastu, kui see on aktsepteeritud projekti juhtrühma poolt. Tellija peab testima enne vastu võtmist, kas antud tulem vastab kirjeldatud nõuetele.

Tabel 9. Tulemite vastuvõtmise mall

Test	X
Testi sisu	Lühikirjeldus, mida testitakse
Testija roll süsteemis	Millistes õigustes testiti
Tulemus	Testi kokkuvõte
Kommentaariid /parandused	Kas midagi vaja muuta?
Tulem vastu võetud/ ei ole vastu võetud	Otsus kas tulem vastab nõuetele

Testimise eesmärk tellijapoolel on vaadata, kas tarkvara toimib nii nagu on soovitud. Intervjuus osakonnajuhtidega tuli välja soov, et tarkvaral võiks olla peakasutaja, kes tunneks tarkvara kui tervikut, mitte ainult ühte osa (Intervjuud 2019). Seega peaks üheks testijaks kindlasti olema peakasutaja

6.9 Järeldused ja valideerimine

Magistritöö raames pakkus autor välja raamistikku projekti läbiviimiseks. Raamistik toetus erinevatele meetodikatele nagu PRINCE2 ja PMBOK ning hõlmas endas ka praktikad, mida autor kirjeldas teooria poole peal. Autori arvates oli õigustatud nõuete kogumise ja rahulolu määramiseks kasutada nii küsitlust, intervjuud kui ka vaatlust, sest

see andis parema ülevaate probleemidest ning ühtlasi näitas ära, ka et osakonniti ja kasutajate seas on vajadus erinev. Kuna sellise mahuga eeltöö ei ole otstarbekas ilmselt kõigi tarkvara projektide puhul, siis autor ka mallis eraldas ainult näidisprojekti raames tehtava osa (Joonis 13). Samuti arvab autor, et valminud mallid ei ole igavesed, vaid peaks üle vaatama teatud aja pärast ning oma sobivust tõestama ettevõtte kontekstis. Magistritöö raames vormistati malli põhjale näidis ühest protsessist.

Autor arvas, et ettevõttele on pigem sobilikum koskmeetod, samas on autor teadlik et suund on praegusel ajal pigem agiilse metoodika suunas. Autor tegi sellise järelduse arvestades asjaolu, et projekt on suur ning projektile on planeeritud ajaraam. Kuna eeldatavasti vahetatakse ka raamatupidamistarkvara, siis peab arvestama ka erinevate ajaliste nüanssidega ja sellest tulenevate kitsendustega. Samuti tuli arvestada ka varasema kogemuse ning meeskonnaliikmete valmidusega. Autor mõnab, et antud valik ei pruugi toimida, selle projekti jaoks, kuid olemasoleva info põhjal sai nii järeldatud.

Kuna ettevõttes on olnud varasemalt probleeme projektide lõpetamisega ehk soovitakse lisafunktsionaalsust juurde kogu aeg tellida, siis tundub selle vältimiseks samuti mõistlikum kasutada koskmeetodit. Võimalik, et ettevõttes teiste tarkvara projekti puhul on mõistlik juurutada agiilset arendusmetoodikat. Autor ei välista ka selle projekti puhul, et mingi osa tehakse ikka iteratiivselt, kuna ka intervjuu käigus selgus, et oluline on ka paindlikkus säilitada. Ka koskmudeli puhul saaks paindlikust sisse tuua ja mõned testid teha juba arenduse ajal, et olla kindel tulemi õigsuses. Mudelit ei peaks autori arvates jäigalt järgima.

Ka teooria poole peal tõi autor välja viidates allikatele, et metoodikast olulisem on see, kuidas seda rakendada. Ükski metoodika ei saa üksi tagada projektile edu või ebaedu. Pigem rõhutab autor läbivalt töös, et väärtuse loomine ettevõttele, on kõige olulisem. Seda väärtus saab luua, siis kui ettevõtte äriprotsessid on analüüsitud ning vajadus on läbimõeldud. Ettevõtte kontekstis viis see järelduseni, et nõuded peaksid olema enne paigas, sest nagu ka juhtumianalüüs näitas, ei ole reageerimispõhiselt tarkvaraarendus toimunud.

Autor leidis, et tarkvaraarendus projekti jooksul peaks kindlasti kasutama dokumentatsiooni, et oleks kõikidele osapooltele arusaadav mida, miks ja kuidas tehakse. Dokumentatsioon aitaks ära hoida autori arvates selle, et tulevikus ei teata, miks selline

funktsionaalsus üldse telliti ja kuidas seda kasutada. Autor kirjeldas teises peatükis probleeme, miks tarkvara projektid ei õnnestu ning järeldas, et ka ettevõtte X on nende probleemidega kokku puutunud. Samas ei olnud see probleemide nimekiri lõplik ning eraldi võiks veel välja tuua probleemi kui kasutajatele ei suudeta tarkvara olulisust tõestada ning vastuseisu ületada.

Magistritöös oli peatükk ka meeskonna moodustamisest tarkvara projekti jaoks. Varasemalt ei ole ettevõtte kaasanud spetsialiste väljaspool organisatsiooni, kuid teatud juhtudel oleks see vajalik. Meeskonnas peaks olema mitmekesise taustaga inimesed ning ka inimene, kes teaks ka paremaid praktikaid. Seda väidet toetab ka juhtumianalüüs, kus siis ettevõtte juht nägi juurpõhjusena projekti ebaõnnestumisel, et protsessis oldi liiga kinni ning ei mõeldud kordagi protsessi muutmisele vaid taheti tarkvara muuta (Juhtumianalüüsi intervjuu 2019). Protsessi efektiivsust ja sobivust ei hinnatud. Väljaspool organisatsiooni tulev konsultant või projekti meeskonnaliige aitaks protsesse hinnata teise pilgu läbi. Seega võiks ettevõtte kaaluda projekti jaoks projektijuhi värbamist.

Magistritööga sai püstitatud eesmärgid täidetud. Rahulolu tarkvaraga on kaardistatud, esialgsed nõuded funktsionaalsusele kirja pandud ning samuti analüüsitud tarkvara hanke protsessi ning tehtud parandusettepanekud.

Kokkuvõte

Magistritöö eesmärkideks oli kaardistada hetkeseis ja rahulolu olemasoleva tarkvaraga ning selgitada välja esialgsed nõuded tarkvarale. Hetkeseis ja rahulolu sai kaardistatud läbi küsitluse ning intervjuude. Küsitlusest sai järeldada, et rahulolu tarkvaraga on lõppkasutajate arvates pigem hea, kuid toodi välja ka puudused funktsionaalsuses. Detailsem intervjuu juhtkonna ja osakonnajuhtidega andis sisendi, et kõige suurem puudujääk funktsionaalsuses on just analüüsi osas ning probleemiks on ka andmete dubleerimine tarkvarades. Kõike eeltoodud kokkuvõttes on ettevõttel reaalne vajadus tarkvara projekti jaoks.

Magistritöö esimeses peatükis on kirjeldatud taust ja probleem. Ettevõtte kaalub tarkvara välja vahetamist ning enne seda on vaja teha põhjalikum analüüs nii tarkvara osas kui ka tarkvara projekti juhtimise osas. Magistritöö esimeses peatükis on püstitatud eesmärgid ning kirjeldatud ka meetoodikat.

Magistritöö teises peatükis kirjeldatakse, mis on oluline nõuete kaardistamisel ning millised on meetodid selle tegemiseks. Ühe olulisema järeldusena peatükist saab välja tuua selle, et nõuetel peab olema fikseeritud päritolu. Peatükis antakse ka ülevaade riskide kaardistamise tehnikast ning projekti edutegurite määramisest. Peatükis käsitletud etappe ja kirjeldust kasutas autor ka näidisprojekti planeerimisel.

Töö kolmanda peatükis on antud ülevaade projektijuhtimise meetoodikatest. Peatüki eesmärk on tuua ülevaade meetoodikatest ning kirjeldada olulisemaid pidepunkte. Näidisprojekti raames ei hakka ettevõtte kumbagi meetoodikat üks ühele rakendama, kuid raamistiku ja etappide kirjelduse jaoks on võimalik meetoodikaid ära kasutada kui parima praktika näitena.

Neljanda peatüki eesmärk oli võrrelda kahte erinevat arendusmeetoodikat ning tuua välja, millises kontekstis mingi meetoodika sobib. Tuginedes allikatele jõudis autor järeldusele, et traditsiooniline koskmeetod sobib projektidele, mis on suuremahulised, kuid mille nõuded on hästi määratletud. Teine vaadeldav meetoodika ehk agiilne meetoodika sobib pigem siis, kui tarkvara on vaja kiirelt tarnida ning kui kõiki nõuded ei ole ette teada.

Viiendas peatükis on kirjeldatud projektimeeskonna planeerimisega seotud tegureid. Peatükiga jõudis autor järeldusele, et meeskonna peaks moodustama mitmekesise

taustaga inimesed ning meeskonnas peaks olema juba enne projekti algust määratletud rollid ja vastutus.

Kuuendas peatükis on kirjeldatud näidisprojekti ettevõtte x põhjal. Näidisprojekt sai alguse hetkeolukorra kaardistamisest, mis hõlmas endas nii kasutajate rahulolu uuringut kui ka tarkvara funktsionaalsuse analüüsi. Küsitluse tulemusena selgus, et rahulolu tarkvaraga sai viie punkti skaalal hinnangu 3,97. Skoor oli oluliselt madalam võrreldes üldise rahuloluga IT toe osas. Peale hetkeolukorra kaardistamist järeltas autor, et ettevõtte on äriplaneeritud tarkvara arendusprojektiks. Peatükis pakub autor välja meeskonna ja rollide jaotuse ning ajakava projekti teostuseks.

Magistritöö eesmärgiks oli ka kirjeldada edutegureid ning pakkuda välja ettevõttele sobiv metoodika või raamistik projekti elluviimiseks. Töö autor järeltas nii intervjuude kui ka varasema kogemuse põhjal, et ettevõttele sobib antud projekti kontekstis pigem traditsiooniline koskmeetod. Teooria osas sai kirjeldatud, mis meetod mingis kontekstis sobib ning kuna tegemist on suurema projektiga, millele on planeeritud ka ajaraamistik, siis sobiks lineaarne lähenemine rohkem.

Juhtumianalüüsi käigus hinnati eelnevat tarkvara hankimise protsessi ja kuna eelnevat tarkvara protsessi efektiivsust hinnati madalaks, siis lõi autor välja ka juurpõhjused ning võimalikud lahendused. Suurima probleemina järeltas autor seda, et nõuded ei olnud piisavalt defineeritud ja nendel ei olnud ka omanikke. Samuti ei olnud kogu protsess piisavalt dokumenteeritud, mis raskendas hilisemat kasutust. Järeldusena võiks veel tuua, et otsused tehti kiirustades ning vajadus ei mõeldud piisavalt hästi läbi. Samuti ei tehtud eeltööd ega ka konsulteeritud liidestuse osa teiste tarkvaradega ning teadmine, et tarkvarasid ei ole võimalik ühildada tuli juba liiga hilja.

Antud töö edasiarendusena näeks autor analüüsi peale näidisprojekti teostamist. Teada, on eelneva tarkvara projekti teostuse probleemid, kuid oluline on, kas neid on näidisprojekti raames välditud. Autor arvab, et näidisprojektis kasutatud malle saaks ettevõtte kasutada ka tulevaste projektide puhul, kuna mallid universaalsed.

Summary

The aim of this thesis was to map current situation and satisfaction with software and to clarify essential requirements for the software. Current situation and satisfactory was mapped by survey and interviews. Author made conclusion that overall satisfactory amongst end-user is quite good, but improvement was needed in functionality. More detailed overview about software was gathered during interviews with department managers and management. The result was that for them a big necessity – analyze tools were not covered with current software and different software's needed duplicated input by the users. Concluding all the above it is clear that the organization needed a software project.

In the second chapter of this thesis a general overview of project steps is given. In this chapter author bring out practices and suggestions based on scientific sources. The most valuable conclusion is that every requirement needs to have specific origin. This conclusion is supported by the case-study as well. In this chapter an overview of risk management techniques is also presented.

In the third chapter the author gives overview of two project management methodologies. The aim of this chapter is to describe ways to organize project using frameworks or methodologies. With example project organization will not copy methodology one to one but the methodology is helpful for organizing project documents and processes.

The aim of fourth chapter was to compare two development methodologies. The author came to conclusion based on sources that choosing the right methodology depends on project. If the project is large scale and requirements are well known, then it is best to use the waterfall method. The second method that author described is agile methods and it is more useful if project needs to be done quickly and the client does not know exactly the requirements, or the requirement are a subject to constant change.

In fifth chapter author discusses team effects on projects and how do choose team members. With this chapter author comes to conclusion that teams should consist of people with different backgrounds and knowledge. Also, roles and responsibility should be clear in the begging of the project.

The sixth chapter is about describing example project based on organization x. Example project starts with mapping the current situation and analyzing the survey results. After the survey and interview author concluded that organization needs software project to be more effective in business processes. In this chapter author presents a time frame for executing the project and roles and responsibility of team members.

The aim of this thesis was to describe success factors and to present a methodology for the example project. The author of this work concluded that organization is more suited to us the waterfall method. In theory was described which method is best for different situations and because it is large scale project then linear approach is more suitable.

Author also did case-study for previous software procurement process. Management and department managers evaluated previous process and acknowledged mistakes and flaws in previous process. Author concluded that the biggest problem was that requirements were not defined and there was not enough knowledge how to use the software. Another problem was a lack of awareness why this functionality was needed. Author also concluded that the process was not documented well and that made it difficult for the users and people who were responsible for the trainings and launching.

For further work author would like to see the analyze after example project is made. Previous process has been evaluated and suggestions for change are made so it would be good to know if suggestions and improvements took effect on results.

Kasutatud kirjandus

- A Guide to the Project Management Body of Knowledge. (2013). Fifth edition. *Project Management Institute, Inc. Pennsylvania, USA* (lk 120, 133)
- Albert, M. Balve, P. Spang, K. (2017). Evaluation of project success: a structured literature review", *International Journal of Managing Projects in Business, Vol. 10 Issue: 4, 2017* (lk 799, 809)
- Altiner, S., Ayhan, M. (2018). An approach for the determination and correlation of diversity and efficiency of software development teams. *South African Journal of Science., Vol. 114 Issue 3/4, Mar/Apr 2018* (lk 69-76)
- Balaji, S., Murugaiyan. M. S. (2012). Waterfall vs V-Model Vs Agile: a comparative study on SDLC. *International Journal of Information Technology and Business Management . Vol.2 No.1, 29th June 2012* (lk 29)
- Barney, S., Aybüke, Claes, W. (2008). A product management challenge: Creating software product value through requirements selection. *Journal of Systems Architecture., Vol. 54 Issue 6, June, 2008* (lk 577)
- Bilgaiyan S., Santwana S., Mishra S., Madhabananda D. (2017). A Systematic Review on Software Cost Estimation in Agile Software Development. *Journal of Engineering Science & Technology Review, Vol. 10 Issue 4, 2017* (lk 59)
- Bowen, P.L., Heales, J., Vongphakdi. (2002). Reliability factors in business software: volatility, requirements and end-users., *Vol. 12 Issue 3, July, 2002* (lk 187-188)
- Brewer, G., Strahorn, S. (2012). Trust and the Project Management Body of Knowledge. *Engineering, Construction and Architectural Management, Vol. 19 Issue: 3, 2012* (lk 293)
- Burke, R. (2013). Project Management Planning and Control Techniques. Fifth edition. *John Wiley & Sons Ltd ,West Sussex, United Kingdom* (lk 169)
- Carton, F., Adam, F., Sammon, D. (2008). Project management: a case study of a successful ERP implementation. *International Journal of Managing Projects in Business, Vol. 1 Issue: 1, 2008* (lk 121)
- Chau, T., Maurer, F., Melnik, G. (2003). Knowledge Sharing: Agile Methods vs. Tayloristic Methods. *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. IEEE*
- Crawford J., Leonard L.N. K., (2012). Predicting post-meeting work activity in software development projects. *Team Performance Management: An International Journal, Vol. 18 Issue: 1/2, 2012* (lk 59-73)
- Dey, P. K., Kinch, J., Ogunlana, S. O. (2007). Managing risk in software development projects: a case study. *Industrial Management & Data Systems, Vol. 107 Issue: 2, 2007* (lk 290-291)
- Gotel, O. C. Z., Finkelstein, A. C. W. (1994) An Analysis of the Requirements Traceability Problem. *Proceedings of IEEE International Conference on Requirements Engineering. Conference dates 18-22 April, 1994* (lk 98)

- Grey, S. (1995). Practical Risk Assessment for Project management. *John Wiley & Sons Ltd West Sussex, United Kingdom*, (1k xiii)
- Hajjdiab H., Taleb, A.S. (2011). Adopting Agile Software Development: Issues and Challenges. *International Journal of Managing Value and Supply Chains (IJMVSC) Vol. 2, No. 3, September 2011* (1k 2,7)
- Hall, T., Beecham, S., Rainer, A. (2002) Requirements problems in twelve software companies: an empirical analysis. *IEE Proceedings Software* , Vol. 149, No. 5, October, 2002 (1k 153-155)
- Hassine J. (2013) Describing and assessing availability requirements in the early stages of system development. *Software & Systems Modeling*., Vol. 14 Issue 4, October, 2013, (1k 1455)
- Hassine, J. (2015) Early modeling and validation of timed system requirements using Timed Use Case Maps. *Requirements Engineering*., Vol. 20 Issue 2, June 2015 (1k 182)
- Hatton, S. (2008) Choosing the “Right” Prioritisation Method. *19th Australian Conference on Software Engineering. IEEE, 2008* (1k 520, 522)
- Hoda, R., Babb, J., Norbjerg, J. (2013) Toward Learning Teams. *IEEE Software. Vol. 30 Issue 4, July 2013* (1k 97)
- Hoda, R., Babb, J., Norbjerg, J. (2014). Embedding Reflection and Learning into Agile Software Development. *IEEE Software*., Vol. 31 Issue 4, July 2014 (1k 56)
- Hoda, R., James, N. Marshall, S. (2013) Self-Organizing Roles on Agile Software Development Teams. *IEEE Transactions on Software Engineering. Vol. 39 Issue 3, March 2013* (1k 422)
- Hughes, D.L., Rana, N.P., Simintiras, A.C. (2017). The changing landscape of IS project failure: an examination of the key factors. *Journal of Enterprise Information Management, Vol. 30 Issue: 1, 2017* (1k 159)
- Huo, M. Verner, J. Zhu, L., Babar. M. A. (2004). Software Quality and Agile Methods. *Proceedings of the 28th Annual International Computer Software and Applications Conference. IEEE*
- Jones, R. L, Rastogi, A. (2004). Secure Coding: Building Security into the Software Development Life Cycle. *Information Systems Security. Vol. 13 Issue 5, Nov/Dec, 2004* (1k 33-34)
- Kannenberg, A., Saiedian, H. (2010) Why Software Requirements Traceability Remains a Challenge. *Journal of the Quality Assurance Institute. Vol. 24 Issue 2, April 2010* (1k 4).
- Karaman, K., Kurt, M. (2015) Comparison of project management methodologies: PRINCE2 versus PMBOK for it projects. *Int. Journal of Applied Sciences and Engineering Research, Vol. 4, Issue 4, 2015* (1k 578)
- Khalifa M., Verner, J.M. (2000). Drivers for Software Development Method Usage. *IEEE Transactions of engineering management, Vol 47, No 3, August 2000* (1k 360-361)
- Krishnan, M. S (1998) The role of team factors in software cost and quality: An empirical analysis. *Information Technology & People, Vol. 11 Issue: 1, 1998* (1k 32-33)

- Lesser, E., Ban, L. (2016) How leading companies practice software development and delivery to achieve a competitive edge. *Strategy & Leadership, Vol. 44 Issue: 1, 2016* (lk 44)
- Liang T., Liu, C., Lin, T., Lin, B. (2007) Effect of team diversity on software project performance. *Industrial Management & Data Systems, Vol. 107 Issue: 5, 2007* (lk 649)
- Lopez-Lorca, A., Beydoun G., Valencia-Garcia, R., Martinez-Bejar, R. (2016) Automating the reuse of domain knowledge to improve the modelling outcome from interactions between developers and clients. *Computing., Vol. 98 Issue 6, June, 2016* (lk 610)
- Matveus, J. (2009). Projekti „Ühtse euromaksete piirkonna krediitkorraluse analüüs“ firma x näitel. *Diplomi töö, Tallinn 2009*. PDF versioon, (lk 29) [<http://enos.itcollege.ee/~lepikult/diplomitoo/naidistood/Matveus.pdf>] kasutatud 12.04.2019
- Maylor, H. (2003). Project management. Third edition. *Pearson education Limited, Essex, England* (lk 28)
- McCormick, M. (2012). Waterfall vs. Agile Methodology. *Revised Edition 8/9. MPCS, Inc.* (lk 4,7)
- Morgan, M.J. (1994) Controlling Software Development Costs. *Industrial Management & Data Systems, Vol. 94 Issue: 1, 1994* (lk 14)
- Paetch, F., Eberlein, A., Maurer, F. (2003). Requirements Engineering and Agile Software Development. *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '03) Conference Linz, Austria, 11.06, 2003* (lk 1-2)
- Pass, S., Ronen, B. (2014). Reducing the Software Value Gap. *Communications of the ACM., Vol. 57 Issue 5, May, 2014* (lk 81)
- Pendharkar, P.C., Rodger, J.A. (2009). The Relationship between Software Development Team Size and Software Development Cost. *Communications of the ACM., Vol. 52 Issue 1, January 2009* (lk 141,144).
- Project Management Body of Knowledge (PMBOK) Guide [<http://www.itinfo.am/eng/project-management-body-of-knowledge-pmbok-guide/>] kasutatud 13.02.2019
- Project Management Fundamentals You Can Apply To Any Project [<http://www.project-skills.com/project-management-fundamentals-you-can-apply-to-any-project/>] kasutatud 13.02.2019
- Ruuska I., Vartiainen, M. (2003) Critical project competences – a case study. *Journal of Workplace Learning, Vol. 15 Issue: 7/8, 2003* (lk 310-311)
- Sawyer, S. (2004). Software development teams. *Communication of the ACM. Vol. 47 Issue 12, December 2004* (lk 96-97)
- Siegelau, M.J. (2004) How PRINCE2 Can Complement PMBOK and Your PMP [http://samuellearning.org/project_management_slides/prince2vspmbok.pdf] (lk 2, 3) kasutatud :27.01.2019

Subramanian, G. H., Klein, G., Jiang, J.J., Chan, C-L. (2009) Balancing Four Factors in System Development Projects. *Communications of the ACM., Vol. 52 Issue 10, October 2009* (lk 118-120)

Zarndt, F. (2011). Project management 101: Plan well, communicate a lot, and don't forget acceptance criteria, *OCLC Systems & Services: International digital library perspectives, Vol. 27 Issue: 3, 2011* (lk 171-173)

Zwikael, O., Meredith J. R., (2018) Who's who in the project zoo? The ten core project roles. *International Journal of Operations & Production Management, Vol. 38 Issue: 2, 2018* (lk 485)

Tellioglu, H., Wagner, I. (1997) Negotiating Boundaries. Configuration Management in Computer Supported Cooperative Work. *The Journal of Collaborative Computing. Vol. 6 Issue 4, November 1997,* (lk 259)

VanderLeest, S. H., Buter, A. (2009). Escape the Waterfall: Agile for aerospace. *28th Digital Avionics Systems Conference, October 25-29, 2009,* (lk 2)

Vatsa A., Kumar, S. (2016) Software Production Issues and Mitigation Techniques: A review. *International Journal of Recent Research Aspects., Vol. 3 Issue 2, June 2016* (lk 8)

Virine L., Trumper, M. (2008). Project decisions. *USA. Managment Concepts Inc* (lk 119, 123-124)

Webb, A. (2003). The Project Managers's Guide to Handling Risk. *Gower Publishing Limited, England* (lk 27- 28)

What is PMBOK in Project Management? [<https://www.visual-paradigm.com/guide/pmbok/what-is-pmbok/>] kasutatud 13.02.2019

White, A. S (2013). A control model of the software requirements process. *Kybernetes, Vol. 42 Issue: 3, 2013* (lk 424)

Wideman, R. M. (2002). Comparing PRINCE2 with PMBoK@ [<http://www.maxwideman.com/papers/comparing/comparing.pdf>] (lk 1, 3) kasutatud 27.01.2019

Yeong, A. (2012). The Marriage Proposal of PRINCE2™ and PMBOK [http://www.fsaeedi.ir/books/standards/OGC/PRINCE2-OGC/15.The_Marriage_of_PRINCE2_and_PMBOK.pdf] kasutatud: 26.01.2019

Juhtumianalüüsi intervjuu (2019):

Ettevõtte X juht. *Juhtumianalüüsi intervjuu.* 08.04.2019

Intervjuud (2019):

Ettevõtte X juht. *Intervjuu projekti teemal.* 02.04.2019

Ettevõtte X finantsjuht. *Intervjuu projekti teemal* 08.04.2019

Ettevõtte X müügi- ja turundusjuht. *Intervjuu projekti teemal.* 09.04.2019

Ettevõtte X personali- ja kvaliteedi juht. *Intervjuu projekti teemal.* 18.04.2019

Küsitlus (2019):

Ettevõtte administratiivtöötajad. *Küsitlus teemal rahulolu IT ja tarkvaraga.* 21.03-27.03.2019

Lisa 1 Küsimustik

1. Hinnake oma rahulolu IT toega 5 punkti skaalal (1 on kõige madalam hinne ja 5 kõige kõrgem) Põhjendage valikut.
2. Kas olete pöördunud viimase kuu jooksul IT toe poole? Mis teemadega olete pöördunud IT toe poole?
3. Hinnake oma rahulolu 5 punkti skaalal ettevõttes X kasutusel oleva tarkvaraga (1 kõige madalam ja 5 kõige kõrgem hinne) Palun põhjendage.
4. Kuidas olete rahul informatsiooni kättesaadavusega olemasolevast tarkvarast (1 kõige madalam ja 5 kõige kõrgem hinne)? Palun põhjendage.
5. Kas olemasolevatel tarkvaradel on mingi oluline funktsionaalsus puudu Sinu jaoks ? Palun põhjendage.
6. Kui vastasite 5. küsimusele jaatavalt, siis palun kirjeldage, kuidas soovitav lisafunktsionaalsus tarkvaras Teie tööd kergendaks?
7. Kas näete, et ettevõttel oleks vajalik soetada mõni lisatarkvara töö lihtsustamiseks või automatiseerimiseks? Kirjeldage seda.

Lisa 2 Intervjuu

1. Juhtkonnas on soov parendada olemasolevat tarkvara ning integreerida olemasolevad süsteemid. Küsitluses, millele vastasid lõppkasutajad, hinnati rahulolu praeguse tarkvara osas pigem heaks. Kas sellised tulemused mõjutavad edasist juhtkonna otsust?
2. Kirjeldage olemasoleva lahenduse suurimaid puudusi
3. Millist ärilist probleemi aitaks uus tarkvara lahendada? Millist väärtust see looks?
4. Kirjeldage ideaalis, kuidas toimiks ettevõtte põhiprotsessid infosüsteemi toel.
5. Kas ettevõtte töötajatel oleks võimalik iganädalaselt panustada projekti pidevalt testides funktsionaalsust ning arendusele nõudeid kirjutades?
6. Ettevõtte on ka varasemalt tarkvara hankinud /olemasolevale funktsionaalsust juurde ostnud? Hinnake, kuidas on varasemalt protsess toimunud?
7. Kas on olnud mingeid suuri takistusi?
8. Milline projekti etapp on Teie arvates kõike keerulisem ja riskirohkem edu või ebaedu osas – algatamine, planeerimine, teostamine, lõpetamine?
9. Kes peaks Teie arvates projekti meeskonnas olema ja miks? Milliseks võiks kujuneda projektimeeskonna suurus?
10. Kas ettevõttele sobiks see, kui nõuded on juba projekti alguses hästi määratletud ja teatakse, mida tellitakse või kui otsuseid tehakse tulemuste pealt projekti keskel? Põhjenda
11. Kas ettevõttes plaanitava projekti jooksul võib teha muudatusi projektis või pigem ollakse arvamusel, et valitud plaan tuleb teostada ning ebamäärust on vaja vältida?
12. Kas oleksite nõus tegema järeleandmisi ajas, kvaliteedis või eelarves kui projekt seda vajab või mitte? Milline neist on kõige olulisem saavutada?
13. Kas oleksite kvaliteedi nimel nõus nii eelarvet kui ka arenduseks kuluvat aega suurendama?

Lisa 3 Juhtumianalüüsi intervjuu

1. Kuidas idee sündis ja miks just see tarkvara?
2. Kui suurt eeltööd tehti tarkvara tehnilise külje uurimisega?
3. Kas projekti jaoks oli loodud meeskond?
4. Kas projekti jooksul loodi dokumentatsioon tarkvara kohta?
5. Kas dokumentatsiooni ajakohasus oleks juurutamise lihtsamaks teinud?
6. Õppetund – mida teha teisiti?