

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Informaatikainstituut

Infosüsteemide õppetool

**Infosüsteemi arendamine allhanke korras,
seadmete rendiga tegeleva ettevõtte näitel**

Magistritöö

Üliõpilane: Kristjan Kabin

Üliõpilaskood: 105138 IABMM

Juhendaja: Lea Elmik
Rein Paluoja

Tallinn
2014

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.

(kuupäev)

(allkiri)

Annotatsioon

Antud töö eesmärk on analüüsida tarkvaraarenduse protsessi ettevõttes, kel endal arendusmeeskond ja vastav *know-how* puudub. Tarkvara arenduse vajadus võib esile kerkida mistahes ettevõttes, kuid antud juhul tuuakse näiteks ettevõtte, kelle äriprotsessid on niivõrd erilised, et laiatarbe majandustarkvara neid katta ei suuda, seetõttu tuleb otsida alternatiive vertikaallahenduste ja „rätsepätööde“ vallast.

Konkreetsed probleemid mida lahatakse:

- Kuidas selgitada arendusvajadust
- Kuidas valida arenduspartner, kui taustainfot palju pole ja seegi vähene info, mis olemas, on väheväärtuslik ja/või subjektiivne
- Miks ja kuidas tuleb juhtida organisatsiooni tarkvara arendus- ja juurutusprotsessi ajal
- Kuidas edukalt korraldada tarkvara arendust, antud vallas varasemat kogemust omamata

Töö tulemuseks on meetodikad, nõuanded ning head tavad, mille järgi oskab igaüks äriprotsesside toetuseks vajaliku tarkvara arendust tulemuslikult koordineerida ja tulemust luua.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 70 leheküljel, 6 peatükki, 8 joonist, 4 tabelit ja 4 lisa.

Abstract

The aim of the work is to prove that any organization that has unique business process, that is not (fully) supported by a commercial off the shelf software package, could implement custom changes to the IT systems they are using. Without having any prior technical know-how of software development or knowledge of methods supporting it.

The problems to be dealt with in this thesis, are following:

- How to decide on the need for IT system / software development.
- How to choose a software developing partner in an outsourced vendor scenario, when there is very little concrete and objective information available about the vendor.
- Why, and how to successfully steer and control the organization during software development and implementation process.
- How to outsource software development successfully without having prior experience on the field.

The main results obtained are a set of best practices and methods, which could be followed to deliver a successful software development and implementation project with an outsourced developer.

The thesis is in Estonian and contains 70 pages of text, 6 chapters, 8 figures, 4 tables and 4 appendixes.

Lühendite ja mõistete sõnastik

[Lühend/mõiste]	<i>ERP – Enterprice resource planning</i> Ettevõtte ressursside planeerimine ehk majandustarkvara
[Lühend/mõiste]	<i>RFP – Request for proposal</i> Pakkumise kutse
[Lühend/mõiste]	<i>AHP – Analytic hierarhy process</i> Hierarhilise analüüsi meetod (Thomas L. Saaty)
[Lühend/mõiste]	<i>SLA – Service level agreement</i> Teenustaseme leping
[Lühend/mõiste]	<i>ITS – Issue tracking system</i> Veebipõhine süsteem probleemjuhtumite haldamiseks (ka.veebipõhine klienditoe süsteem, <i>ticket handling system</i>)
[Lühend/mõiste]	<i>BPMN – Business process modelling notation</i> Äriprotsesside modelleerimise keel
[Lühend/mõiste]	<i>Business Case</i> Äriline vajadus
[Lühend/mõiste]	<i>Best practice</i> parim praktika või hea tava
[Lühend/mõiste]	<i>Prestudy</i> Eelanalüüs, mis eelneb tegelikule analüüsile
[Lühend/mõiste]	<i>Outsourcing</i> Mingi konkreetse tegevuse usaldamine välisele partnerile; allhange
[Lühend/mõiste]	<i>Common sense</i> Terve mõistus

Jooniste nimekiri

Joonis 1. Kvaliteedi ja kogemuste kriteeriumite kaalud	29
Joonis 2. Kommunikatsioonikriteeriumite kaalud	30
Joonis 3. Grupiotsuse kaalud	31
Joonis 4. Põhikriteeriumite omavahelised kaalud	31
Joonis 5. Web-HIPRE täidetud kriteeriumite ja seoste mudel	32
Joonis 6. Arenduspartneri valiku lõpptulemus ilma grupiotsuseta.....	34
Joonis 7. Arenduspartneri valiku lõpptulemus koos grupiotsusega.....	35
Joonis 8. Arenduspartneri valiku lõpptulemus võrdsete põhikriteeriumi kaaludega.....	36

Tabelite nimekiri

Tabel 1. Arenduspartneri valikuprotsessi kriteeriumid	26
Tabel 2. Kandidaatide kriteeriumitele vastavus	33
Tabel 3. Neli põhjust, miks on inimesed muutuste vastu [12]	38
Tabel 4. Strateegiad muutustele vastuseisu ületamiseks	39

Sisukord

1. Sissejuhatus.....	10
1.1 Taust ja probleem.....	10
1.2 Ülesande püstitus	12
1.3 Metoodikad ja ülevaade tööst	13
2. Arendusvajaduse otsustamine.....	15
2.1 Meetodid arendusvajaduse defineerimiseks	15
2.2 Kuidas arendusvajadus sünnib	19
2.3 Arendusvajaduse kokkuvõte.....	20
3. Arenduspartneri valik.....	22
3.1 Kandidaatide valik, I etapp	23
3.2 Kandidaatide valik, II etapp.....	25
3.2.1 Kriteeriumid	26
3.2.2 Kandidaadid	32
3.2.3 Tulemuste analüüs	34
3.3 Arenduspartneri valiku kokkuvõte.....	36
4. Kliendi organisatsiooni ettevalmistamine ja juhtimine.....	37
4.1 Miks on vaja kliendi organisatsiooni ette valmistada	37
4.2 Kuidas kliendi organisatsiooni ette valmistada.....	38
4.2.1 Suhtlemine	39
4.2.2 Kaasamine	40
4.2.3 Juhendamine.....	40
4.2.4 Läbirääkimised	42
4.2.5 Sundimine	42
4.3 Juhtimisstrateegiate analüüs	42
4.4 Kliendi organisatsiooni ettevalmistamise ja juhtimise kokkuvõte.....	45
5. Arendusprotsess	46
5.1 Kas arendada või kasutada standardset funktsionaalsust	48
5.2 Nõuete kogumine ja esitamine.....	50
5.3 Leping.....	52
5.4 Arendamine.....	53

5.5 Testimine ja aktsept.....	55
5.6 Käivitamine ehk <i>Go-Live</i>	56
5.7 Tarkvaraprojekti järeltegevused.....	56
5.7.1 Hooldusleping:	57
5.7.2 Koolitused	58
5.7.3 Vana süsteemi väljalülimine / arhiveerimine	58
5.8 Juhtimisstrateegiad arendusprotsessis	59
6. Kokkuvõte	61
Summary	63
Kasutatud kirjandus	65
Lisa 1. Meetodid muutuste juhtimiseks organisatsioonis	67
Lisa 2. Hierarhilise analüüsimeetodi kriteeriumite kaalud.....	68
Lisa 3. Hierarhilise analüüsimeetodi grupiotsuse kaalud.....	69
Lisa 4. Microsoft Dynamics Sure Step ülevaade	70

1. Sissejuhatus

Allhange ehk *Outsourcing* on tänapäeval mingis mõttes edukuse võtmesõna. Ei ole ju mingit mõtet püüda olla spetsialist igal võimalikul ärialal, pigem tasub fokuseerida oma põhitegevusele ning usaldada muud, kõrvalised tegevused, kellelegi teisele, kelle jaoks need tegevused on igapäevased. Enamus ettevõtteid tänapäeval kasutab oma igapäevatoos mingisuguseid IT süsteeme, paljudel juhtudel õnnestub kasutada standardseid lahendusi, nõ pakitooteid riulilt, kuid harvad pole olukorrad, kus on vajadus millegi enama järele. Sellistel puhkudel tuleb arendada ettevõttespetsiifilist IT süsteemi. Konkreetne töö on ehitusseadmete rendiga tegeleva firma X näitel, ning toetub enamasti firma X kasutuses olevale majandustarkvara süsteemile ehk ERP [1] süsteemile MS Dynamics NAV [2]. Töös kasutatakse edaspidi majandustarkvara puhul lühendit ERP, mis tegelikult ka Eesti keeles ettevõtte ressursside planeerimise tarkvara lühendiks on.

Firma X on tegutsenud Eesti turul alates aastast 1994. Esimene ERP laadne lahendus juurutati firmas aastal 1999, mis oli kombinatsioon raamatupidamisprogrammist ja müügi- ning laoprogrammist. Suur samm ettevõtte IT-süsteemide arengus toimus aastal 2006, kui toimus üleminek „päris“ ERP süsteemile, mis koondas finantsi, ostu, müügi ja lao kokku ühte programmi. Turuolukord, tarkvara tormiline areng, aruandlusvajadus ning mõned muud nüansid tingisid uuesti vajaduse otsida veelgi paremat ERP süsteemi.

Sellela seoses käivitus projekt, mille raames juurutati firmas X aastal 2008 Microsofti ERP Dynamics NAV. Ettevõtte edasise arengu käigus liideti sama lahendusega, samasse andmebaasi veel firma X filiaalid Lätis ja Leedus, mis käivitati omal ajal väga tõhusate ja lühikeste (3-kuuliste) projektidega tähtaegselt, vastavalt 1. detsembril 2009 ja 1. märtsil 2010.

1.1 Taust ja probleem

Tänapäeval on IT süsteemid ja tarkvara iga ettevõtte argipäev ja ilma arvutite ning tarkvarata ei kujutataks äri tegemist ettegi. Ükskõik kas kaupmees või remonditöökoda, tootmine või büroo, IT süsteemidest pole pääsu, kui soov oma alal edukas olla.

Üldiselt on enamusi äritegevusi õnneks toetatud mitmete erinevate pakitoote-tüüpi tarkvaralahenduste poolt. Erinevaid tarkvarade tootjaid on palju ja seega valikuvõimalusi on

mitmeid, vastavalt kliendi nõudmistele ja vajadustele. Seega, reeglina on turul päris hea võimalus valida endale sobiv lahendus. Samas ei tohi unustada, et pakitoode on üldjoontes samalaadne kõigi kasutajate jaoks, st. loob kõigile kasutajatele samad eeldused eduks.

Nüüd kui keegi soovib oma alal olla edukam teistest, innovaatiline ja eesrindlik, siis tekib suure tõenäosusega olukord, kus standardtarkvara sellele kasutajale ei sobi ning tuleb tellida / teha „rätseplahendus“. Ei saa välistada ka olukordi, kus firma äriprotsess on niivõrd unikaalne, et vastavat protsessi järgivale standardtarkvarale polekski reaalselt turgu, kui ainult see üks eriline klient. Seetõttu sellist standardset tarkvara ei eksisteerigi.

Kliendispetsiifilise tarkvara arendus tänapäeval on levinud ja laialdane, näiteid ei pea otsima kuigi kaugelt. Praktiliselt iga omal alal edukas ettevõtte vajab mingil määral individuaalset lähenemist. Piisab kui lugeda pabermeediat, kus tihtilugu kirjutatakse mõne suurema tarkvarahanke venimisest või ebaõnnestumisest. Illustreerimaks antud väidet oleks markantseim näide välismaalt, California kohtute kohtuasjade juhtimise süsteem (*California Court Case Management System*), mille arendus algas aastal 2001 ja lõppes krahhiga aastal 2012. Raha kulutati suurusjärgus 500 miljonit dollarit, tulemust aga ei tulnudki [3]. Edulugudest räägitakse/kirjutatakse millegipärast harva.

Antud töö vaatab tarkvara arendusele peamiselt kliendi ehk tellija perspektiivist. Klient antud kontekstis on ettevõtte, kel pole oma isiklikku IT-osakonda või arendusmeeskonda, kogu juurdekuuluvaga. Heal juhul on palgal IT-spetsialist või –juht, kes hoiab serverid ja arvutid töökorras ning abistab kasutajaid igapäevaprobleemide lahendamisel ja on ühtlasi olemasolevate IT süsteemide peakasutaja. Tema siis vastutabki IT-süsteemide toimimise eest. Seega võiks antud töö huvi pakkuda väikesele või keskmise suurusega ettevõttele, kel oma majasisene tarkvara arendusmeeskond puudub, kuid kel plaanis mistahes IT süsteemi arendus välise partneriga.

Eraldi tahaksin välja tuua osapooled, kes suure tõenäosusega kliendi organisatsioonist arenduspartneriga suhtlema pannakse – finantsjuht, tehnikajuht-tootmisjuht, müügijuht, sekretär, raamatupidaja(d), klienditeenindaja(d), osakonnajuhataja(d) jne. Need on tavalised töökohad ja tavalised inimesed, aga keegi neist pole tingimata IT süsteemide arendaja, testija või projektijuht, pahatihti pole nad eriti kõrgetasemelised arvutikasutajadki. Siit koorub välja teinegi probleem – kliendi organisatsiooni ja arenduspartneri organisatsiooni vaheline suhtlus

arendustegevuse ajal tavalises Eesti ettevõttes. Usutavasti oleks käesolev töö kasulik õppematerjal kõikidele eelpool loetletud osalistele.

Käesolevas töö analüüside ja näidete juures jäädakse üldjuhul firma X organisatsiooni ja MS Dynamics NAV ERP juurde. Põhilised näited antud töös on ERP süsteemi juurutamise perioodist, aga ka hilisematest arendustest. Näitena toodud ettevõttes on tarkvaraarenduse allhankega tegeldud alates aastast 2001. Tellitavate arenduste maht näitab pigem kasvutendentsi, mis on seletatav organisatsiooni pideva arenguga, mis on tingitud mitmeid uusi suundi firma protsessides. Seega saab firma X näitel väita, et tarkvara arendus ettevõttes ei ole mingi möödunud nähtus vaid on pigem pidev protsess. See on iseenesest loogiline, kuna alati, kui muutub organisatsioon, muutuvad ka nõudmised IT süsteemidele. Firma X näitel saab väita, et pidevas muutumises, so. arenduses on nii firma organisatsioon kui ka IT süsteem(id).

1.2 Ülesande püstitus

Antud töö põhiline eesmärk on selgitada, kuidas arendada tarkvara, väikesest mikroarendusest kuni majandustarkvara erifunktsionaalsuse arendamise ja juurutamiseni ettevõttes, kus puudub vastav kompetents – nii konkreetseks arendamiseks kui ka arenduse tellimiseks väljastpoolt. Töö on jaotatud osadeks, mis kirjeldavad ja analüüsivad protsessi ja selle osi. Eks lähenemisi on mitmeid erinevaid ja ükski neist ei pruugi olla õigem kui teine, allakirjutanu nägemus, antud protsessist on alljärgnev:

Alustuseks on vaja langetada otsus arendusvajaduse üle.

Antud peatükis ja alajaotistes selgitatakse, analüüsitakse ja põhjendatakse erinevaid meetodikaid arendusvajaduse otsustamisel.

Järgmiseks sammuks välise parteriga tarkvaraarenduses on arenduspartneri valik.

Reeglina on turul mitmeid pakkujaid, kes mingil määral eristuvad teineteisest. Näiteks ERP tarkvara turul on ainuüksi Eesti tooteid mitmeid erinevaid, lisaks veel välismaised tootjad, kellel on omakorda mitmeid pakkujaid. Osad pakkujad on niivõrd mitmekülgsed, et pakuvad samal ajal mitme erineva tootja samalaadset toodet.

Kuidas juhtida kliendi organisatsiooni, nii, et tarkvaraprojekt õnnestuks?

Pahatihti on juhtkonna tasemel otsustatud, et mingi IT süsteem on vaja luua või välja vahetada, arvatakse, et see ongi kogu organisatsiooni juhtimine. Kuidas juhtida organisatsiooni aga nii, et selle IT süsteemi loomine-juurutamine võimalikult valutu ning efektiivne oleks ja võimalikult suure tulu ettevõttele tooks? Organisatsiooni puudulikult juhtides võib väga lihtsalt leida ennast olukorras, kus tarkvaraprojekt ei arene mitte soovitud suunas vaid kipub venima või ebaõnnestub täielikult – unustatakse fakt, et tellija organisatsiooni täita on reeglina võtmerollid.

Kuidas tarkvara tellida ja juurutada nii, et valminud lahendus ja valmimisprotsess oleks kliendi jaoks optimaalne?

Reeglina on kliendiorganisatsioon valdkonnaspetsiifiline, näiteks on toodud ettevõtte, kes on ekspert ja küllalt edukas ehitusseadmete rendi alal, aga kel pole erilisi teadmisi tarkvara arendusprotsessi komponentidest ja telgitagustest. Hea tulemus nõuab tulemuslikku koostööd tellija ja tarnija organisatsioonide vahel. Kuidas mõõta ja mõista, et toode, mille tarnija kliendi esitatud nõudmiste järgi tegi, on sama asi, mida klient, tellimust esitades, oma ettekujutustes nägi? Kuidas vältida olukordi, kus tagantjärele tuleb tõdeda, et arenduspartner on teinud toote, mis ei vasta kliendi vajadustele ja tellitule?

1.3 Metoodikad ja ülevaade tööst

Näites toodud ettevõtte loomuliku arengu käigus on arenenud välja ka uus tase arenduste tellimises ja tellimuste juhtimises, paslik oleks sellele metoodikale anda nimeks *common-sense* ehk nii öelda terve mõistuse metoodika. Oma metoodika väljakujunemise käigus ei olnud võib-olla tegevuste järjekord alati optimaalne, võib-olla jäi mingi tegevus tegemata, aga tulemusteni siiski jõuti. Kuna esialgu tellija organisatsioonis vastavad juhtimisprotsessid puudusid, siis paljuski lasti tarkvaraprojekte juhtida tarkvara tarnija organisatsioonil. Selliselt käitudes õnnestus, ehkki näiliselt, ka vastutus delegeerida täitja organisatsioonile.

Värskemad kogemused on saanud mõjutusi ka akadeemilisest maailmast, rakendatud on ka teaduslike metoodikaid ning tehtud jõudumööda võrdlusi ja analüüsi teaduslike- ning *talupojatarkuslike* teadmiste vahel.

Töö esimeses osas keskendutakse arendusvajaduse väljaselgitamise protsessidele. Arendusvajaduse üle otsustamisel kasutatud metoodikad on enamasti kogemuslikud. Püütud on vältida tarbetut bürokraatiat ja võetud arvesse, et protsessis osalejad ei pruugi omada

sügavaid arvuti- või projektijuhtimisealaseid teadmisi. Suuremate arenduste põhjendamiseks tuuakse sisse ka laiemalt tuntud ärijuhtimise otsustusmetoodikaid.

Töö teises osas käsitletakse arenduspartneri valiku võimalikku protsessi. Peatükis on kirjeldatud ja analüüsitud metoodikaid andmete kogumiseks ja subjektiivsete hinnangute andmiseks. Kogutud andmetest järeluste tegemisel kasutatakse Thomas L. Saaty poolt loodud hierarhilise analüüsi meetodit *Analytic Hierarchy Process – AHP* [4], mille abil on võimalik subjektiivsetest hinnangutest teha korrastatud järelusi, otsuseid.

Töö kolmandas osas, organisatsiooni muutuste juhtimise peatükis on analüüsitud erinevaid võimalikke meetodeid organisatsiooni juhtimiseks muutuste protsessi ehk antud kontekstis tarkvaraprojekti ja sellega seotud muutuste rakendumise ajal. Enamus selle peatüki sisust on ajendatud „Organisatsiooni käitumise“ kursuselt magistriõppe ajal.

Töö viimane osa fookuseerub tarkvara arendusprojekti protsessile, kliendi ehk tellija perspektiivist, ning tarkvara kvaliteedi hindamisele / kindlustamisele. Hindamist näitlikustatakse nõuete ja testidega. Esitatakse funktsionaalsed ja mittefunktsionaalsed nõuded ning testid. Suur rõhk on osutatud dokumentatsioonile ja juhenditele, et hiljem oleks võimalik tehtud arendust ka uuendada. Töös ei keskenduta ühelegi tarkvaraarenduse raamistikule. Viimases osas püütakse anda nõu, leidmaks tasakaalupunkti erilahenduse ehk nn. *rätseplahenduse* ja standardtarkvara vahel.

2. Arendusvajaduse otsustamine

Kas soovitakse uut funktsionaalsust või mingi olemasoleva funktsionaalsuse muutmist? Kes soovib? Kas esitatud soov on ühe inimese nõue või mitme osakonna kombineeritud nõue kõnealusel organisatsioonis?

Näitena kasutatavas organisatsioonis on võetud normiks, et iga tellitav arendus peaks olema äriiselt õigustatud, selleks tuleb iga tellitava arenduse puhul täita nn äriis vajaduse nõue (*business case* [5]) ehk äriis vajadust kirjeldav põhjus ja selgitus, mingis salvestatavas, taasesitatavas vormis, soovitava dokumendi kujul. Kui eksisteerib õigustatud äriis vajadus, siis järelikult tuleb nõutud funktsionaalsus kuidagi võimaldada, kui olemasoleva standardtootega antud äriis vajadust pole võimalik rahuldada, siis järelikult eksisteerib vajadus tellida arendus. Kui põhjendatud äriis vajadust ei suudeta sõnastada, siis järelikult pole vaja arendada.

Äriis vajadust võib mõista ja esitada mitmeti, see võib olla e-mail, millele tegevjuht vastab „OK, telli ära.“, samas võib see olla mahukas dokument, mis selgitab sügavuti, miks ja kellele uut funktsionaalsust on vaja, ning mida see täpselt peaks võimaldama teha.

2.1 Meetodid arendusvajaduse defineerimiseks

Kuna kõik arendusvajadused vajavad aktsepteerimiseks ja töösse laskmiseks äriis vajadust, siis peaks uue arenduse korral alustama äriis vajaduse kirjeldamisest. Meetodi valik sõltub eelkõige arendusvajaduse olemusest, milline eeldatav arendus peaks olema. Töö aluseks olevas ettevõttes on välja kujunenud alltoodud „arenduspaketid“ ja neile vastav metoodika arendusvajaduse kirjeldamiseks.

Mikroarendus, see on ERP'de puhul suhteliselt tavaline arendusvajaduse liik, mida tehakse tihti ja mida on lihtne lahterdada isegi ilma arendaja abita (ajakulu on prognoositav):

Eeldatav arendusmaht: 0,25-3h.

Arendusvajaduse sisu: Uus informatiivne väli mingi andmetabeli struktuuris või mõne olemasoleva andmevälja toomine olemasolevale vormile. Arendusvajaduse

püstitaja kohus on võimalikult täpselt kirjeldada soovitud funktsionaalsust ja oma sõnadega põhjendada selle vajalikkust.

Kes otsustab: Selliste arendusvajaduste üle otsustab ainuisikuliselt arenduste koordinaator, kellele arendusvajadus tuleb selgelt põhjendada.

Äriline vajadus: Kogemuslik statistika ütleb, et äriline vajadus on 95% juhtudel „kiirem töö“.

Koolitus: Eraldi koolitus pole harilikult selle arendusliigi puhul vajalik.

Miniarendus, ERP puhul samuti suhteliselt tavaline arendusvajadus, eriti uue ERP käivitusfaasis, kus jooksvalt leitakse, et siin-seal tuleb midagi arendada.

Eeldatav arendusmaht: 3-10h.

Arendusvajaduse sisu: Selle mahu sisse mahub näiteks mingi uue, väga lihtsa tekstiaruande loomine, mis kombineerib suurusjärgus 1-3 tabeli andmed. Arendusvajaduse kirjeldavad koostöös arendusvajaduse algataja ja arenduste koordinaator.

Kes otsustab: Arendusvajaduse üle otsustab väike komitee, liikmeteks arenduste koordinaator ja finantsjuht.

Äriline vajadus: Äriline vajadus on reeglina põhjendatud kui „osakonna aruandlusvajadus äriotsuste tegemiseks“.

Koolitus: Reeglina pole vajalik, piisab juhenddokumentatsioonist.

Midiarendus, selline arendusmaht on meie ettevõttes küll tavapärane, ent mitte igapäevane. Sellise mahuga saab juba mõningase uue funktsionaalsuse, mille ülesandepüstituse loomine võib tingida vajaduse arendaja ja tellija vaheliseks tihedamaks koostööks, nt sidesessioonide ja ka näost-näku kohtumiste näol.

Eeldatav arendusmaht: 10-30h, täpse mahu saab reeglina alles peale ülesande esitamist arendajale.

Arendusvajaduse sisu: ERP maailmas saab antud arendusmahu juures päris keerulise aruande või mingi lihtsamat tüüpi lisafunktsionaalsuse. Reeglina mahub sellesse

arendusliiki, näiteks välise teenusepakkuja andmemudeli põhjal vastava andmeekspordi loomine ja perioodilise andmete vahetuse automatiseerimine. Sellises mahus arendusvajadus kaalutakse enne põhjalikumalt läbi, kaasatakse kõik osapooled kellele arendus huvi võiks pakkuda, koostöös pannakse paika nõuded, et saada tulem, mis kõigile sobiks. Sellise arenduspaketi puhul tehakse koostöös arendusvajaduse kirjeldamise faasis selgeks uur osa esitatavatest nõuetest. Olenevalt ülesande sisust võib olla otstarbekas kaasata arendaja konsultant.

Kes otsustab: Lisaks arenduste koordinaatori ja finantsjuhi aktseptile vajatakse ka tegevdirektori aktsepteringut.

Äriline vajadus: Näiteks parem klienditeenindus, väiksemad kulud.

Koolitus: Koolitus on reeglina vajalik, kuna kasutajate ring on laiem, oluline on korrektne juhenddokumentatsioon.

Kesksuur arendus. Sellise mahuga arendus toob arvestatava uue funktsionaalsuse, sellise arenduse näidet kasutan ka antud töös edaspidi.

Eeldatav arendusmaht: 30-100h, täpse mahu saab reeglina alles peale ülesande esitamist arendajale.

Arendusvajaduse sisu: Sellise mastaabi arendusvajaduse sündi on analüüsitud peatükis 2.2, leheküljel 19.

Sellises arendusmahu korral luuakse alati koostöös arendaja esindaja ehk konsultandiga nn disainidokument, millest selgub täpne arendusvajadus. Siin on otstarbekas rakendada juba üldtunnustatud tööriistu, näiteks puudujääkide analüüsi (*gap analysis*). Väga tõhus abimees identifitseeritud arendusvajaduse kirjeldamiseks võiks olla ka teksti või graafilisel kujul esitatud, soovitud äriprotsessi kujutav mudel (*business process model, lühendatult: BPMN* [6]). Äriprotsessi mudelit saab luua nii tahvlile käsitsi kui kasutades mõnda paljudest äriprotsesside modelleerimise keele tööriistadest arvutil. Äriprotsessi mudeli kasuks räägib tema lihtsus ning üldine arusaadavus, vastavat mudelit oskab luua kui ka lugeda nii ärilise kui infotehnoloogia taustaga inimene. Loodav äriprotsessi mudel võiks olla aluseks arendusvajaduse täpsemale defineerimisele, kuna selle abil saab tuletada esialgsed nõuded soovitud arendatavale süsteemile.

Kes otsustab: Arendusvajaduse üle otsustab väike komitee, mis sisaldab endas arenduste koordinaatorit, arendusvajaduse algatajat ning finantsjuhti. Lõpliku otsuse langetab tegevdirektor, sellist arendust on juba otstarbekas juhtida kui projekti.

Äriline vajadus: Seadusandluse muudatus, olulise kitsaskoha kõrvaldamine süsteemis – vähem vigu, parem klienditeenindus.

Koolitus: Koolitus on reeglina vajalik, ainult juhenddokumentatsioonist kindlasti ei piisa.

Maksiarendus, sellises mahus arendused klassifitseerib näiteks toodud ettevõtte ERP arenduse valdkonnas suurteks arendusteks.

Eeldatav arendusmaht: Üle 100h.

Arendusvajaduse sisu: Siia alla mahub oluline funktsionaalsuse lisamine või olemasoleva tarkvara versiooniuuendus. Maksiarenduse puhul on allkirjutanu meelest samuti paslik kasutada töövahenditena puudujääkide analüüsi (*gap analysis*) ning äriprotsessi mudelit. Nende abitööriistade kasutamise tulemusel sünnib põhjendatud äriline vajadus arenduse täpsemaks formuleerimiseks.

Maksiarendust juhitakse alati kui formaalset projekti, juba seetõttu, et sellise mastaabi projekti käivitamiseks on vaja küsida luba emaettevõtte esindajatelt, kelle poolt on nõutud nii ärilise vajaduse olemasolu kui ka projekti plaan. Sellises mahus arendusi on näitena kasutatud ettevõttes reeglina igal aastal 1-2.

Kes otsustab: Maksiarendus otsustatakse alati juhtkonna tasemel, kuna mahud on suured, siis tuleb sellisele tarkvaraprojektile saada emaettevõtte aktsept.

Äriline vajadus: Äriliseks vajaduseks on nt versiooniuuendusega võimaldatud lisafunktsionaalsus, mis lihtsustab oluliselt olemasolevaid protsesse. Näiteks automatiseeritud elektrooniline ostuarvete käitlemine, mis vähendab raamatupidamise osakonna töökoormust ning kiirendab ostuarvete käitlemise protsessi 75%. Tulemuseks on parem rahavoogude planeerimine ning tähtaegselt tasutud ostuarved.

Koolitus: Koolitused on vajalikud ja tuleb hästi planeerida, kuna maksiaarendus puudutab tavaliselt suurt hulka tarkvarasüsteemi kasutajaskonnast.

Kuidas siis selgub, milline arenduspakett antud juhul sobib? Eelkõige peaks lähtuma tellitava arenduse olemusest. Kui on selge, mis liiki lisafunktsionaalsust vajatakse, saab juba kasutada iga arenduspaketi alla eraldi *arendusvajaduse* sisus toodud näiteid, mis aitavad uue arenduse liigitada. Siis selgub ka protsess edasiseks, kelle aktsepte vajatakse ja millises mahus- ning kujul tuleks kirjeldada äriplane vajadus.

2.2 Kuidas arendusvajadus sünnib

Antud peatükis näitlikustatakse ühe kesksuure arenduse sünni.

Oletame, et ettevõtte IT juht või keegi, kes ettevõttes tarkvaraarenduste koordinaatori rolli täidab, saab ettevõtte erinevatest üksustest järgnevaid signaale:

Klienditeenindus kurdab, et väga tülikas ja aeganõudev on rendile anda tellingukomplekte, kuna need koosnevad eraldi detailidest, mis kõik on rendisüsteemis arvel eraldi koodina. Sisuliselt tuleb igale rendilepingule sisestada iga detail eraldi uuele reale käsitsi, mis on väga tülikas ja aeganõudev, kas protsessi oleks kuidagi võimalik lihtsustada?

Umbes samal ajal kirub koosolekul tehnikaosakonna juht, et klienditeenindus teeb tellingukomplektide väljastamisel palju vigu. Tihti antakse lepingul välja komplekte, mis ei vasta tootja spetsifikatsioonile ja on potentsiaalne oht, et selle tõttu võib kasutajal juhtuda renditud tellingukomplekti kasutamise käigus õnnetus.

Lao inventuuride käigus on aga tuvastatud puudujääk, inventuuriaruande alusel peaks tellingutorni diagonaalsidemeid olema laos 117 tükki, lugemise käigus loeti vaid 84, puudujääk 33tk?

Korraldatakse uurimine, mille käigus selgub, et osad laod annavad tellingukomplekte rendile nime järgi, kui rendilepingul on nimi „Alumiinumtellingutorn 1,4m x 2,5m h=5m“, siis laotöötaja väljastab tootja spetsifikatsiooni järele õige komplekti, kui nüüd mingi detail on jäänud lepingule kandmata, siis seda ei märka keegi, realselt antakse kliendile välja õige komplekt, kuid lepingul on detailide kogused valed, halvemal juhul puudub lepingul mõni väljastatud positsioon sootuks. Mõni klient paneb seda tähele ja kasutab enda huvides ära, tagastades ainult need detailid, mis tema kätte lepingu alusel anti. Rendileping läheb peale tagastust suletud staatusesse, kuigi mingid detailid on kliendi käes. Nii tekibki puudujääk.

Tulenevalt eeltoodud näitest jõutakse rendifirmas arusaamiseni, et midagi on vaja muuta. Peale kõikide valikute kaalumist jõutakse otsuseni, et loogiline tundub olevat muuta IT süsteemi. Korraldatakse koosolekuid, toimub mitmeid ajurünnakuid (*brainstorming*), joonistatakse plaane, graafikuid ja mudeleid, analüüsitakse, kõige lõpuks sõnastatakse järgnev vajadus:

Rendisüsteemis puudub võimalus luua kaubakomplekte ehk malle, tuleb võimaldada olukord, kus saab defineerida komplektimalli iga tellingutorni komplekti kohta. Samuti tuleks võimaldada komplektimalli kasutamist rendilepingu koostamisel selliselt, et lepingu reale sisestatud komplektimalli koodi puhul oskab rendisüsteem ise komplektimalli kuuluvad kaubad lepingule lisada.

Kas eelkirjeldatud ülesandepüstitus on äriplane vajadus? Nagu oleks, aga tegelikult ei ole, eeltoodu on rohkem soovitud funktsionaalsuse kirjeldus. Äriplane vajadus peaks olema mõõdetav kasus, mida ta äriplane toob, näiteks kiiremas töös või säästetud rahas. Seega oleks äriplane vajaduse korrektne sõnastus pigem alljärgnev.

Rendisüsteemi on vaja luua võimalus, kus tellingutornide komplektid saab rendilepingule sisestada eeldefineeritud komplektikoodide alusel, mis välistaks olukorra, kus klienditeenindajad koostavad kiirustades vale kompleksusega rendilepinguid. Ühtlasi oleks seetõttu süsteemis vähem vigu ning rendilepingu koostamiseks kuluks vähem aega.

Lahenduskäik eeltoodud probleemile jätkub nõuete peatükis, leheküljel 50.

2.3 Arendusvajaduse kokkuvõte

Kokkuvõtvalt, arendusvajaduse määrab puudujääk olemasoleva tarkvara funktsionaalsuses, mis ei luba rakendada soovitud äriplane protsesse. Iga arendusvajadus peab olema äriplane põhjendatud – pole mõtet arendusele aega ja raha kulutada, kui see midagi vastutasuks ei too. Äriplane põhjendused võivad olla näiteks:

- Lihtsustatud tarkvarasisesed protsessid ja seetõttu vähem inimese poolt tehtud vigu süsteemis
- Mõttetu rutiinse töö pealt aja kokkuhoid, kus töötaja saaks teha midagi tulemuslikumat
- Suurem käive, kasum või kliendirahulolu

- Seadusandluse nõue
- Süsteemi vigade parandus (turvalisuse tõstmine)

Seega arendusvajadus on oma olemuselt midagi eelanalüüsi ehk *prestudy* laadset, mille põhjal langetatakse otsus, kas alustada tegeliku analüüsiga juba arendusprotsessi raames või mitte.

Seega leidke põhjus ja kirjeldage äriiline vajadus, sellega on arendusvajadus enamasti tõestatud ja põhjendus süsteemi arendamiseks olemas.

3. Arenduspartneri valik

Et protsessiga nii kaugemale jõuda, peaks ettevõttel olema kujunenud nägemus, mida arendatavalt süsteemilt oodatakse ning millisel platvormil see peaks toimima. Antud töö aluseks kasutatud näite puhul oli platvormi otsus eelnevalt langetatud. Mõjuvad põhjused rääkisid ühe konkreetse tootja kasuks.

Majanduslikult otstarbekas oli valida Microsofti Dynamics NAV, kuna Microsoft pakkus just sel ajal olulist allahindlust klientidele, kes vahetavad oma teatud tootja ERP tarkvara Microsofti Dynamics NAV vastu. Microsoft tegi sõna otseses mõttes allahindlust olemasoleva ERP litsentsi maksumuse võrra.

Teine mõjuv põhjus oli kolmanda osapoolse vertikaallahenduse olemasolu MS Dynamics NAV'le, mis toetas juba standardis väga suurt osa (vähemalt 90%) soovitud äriprotsessist ja funktsionaalsusest, milleks oli seadmete rentimise funktsionaalsus. Seega jäi üle leida ainult sobiv arenduspartner, kes arendaks-juurutaks puuduoleva ning lokaliseeriks ja tulevikus pakuks ka tuge.

Kolmas põhjus ja põhiline valiku põhjendus oli aga turg ise, nimelt valiku tegemise ajal ei olnud turul arvestatavat alternatiivi. Ainuke alternatiiv oluks midagi arendada nullist, kuid selleks polnud näitena kasutatud ettevõtte valmis.

Arenduspartneri valik sellisele projektile, mis mõjutab firma igapäevatööd otseselt, on reeglina otsus, mida ei tasu teha kergekäeliselt, kuna partner tuleks valida pikaajaline, võibolla kogu tarkvara elutsükli pikkune. Seetõttu tasub teha väga kaalutletud otsus, ei maksa oma otsust rajada ainult ühele-kahele valikukriteeriumile. Samas on see punkt kus konkreetset alati sobivat, õiget vastust leida kõige raskem, kuna soovid, vajadused ja ka võimalused on väga erinevad.

Antud peatüki järgnevates osades käsitletakse arenduspartneri valiku protsessi näiteks võetud ettevõttes. Antud juhul otsiti arenduspartnerit, kes juurutaks ettevõttes väljavalitud ERP standardtoote ning täidaks lisaks alltoodud funktsioonid:

- Välismaa ettevõtte toodetud seadmete renti võimaldava vertikaaltoote Eesti oludele lokaliseerimine ja antud toote esindamine Baltikumis

- Nimetatud vertikaaltoote juurutamine koos standardse ERP'ga
- Nõutud, kuid puuduoleva funktsionaalsuse arendamise juurutusprojekti raames
- Kõikide kasutajagruppide kasutajate koolitamise juurutusprojekti käigus
- Igapäevase kasutajatoe pakkumine nii standardtoodetele kui ka teostatud eriarendustele
- Olla arenduspartner ka tuleviku arenduste perspektiivis

3.1 Kandidaatide valik, I etapp

Esimene tegevus arenduspartneri leidmiseks oleks koostada ja välja saata pakkumise kutse (*RFP* [7]). Tegelikult oleks eelnevalt vaja teada, kellele kutset saata. Siin tuleb appi tootja, kellelt tasub küsida referentse ja iseloomustusi. Microsoft Eestilt õnnestuski saada nimekiri firmadest kes MS Dynamics NAV'i Eestis müüvad ja arendavad.

Pakkumise kutse võib olla traditsiooniline või vähem traditsiooniline, antud näitel kasutati tagumist varianti, nimelt, ei küsitud konkreetset pakkumist kogu juurutusprojektile vaid küsiti lihtsalt mõned küsimused:

- Turul tegutsemise aeg – aastates; aitab ettevõtte jätkusuutlikkuse määramisel ja annab vihje varasematest kogemustest
- Töötajate arv ettevõttes:
 - o Arendajate arv; aitavad ettevõtte suurust hinnata
 - o Konsultantide arv; aitavad ettevõtte suurust hinnata
 - o Projektijuhide arv; aitavad ettevõtte suurust hinnata
- Tehtud tööd, referentsid; annavad aimu kandidaadi kogemustest ja saab küsida tagasisidet ettevõtetelt, kellega neil koostöö on olnud
- Kõige suurem juurutus – suurim töökohtade arv; annab ettekujutuse firma kogemustest suurte projektide elluviimisel

- Varasemad kogemused kolmanda osapoole vertikaali juurutamisel MS Dynamics NAV keskkonnas; annab infot varasemate kogemuste kohta vertikaali tootjaga suhtlemisel ja vertikaaltoote esindamisel
- Prognoositav töö maht tundides kogu juurutusprojektile, varasemast kogemusest lähtuvalt; sooviti eelarvestamise abivahendiks
- Prognoositav koolituste maht meie kasutajaskonnale, varasemast kogemusest lähtuvalt; sooviti eelarvestamise abivahendiks
- Tunnihind projekti raames ja edaspidi; sooviti eelarvestamise abivahendiks, aga ka valikuotsuse langetamiseks, soodsam hind oluks eelistatud
- Mille poolest on teie teenus või ettevõtte eriline teistest, kes Eesti turul tegutsevad; sooviti teada konkreetse ettevõtte väärtushinnanguid, mida oluliseks peetakse, aitasid kujundada subjektiivseid seisukohti ettevõtte pihta

Kogutud andmeid plaaniti hiljem kasutada võrdleva analüüsi koostamisel, et parimad kandidaadid välja valida. Ainuisikuline valik oli välistatud, nimelt oli välismaise vertikaali tootja poolt seatud lisatingimus, nende kui tootja huvi oli omakorda valitud kandidaatidega tutvuda, et valida välja parim kandidaat nende toodet Eestis ja võimalusel kogu Baltikumis kohalikele oludele vastavaks lokaliseerima, esindama ja müüma.

Positiivse üllatusena saime vastuste saabuma hakkamisel lisaks veel ühe valikukriteeriumi, nimelt mõned pakkujad vastasid meile otse lihtsalt e-mailiga juba samal või järgmisel päeval, eelnevalt meie vastu mingit huvi üles näitamata. Samas kui teised võtsid ühendust, uurisid kui kiiresti pakkumist vajatakse ja esitasid täpsustavaid küsimusi, enne kui omapoolse pakkumise korrektselt dokumenteerituna esitasid. On vist liiane ära mainida, et e-kirja sees koheselt vastanutega edasist diskussiooni ei alustatud.

Põhjused, miks me ei korraldatud nõ fikseeritud hinnaga pakkumise küsimist olid:

- Oskamatus iseseisvalt vormistada korraliku, korrektset ja kõiki detaile hõlmavat nõuete dokumenti
- Teadmatus, mis on tarkvaral olemas ja kohe kasutatav standardses tarkvarapakettis ning mis tuleks juurde arendada

- Soov siduda ennast arenduspartneriga pikemalt kui ainult juurutusprojekt, kuna varasemast kogemusest teiste tootjate ERP-dega oli selge, et arenduspartneri abi on vaja jooksvalt ka edaspidi, hoolduslepingu raames.

Õigustatult võib küsida, miks ei pöördutud teadmiste ja kogemuste puudumise tõttu välise konsultandi poole, kes vajadusel pakkumise küsimise staadiumi korrektselt oleks toimetanud? Põhjuseks oli soov end mitte siduda välise konsultatsioonifirmaga, kelle taust ja motiivid oleks omakorda olnud küsimärgiga. Elu on ühtlasi õpetanud, et unikaalse äriprotsessiga ettevõttes on konsultandi abi majandustarkvara vahetuse programmis raske rakendada, pigem kulub konsultandil pikk aeg, et kõike toimuvat hoomata ja mõista. Võib juhtuda, et kuitahes põhjalikul olemasoleva olukorra kaardistamisel jääb konsultandil midagi olulist märkamata.

Lisaks ettevõtte kogemused varasemast, mis julgustasid ka nüüd omapäi tegutsema ja õigupoolest ei tulnud välise konsultandi palkamine isegi jutuks. Lõppude-lõpuks oli ka juurutamise ning arendamise protsess keskmisest keerulisem, kuna oli põhitoode, oli lisatoode vertikaali näol ja oli organisatsioon, kel süsteemile omad nõudmised, mis nõuavad detailset spetsifikatsiooni – mille koostamiseks oluks kõige otstarbekam kasutada väljavalitud arenduspartneri konsultantide abi. Selleks aga tuli endale arenduspartner valida.

Kandidaatide valiku I etapi lõpuks oli laual kolm reaalselt pakkujat, kes oma näitajatelt tundusid, et võiksid kvalifitseeruda. Kõigi poolt oli koostatud vormikohane pakkumise dokument, kõigiga oli näost-näku kohtunud. Järgmine etapp oli teha olemasolevate andmete pealt otsus ja sõlmida valitud partneriga leping.

3.2 Kandidaatide valik, II etapp

Peale taustaandmete kogumist tuleks nende andmete alusel langetada otsus kellegi kasuks. Olenevalt projekti mahust ja olulisusest tuleks partneri valiku protsessi suhtuda täie tõsidusega. Esmaste riskide hajutamiseks on tihti peale juba omanike või ematettevõtte poolt kehtestatud nõue, et kõik tarnijad-pakkujad peavad läbima taustakontrolli ja vastama teatud kriteeriumitele – nt turul tegutsemise aeg ja/või majanduslik seis, majandusaasta aruande põhjal. Kuna vastustena kogutud andmed ei omanud otsest sõltuvust võimaliku arenduspartneri edukuse ja tulemuslikkusega, siis otsustati potentsiaalselt parima valiku tegemiseks kasutada USA matemaatiku Thomas L. Saaty ca. 40 aastat tagasi loodud nn hierarhilise analüüsi meetodit (*AHP* [4]).

3.2.1 Kriteeriumid

Saaty meetodi kasutamiseks tuleks defineerida kriteeriumid, antud näites sai hulk sobivaid kriteeriume kogutud eelmises etapis pakkumise küsimise käigus ja kujunesid ka suhteliselt subjektiivsed hinnangud võimalikele partneritele nendega kohtumiste ajal. Kasutatud kriteeriumid on toodud tabelis: Tabel 1

Tabel 1. Arenduspartneri valikuprotsessi kriteeriumid

Põhikriteerium	Alakriteerium	Kriteeriumi kood
kvaliteet ja kogemused		
	aastaid turul	Vanus
	konsultantide arv	Konsultandid
	arendajate arv	Arendajad
	projektijuhtide arv	Projuhid
	varem juurutatud projektid	Projektid
	kõige suurem juurutus- töökohtade arv	Suurim
	pakutav tunnihind projekti ajal ja edaspidi	Tunnihind
Kommunikatsioon		
	Eelistatud suhtluskanal projekti käigus	Suhtluskanal
	Eelistatud suhtluskeel projekti käigus	Suhtluskeel
	Arendajate-konsultantide füüsiline asukoht	Asukoht
Subjektiivne grupiotsus		
	Subjektiivne hinnang ettevõttele (pakkumise ja presentatsiooni põhjal)	Hinnang

Esmaseks kriteeriumiks sai **kvaliteet ja kogemused**, selle kriteeriumi alamkriteeriumite tähtsust hinnati paari kaupa võrdluse meetodil järgnevalt:

Vanus: Ettevõtte turul oldud aeg hinnati tähtsamaks kui tunnihinne ja sama oluliseks kui projektijuhtide arv, kuid vähemtähtsaks kui konsultantide, arendajate, sooritatud projektide arv ja suurim projekt.

Miks oluline: Turul oldud aja järgi saab hinnata ettevõtte jätkusuutlikkust muutavas majandusolukorras.

Hindamiskriteeriumid: Vahe 0 kuni 2a - hinnang 1, vahe 3-5a - hinnang 3, vahe üle 5a - hinnang 5.

Konsultandid: Konsultantide arv oli vähemtähtis kui pakutav tunnihinne, ent pisut tähtsam kui ettevõtte vanus, muud alamkriteeriumid jäid konsultantide arvuga samaväärseks.

Miks oluline: Sooviti olukorda, kus oleks garanteeritud üks konkreetne konsultant, kes tunneks kõnealust lahendust läbi ja lõhki, samas eeldati, et ettevõttes on konsultante rohkem kui üks, kes vajadusel saaks toote kasutajatoe üle võtta.

Hindamiskriteeriumid: Vahe 0 kuni 2 konsultanti - hinnang 1, vahe 3-5 konsultanti - hinnang 3, vahe üle 5 konsultandi - hinnang 5.

Arendajad: Arendajate hulka hinnati olulisemaks ettevõtte vanusest, projektijuhtide hulgast, sooritatud projektide hulgast ja prognoositud tööde mahust, muid alamkriteeriume peeti arendajatega samaväärseks.

Miks oluline: Varasemast oli kogemus, kus ettevõttes oligi ainult üks inimene, kes antud lahendust tundis, olles korruga konsultant, arendaja ja projektijuht. Sooviti sellisesse olukorda uuesti mitte sattuda.

Hindamiskriteeriumid: Vahe 0 kuni 1 arendaja - hinnang 1, vahe 2-4 arendajat - hinnang 3, vahe üle 4 arendaja - hinnang 5.

Projehid: Projektijuhte peeti sama olulisteks kui ettevõtte vanust, konsultantide arvu, varem läbiviidud projekte ja tunnihinnet, kuid vähem olulisteks kui arendajate arvu ja suurimat projekti.

Miks oluline: Sooviti, et kõnealuse tarkvara juurutusprojektiga tegeleks üks konkreetne inimene pühendunult, ettevõtte kel ainult üks projektijuht ongi, seda endale lubada ei saaks.

Hindamiskriteeriumid: Vahe 0 kuni 1 projektijuht - hinnang 1, vahe 2-3 projektijuhti - hinnang 3.

Projektid: Läbiviidud projekte hinnati olulisemaks kui ettevõtte vanus kuid vähemoluliseks arendajate arvust, muus osas oli see kriteerium võrdne ülejäänutega.

Miks oluline: Sooviti valida ettevõtte, kel kogemus reaalsete juurutusprojektidega.

Hindamiskriteeriumid: Vahe 0 kuni 3 projekti - hinnang 1, vahe 4-10 projekti - hinnang 3, vahe üle 10 projekti - hinnang 5.

Suurim: Suurimat läbiviidud juurutust hinnati olulisemaks kui ettevõtte tegutsemisperioodi, projektijuhtide arvu ja tunnihinnet. Muude kriteeriumitega peeti suurimat juurutust võrdseks.

Miks oluline: Enda suurst teati juba ette, sooviti vältida arenduspartnerit, kelle jaoks pakutav juurutusprojekt ei oleks piisavalt suur ja oluline. Ühtlasi sooviti teada kui suurte juurutusetega on kogemusi.

Hindamiskriteeriumid: Vahe 0 kuni 5 kasutajat - hinnang 1, vahe 6-15 kasutajat - hinnang 3, vahe üle 16 kasutaja - hinnang 5.

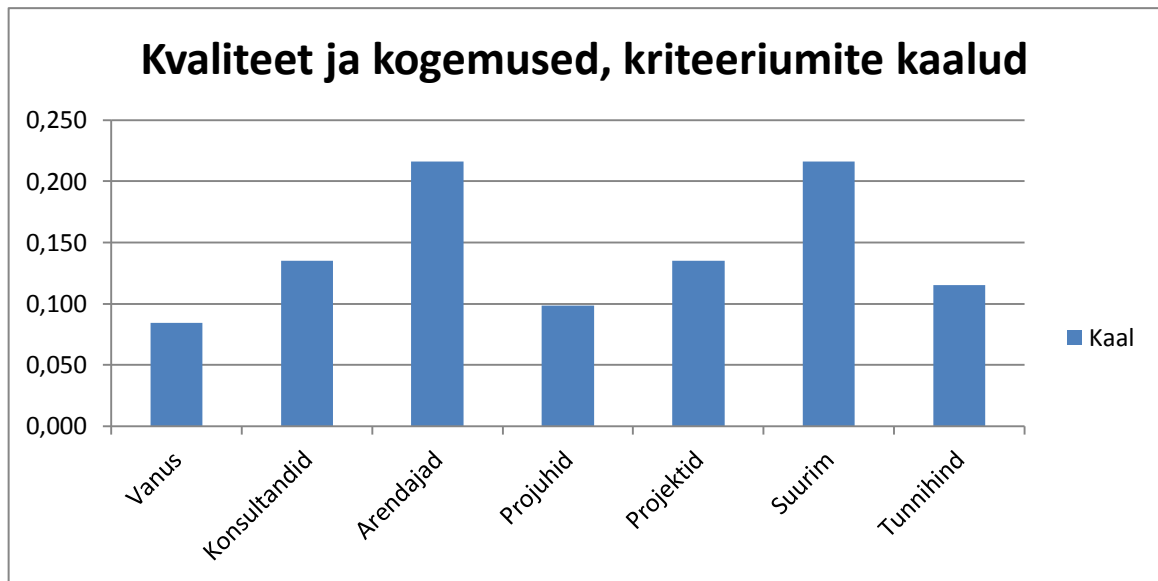
Tunnihind: Pakkumises välja toodud tunnihinnet peeti olulisemaks kui konsultantide arvu, muude kriteeriumitega peeti teda samaväärseks, ent ettevõtte vanuse ja suurima juurutatud projekti osas peeti pakutud tunnihinda vähem oluliseks.

Miks oluline: Kuigi tunnihind ei ole kuidagi subjektiivne, kaasati tunnihind valikuprotsessi, kuna see mõjutab otseselt juurutusprojekti hinda ja ka hilisemaid arendusi.

Hindamiskriteeriumid: Vahe 0 kuni 3€ - hinnang 1, vahe 3-5€ - hinnang 3, vahe üle 5€ - hinnang 5.

Täpsed valikuprotsessi kaalud leiti võrdlustabeliga, võrdlustabel on toodud lisas: Lisa 2. Hierarhilise analüüsimeetodi kriteeriumite kaalud, mis leitav leheküljelt 68.

Valikuprotsessi kvaliteedi ja kogemuste kriteeriumite kaalud graafiliselt väljendatuna on joonisel: Joonis 1:



Joonis 1. Kvaliteedi ja kogemuste kriteeriumite kaalud

Teiseks kriteeriumiks sai **kommunikatsioon** ehk kuidas toimub ettevõtetes tavaliselt suhtlus kliendiga ning kus paiknevad ettevõtte konsultandid ja arendajad, oluliseks peeti kolme alamkriteeriumit, mille kohta infot eksisteeris, meetodiks jällegi paari kaupa võrdlemine.

Suhtluskanal: Peeti vähemoluliseks kui suhtluskeelt.

Miks oluline: Varasemast kogemusest lähtuvalt sooviti vältida arenduspartnerit, kellel puudub veebipõhine kliendivaatega veajälgimiste süsteem probleemide lahenduste haldamiseks (*Issue tracking System* [8]), kuna telefoni ja e-mailiga edastatud probleemide lahendust on keeruline jälgida (*follow-up*). Juhtumid võivad lihtsalt ununeda.

Hindamiskriteeriumid: Pakkujad kellega suhtlus käib läbi kliendi poolt ligipääsetava veebipõhise veajälgimise süsteemi, ehk *Issue tracking System* [8] laadse süsteemi, saavad hinnanguks 3 võrreldes pakkujaga kellel vastav süsteem puudub.

Suhtluskeel: Peeti olulisemaks kui asukohta või suhtluskanalit.

Miks oluline: Kuna juurutusfaasis on suhtlemist palju ning mitteemakeelne suhtlus võib põhjustada olulisi möödarääkimisi, mis võivad kaasa tuua projekti venimise ja kulude kasvu.

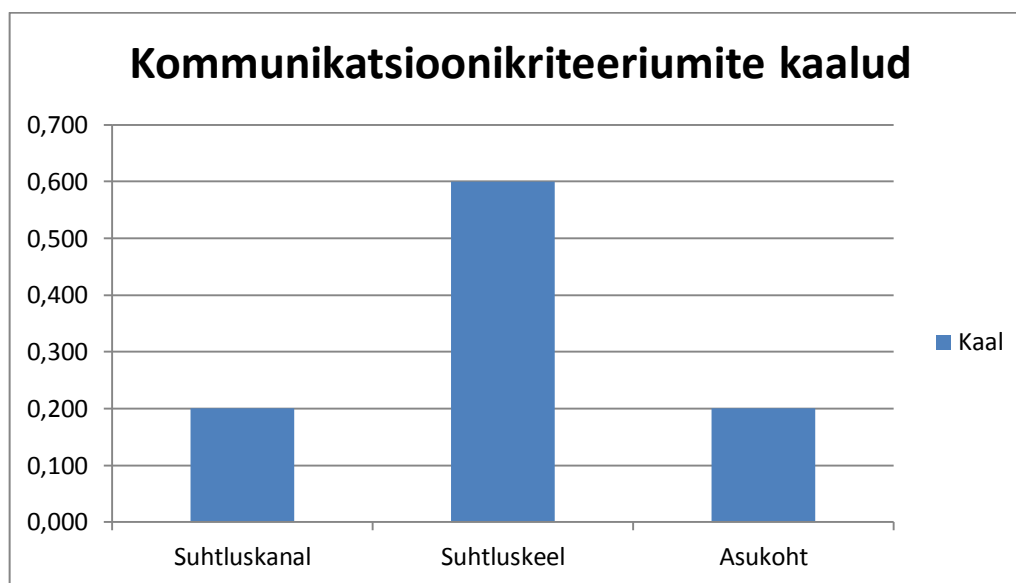
Hindamiskriteeriumid: Pakkujad kellega saab kõigil teemadel suhelda eesti keeles, saavad eelise (3p) võrreldes pakkujaga, kes eestikeelset tuge ei paku.

Asukoht: Peeti suhtluskeelest vähem oluliseks.

Miks oluline: Eelistati arenduspartnereid, kellel füüsiline kohalolek võimalikult lähedal, et vajadusel saada kiiret lokaalset „abi“.

Hindamiskriteeriumid: Pakkujad kellel on kontor Eestis, hinnang 3 võrreldes pakkujaga kellel on kontor väljaspool Eestit.

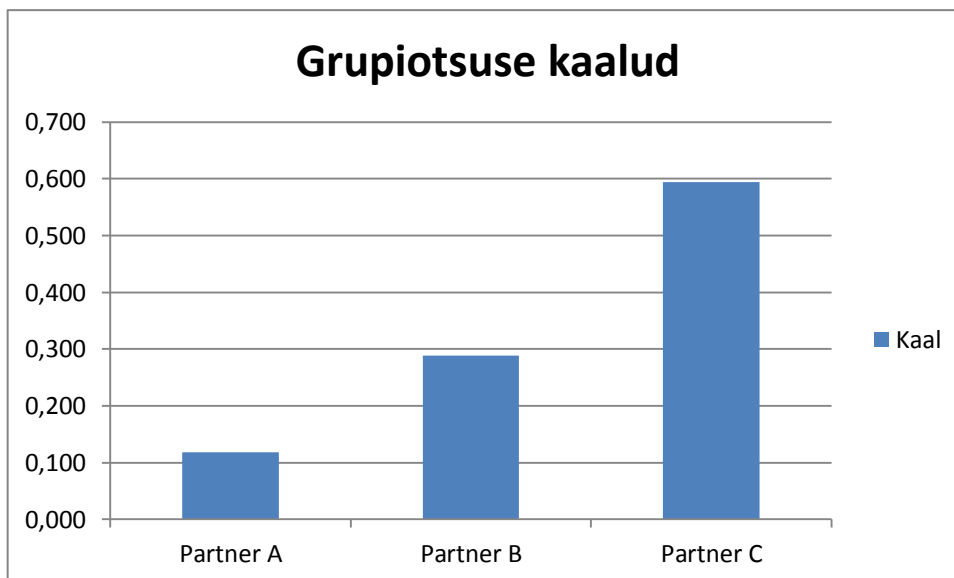
Ka kommunikatsioonikriteeriumite kaalud leiti võrdlustabeliga, mis leitav lisast: Lisa 2. Hierarhilise analüüsimeetodi kriteeriumite kaalud, leheküljel: 68. Graafiliselt kujutatuna on kommunikatsioonikriteeriumite kaalud kujutatud joonisel: Joonis 2.



Joonis 2. Kommunikatsioonikriteeriumite kaalud

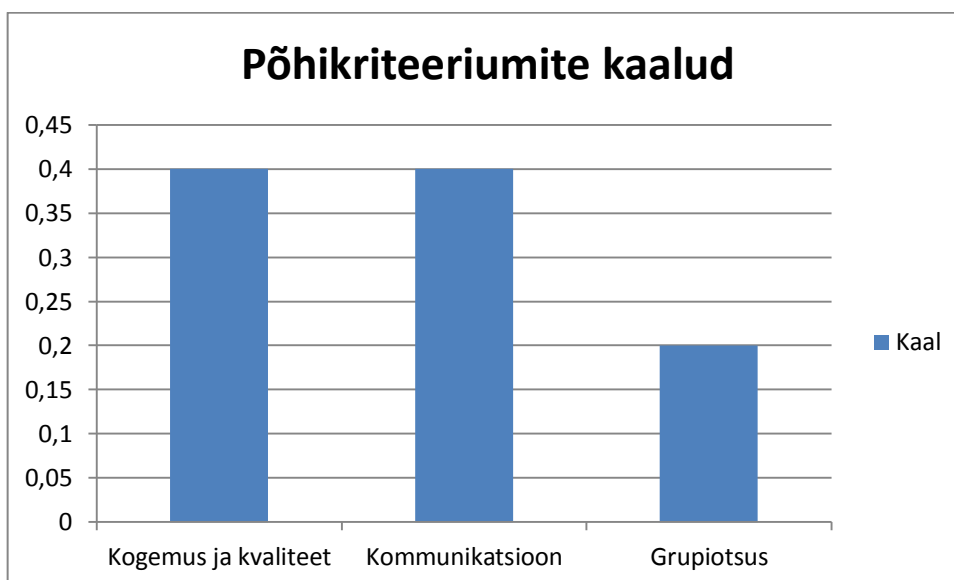
Kolmandaks kriteeriumiks oli subjektiivne **grupiotsus**, millest võtsid osa ettevõtte juhid: finantsjuht, IT juht ja tegevdirektor. Grupiotsuse sisuks oli partnerite subjektiivne paari kaupa võrdlemine võttes aluseks esitatud pakkumise dokumendi ja füüsilise kohtumise käigus tekkinud emotsioonid. Sellise grupiotsuse tähtsust ei maksa alahinnata, kuna otsuse tegemiseks on ju kasulikke andmeid suhteliselt vähe ja need ei ole otsuse langetamiseks kuigi „tugevad“. Seetõttu võib selline mitme inimese subjektiivsetel hinnangutel tehtud otsus aidata realselt õige otsuse langetamisel.

Osaliste võrdlustabelite lõppkaalud liideti kokku geomeetrilise keskmisega ning teostati normaliseerimine, millest kujunes nõ. grupiotsus. Grupiotsuse kaalude tabelid on töö lisas: Lisa 3. Hierarhilise analüüsimeetodi grupiotsuse kaalud, leheküljel 69. Graafiliselt kujutatult on grupiotsuse tulemus näha joonisel: Joonis 3.



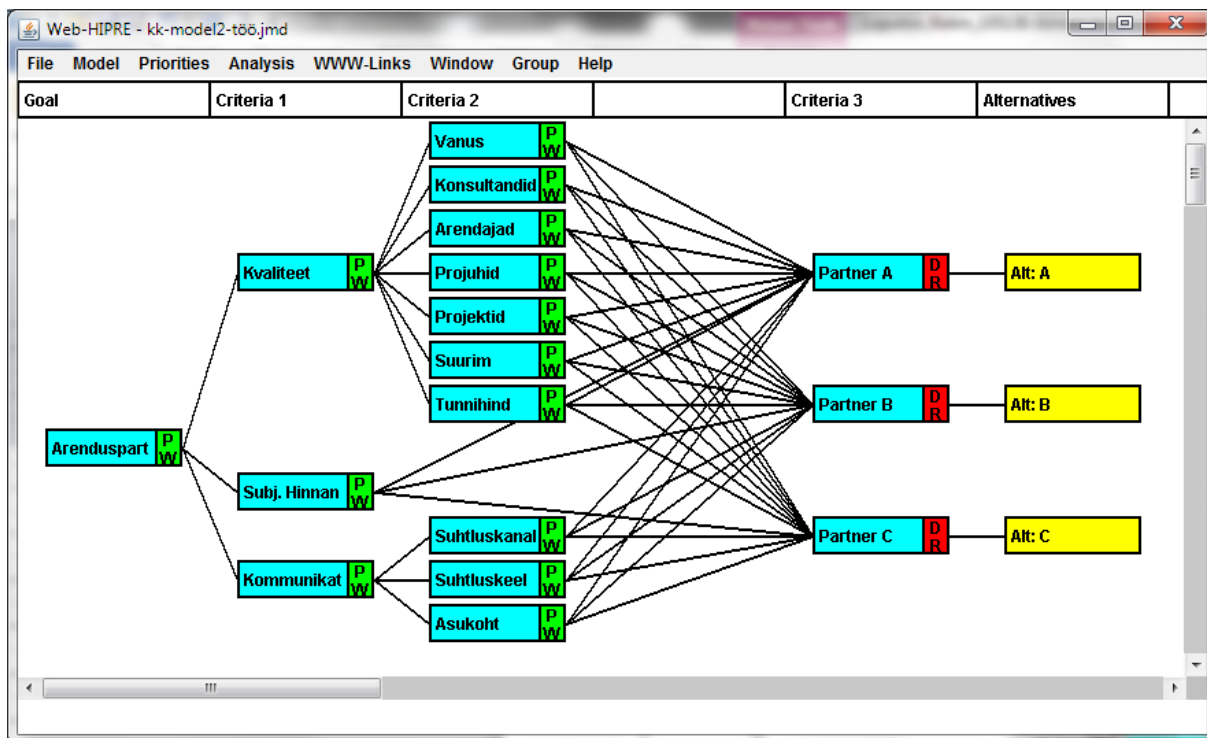
Joonis 3. Grupiotsuse kaalud

Põhikriteeriumite kaalude määramisel kasutati samuti paarikaupa võrdlust, mille tulemusel hinnati kvaliteeti 2/5 väärtuses kaaluga, sama kaalu sai suhtluskriteerium, kõige subjektiivsem ehk kohtumise ja dokumentatsiooni pealt tehtud grupiotsust hinnati kaaluga väärtuses 1/5 kogukaalust. Sellest tulenev põhikriteeriumite kaalude jaotus on näha graafikul: .



Joonis 4. Põhikriteeriumite omavahelised kaalud

Kriteeriumite ja tulemuste visualiseerimiseks kasutati Aalto Ülikooli poolt majutatud tööriista Web-HIPRE [9]. Web-HIPRE kriteeriumite mudel on toodud joonisel: Joonis 5



Joonis 5. Web-HIPRE täidetud kriteeriumite ja seoste mudel

Jooniselt joonistuvad selgelt välja kriteeriumite ja valikute seosed, kõik kriteeriumid olid omavahel võrreldud paaritise võrdluse meetodil, mida tähistab kriteeriumi taga näha olev „PW“ (*PW – pairwise comparison*).

3.2.2 Kandidaadid

Teise ringi kandidaatideks otsustati jätta kolm võimalikku partnerit:

Partner A – välismaa päritolu ettevõtte, kellel harukontor müügiosakonnaga ka Eestis

Partner B – globaalne ettevõtte, kellel harukontor koos arendusmeeskonnaga ka Eestis

Partner C – kohalik ettevõtte, peakontor Tallinnas

Alternatiivide täpsemad andmed on toodud tabelis Tabel 2. Eraldi tabeli kujul antud kaalude leidmist, sarnaselt grupiotsuse kaalude leidmisega, alternatiivide vahel ei teostatud, vaid kasutati tabelis Tabel 2 kajastuvaid andmeid ning teostati paaride võrdlus vahetult Web-HIPRE süsteemi täites, vastavalt hindamismetoodikale, mis kirjeldatud peatükis 3.2.1.

Tabel 2. Kandidaatide kriteeriumitele vastavus

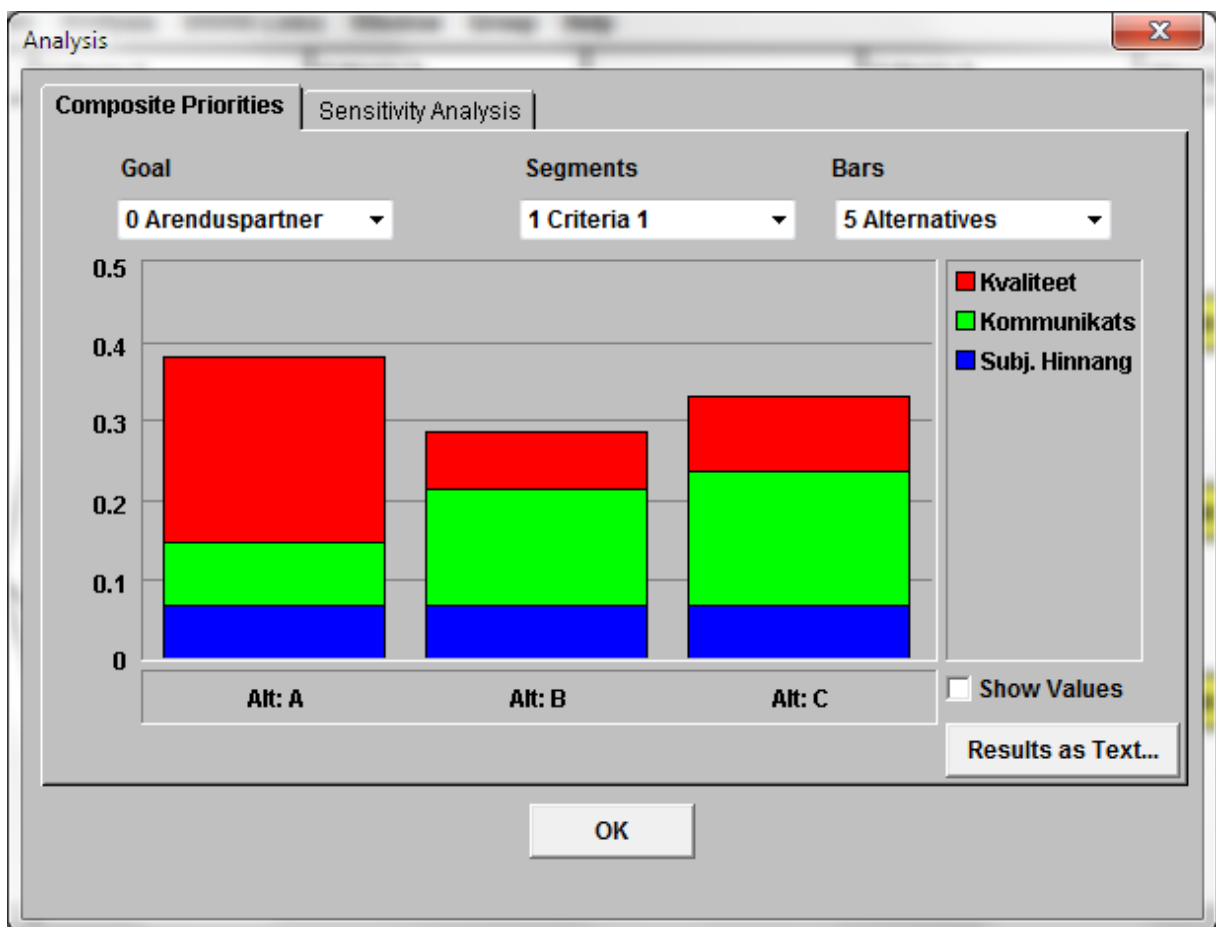
Kriteerium	Partner A	Partner B	Partner C
Vanus	5	10	13
Konsultandid	4	2	6
Arendajad	23	12	8
Projehid	6	3	3
Projektid	120	8	60
Suurim	200	100	110
Tunnihind	48	55	50
Suhtluskanal	its/e-post/tel	e-post/tel	its/e-post/tel
Suhtluskeel	Inglise	Eesti	Eesti
Asukoht	Välismaa	Tallinn	Tallinn

Tabelist selguvad peamiste valikukriteeriumite väärtused, näeme, et toodud andmed on küllaltki väheolulised, nende pealt ainuõiget otsust langetada saab olema keeruline. Samas, paaritiste võrdluseks sobivad need andmed küll, nt paistab kohe välja, et välismaise päritoluga firma, partner A, on kõige suurem, probleemina, aga on nende Eesti esindus tegelikult väga väike ja seetõttu on ka kommunikatsiooni kriteeriumil suhteliselt suur kaal.

Tunnihindade suurusjärk on suhteliselt sarnane, kuid, kui võtta aluseks, et projektis kulub tõenäoliselt tuhandeid tunde, siis muutuvad isegi mõne eurosed tunnihinna vahed automaatselt tuhandeteks eurodeks.

3.2.3 Tulemuste analüüs

Tulemuste analüüsi läbiviimiseks kasutame Web-HIPRE võimalusi kuvada esitatud andmeid visuaalselt. Esimese tulemusena näidatakse mudelit, kus sisestatud on kõigi kriteeriumite kaalud va. kliendiorganisatsiooni juhtide subjektiivne hinnang – grupiotsus. Joonisel, Joonis 6, on selgelt näha, et defineeritud kvaliteedikriteeriumid soosivad alternatiivi A ehk arenduspartneri A valimist. Samas on kommunikatsioonikriteeriumite järgi selge eelis arenduspartneril C.



Joonis 6. Arenduspartneri valiku lõpptulemus ilma grupiotsuseta

Peale grupiotsuse lisamist muutub lõpptulemuse pilt oluliselt. Joonisel, Joonis 7, leheküljel 35 on näha kuidas grupiotsuse lisamisel muutub parimaks valikuks alternatiiv C. Alternatiiv A kaotab grupiotsuse lisamisel kõige enam, nende tulemused olid kõige nõrgemad, põhjused, võib oletada, olid seotud kandidaadi välismaise päritoluga.



Joonis 7. Arenduspartneri valiku lõpptulemus koos grupiotsusega

Kuna kolme põhikriteeriumi kaalud on küll leitud paaride võrdluse meetodil, ent paaride võrdluse tulemused on otseses sõltuvuses võrdluse teostaja isiklike hinnangutega ehk samuti subjektiivsed, siis oleks õige teostada ka vastav tundlikkuse analüüs. Web-HIPRE võimaldab tundlikkuse analüüsi teostada iga kriteeriumi suhtes ja seda saab teha jooksvalt, ilma alusandmeid muutmata. Esitan mõned näited, näitamaks, mis juhtub valikutega, kui muuta põhikriteeriumite kaalud võrdseks ehk, et kõik põhikriteeriumid on sama olulised. Vaata tulemust joonisel, Joonis 8, leheküljel 36. Selgelt on näha, et alternatiivi C ülekaal teiste ees kasvas veelgi ning alternatiiv B lähenes oluliselt alternatiivile A.



Joonis 8. Arenduspartneri valiku lõpptulemus võrdsete põhikriteeriumi kaaludega

3.3 Arenduspartneri valiku kokkuvõte

Käesolevas peatükis rakendati erinevaid meetodikaid, Saaty meetod, gruppotsus ja tulemuse visualiseerimine, leidmaks sobivat arenduspartnerit. Põhimõtteliselt saab samu kriteeriume ja meetodikaid kasutada ka tulevikus, arenduspartneri valiku tegemise puhul. Toodud näites osutus valituks partner, kes oli 100% fokuseerunud ainult MS Dynamics NAV tootele. Samuti oli väljavalitud kõige suurem kliendibaas Eestis, kuid tegemist polnud Eesti suurima juurutuse läbiviijaga, samas oli neil kogemusi suurte juurutustega (>50 töökoha) juba minevikust. Valitud arendajal oli küllalt suur arendajate ring ning parim iseloomustus Microsoft Eestilt. Tagantjärele saab tõdeda, et kliendi ja tarkvarapartneri vaheline koostöö on kestnud aastaid ja kestab edasi, koostöö olemusest võib järeldada, et rahul on mõlemad pooled. Tulenevalt eelöeldust võib peatüki kokku võtta, et tehtud valikud ja kasutatud meetodikad õigustasid end.

4. Kliendi organisatsiooni ettevalmistamine ja juhtimine

Edukas tarkvaraprojekt nõuab kõigi poolte motiveeritust, nii kliendi organisatsioonis kui ka arenduspartneri organisatsioonis. Eriti kui lähtuda eeldusest, et tarkvaraprojekti eesmärgiks on asendada kliendi olemasolev ERP süsteem, kus tehakse kõik igapäevased ja kriitilised äritegevused. Arenduspartneri motiveerimiseks on mõningad tööriistad, peaaesjalikult on selleks leping ja teenustaseme leping ehk *SLA* [10]. Kliendi organisatsiooni positiivseks meeletamiseks tuleks aga läheneda muul moel. Ei ole haruldane, kus suurema IT projekti korraldamisel jääb kliendi organisatsiooni motiveerituse tase siiski liiga madalaks, see on seletatav tõsiasjaga, et kuigi inimesed reeglina on nõus end muutma, siis tegelikkuses muudatusi ei soovita ja pigem eelistatakse vanaviisi tegutsemist [11]. Selline vähese motiveerituse ignoreerimine juhtkonna ja/või projektijuhi poolt võib tarkvara arendusprojektis väga valusalt kätte maksta.

4.1 Miks on vaja kliendi organisatsiooni ette valmistada

Mis on ohtudeks tarkvaraarendusprojektile, kui kliendi organisatsioon on korrektselt ette valmistamata? Ohud on seotud muutustega, mida organisatsioon ei pruugi ühehäälselt tervitada. Uued tarkvaralahendused toovad reeglina muudatused töös, mis alati tingimata ei vähenda reatöötaja kohustusi vaid võivad kohustuste hulka hoopis suurendada. Enne uut tarkvaralahendust tuleb aga selle arendus- ja juurutusprojekt, mis olenevalt mahust võivad olla tellija organisatsioonile väga kurnavad. Juurutusprojekt nõuab alati ka tellija organisatsiooni osalemist, olenevalt projekti liigist võib see osalus olla ainult koolitus, aga ka kogu protsess, alates lahenduse nõuete defineerimisest ja disainist läbi reaalse töö, kuni koolituste ja käivitamiseni, mis võib muuhulgas endas sisaldada nt käsitsi andmete migratsiooni. Eelöeldust võib järeldada, et tellija organisatsioon võib muutusele ohtu kujutada olles osaliselt või isegi täielikult muutusele vastu. Et nende vastuseisudega tegelda, tuleks nad alustuseks identifitseerida. Kotter ja Schlesinger oma 1979 a töös „Choosing Strategies for Change“ [12] on identifitseerinud neli peamist põhjust, miks töötajad muutusi ei tervita. Kõnealused neli põhjust koos detailsema kirjeldusega on toodud tabelis: Tabel 3 leheküljel 38.

Tabel 3. Neli põhjust, miks on inimesed muutuste vastu [12]

Põhjus	Kirjeldus
Isiklikud huvid	Inimesed kardavad kaotada võimu, palka või töökohta
Teadmatus ja valearusaamad	Kommunikatsioonipuudus, infopuudus
Muutuste soovimatus	Inimesed on vanas kinni, vana on tuttav ja turvaline
Erinevad arusaamad muutuste põhjustest	Alati on inimesi, kel on muutustest oma isiklik arusaam, mis ei ühti üldisega

Järgmises alapeatükis vaadatakse põgusalt võimalikke strateegiaid muutuste vastuseisu murdmiseks.

4.2 Kuidas kliendi organisatsiooni ette valmistada

Kuidas siis neid muutuste vastuseise lahendada? Kotter ja Schlesinger oma 1979 a töös „Choosing Strategies for Change“ [12] on välja pakkunud mõned strateegiad, kuidas vastuseisu murda, vt. Lisa 1. Meetodid muutuste juhtimiseks organisatsioonis, leheküljel 67, millest omakorda olen loonud tarkvara arendusprojekti muutustega seoses sobivad meetodid ja praktikad, mis on loetletud tabelis: Tabel 4 leheküljel 39. Tabelis on ka lihtsad autori soovitusel, kuidas ja millal neid võiks rakendada, täpsemad soovitusel ja muutuste juhtimise strateegiate analüüs on toodud alapeatükis 4.3 leheküljel 42.

Tabel 4. Strateegiad muutustele vastuseisu ületamiseks

Strateegia	Millal kasutada
Suhtlemine, selgitamine	Selgita muutuste vajadust ja loogikat, et ei saaks tekkida kuulujutte ja väljamõeldisi
Kaasamine	Kaasa inimesed muutuste protsessi võimalikult varakult, need kes on osaks protsessist on väiksema tõenäosusega selle vastu
Juhendamine	Kui inimestel on hirm, et uus süsteem või selle juurutusprotsess on keeruline, siis tuleks inimesed ära kuulata ja võimalusel juhendada
Läbirääkimised	Juhul kui muutuste tõttu on inimestel hirm, et organisatsioonis on oodata suuri muutusi, siis selgitage muudatusi täpsemalt ja vajadusel leppige kokku tingimustes
Sundimine	Kui on vaja tegutseda erakordselt kiiresti, aitab juhtkonna tasemel otsus – tuleb teha ja kõik – parema tulemuse saab siiski nii, et muutuste protsessis osaletakse omal initsiatiivil

Järgmises alapeatükis vaadatakse iga muutusstrateegiat lähemalt koos detailsema kirjeldusega ning püüdes strateegiaid näidata enam tarkvaraarenduse kontekstis.

4.2.1 Suhtlemine

Kui otsus muutusteks on tehtud, siis peaks esimese asjana looma kommunikatsiooniplaani. Kommunikatsioon on muudatuste puhul hädavajalik tegevus. Vastasel juhul on suur oht väljamõeldiste ja kuulujuttude tekkeks – sõltub loomulikult organisatsiooni suuruselt. Ettevõttesiseseid kommunikatsioonivahendeid võiks ära kasutada vastavalt olemasolule: koosolekud, intranet, e-maili listid, uudisteleht, teadete tahvel jne. Oluline on oma kommunikatsioonis selgitada muutuste vajadust ja loogikat, samuti võimalikku ajakava. Antud informatsioon peab kindlasti jõudma kõigi asjassepuutuvateni.

4.2.2 Kaasamine

Keskse IT süsteemi, nagu seda on ERP, vahetusel on oluline ja hädavajalik olemasolevate andmestute ülevaatus ja korrastamine, täiesti igapäevased on juhtumid, kus suvalistes tabelites, nt kliendi- või tootetabel, on andmevälju ära kasutatud mittesihipäraselt. Näiteks võib kergematel juhtudel olla segi aetud telefoni, faksi ja/või e-posti väljad, ent ebaharilikud pole ka olukorrad, kus kliendi nimi on näiteks kirjutatud aadressiväljale ja postiindeks sattunud asula väljale.

Siin tulebki mängu kogu organisatsiooni kaasamine, otstarbekas on andmetabelid puhastada-korrastada juba vanas süsteemis, viia andmed korrektseks ja eraldi märkida ära need andmed (nt. pankrotistunud kliendid, müüdnud ja maha kantud seadmed ja põhivara), mida uude süsteemi üle kanda ei soovita. Nii väheneb migratsioonile kulutatav aeg ja maht ja lõpptulemuseks on puhtam, korrektsem ja kiirem andmebaas uues IT süsteemis.

Kes mureseb, et sellise „valikulise“ migratsiooni käigus võib vajalik ajalugu kaotsi minna, siis praktikas kõiki ajaloolisi andmeid ei migreerita uude süsteemi pea mitte kunagi, st reeglina säilib vajadus hoida mõnda aega uus ja vana süsteem paralleelselt töös. Et vana süsteemi ülalpidamine liialt kulukaks ei muutuks, hoitakse teda reeglina arhiveeritult või paigaldatuna kuskile failiserveri „nurka“, kust ta siis vajadusel ellu saab äratada ja vajalikke andmeid kontrollida. Täpsemalt vaata peatükk 5.7 leheküljel 56.

Andmekorrastustegevusse saab reeglina kaasata kogu puudutatud organisatsiooni, kuna igäühel neist on oma osa IT süsteemi andmestutes ja seega peaks igäüks neist vastutama, et nende vastutusala andmed saaksid korrastatud.

4.2.3 Juhendamine

Juhendamine on üldiselt antud kontekstis hädavajalik, kuna kokku on saanud väga erinevad inimesed, ühelt poolt kliendiorganisatsioon oma töötajatega, kes on reeglina haritud raamatupidajad, klienditeenindajad ja juhid ning teiselt poolt konsultandid, arendajad ja projektijuht. Kas ja kuidas on võimalik panna efektiivset ja tõhusat koostööd tegema, ühelt poolt kliendi organisatsioon, kes pole ülikoolis 5 aastat infotehnoloogiat ega projektijuhtimist õppinud ega tea midagi arendusmetoodikatest või isegi elementaarsest ülesandepüstitusest. Teiselt poolt aga arenduspartner, keda hirmutavad ja ajavad segadusse kliendiorganisatsiooni käibefraasid: „bilanss“, „EBIT“ ning „klient ootab leti taga“ ning kes vihkab veakirjeldusi stiilis „see aruanne ei tööta“, „finants ja moodul ei lähe kokku“ või kõigi veakirjelduste kroon:

„Navi ei tööta“. Lihtsalt ära märkimiseks, isiklikust kogemusest, viimane väljend võib tähendada:

- a. Ongi mingi tõsine probleem:
 - a. Kliendarvutil puudub võrk ja terminaliga ei saada ühendust
 - b. Arendaja on parandanud mingi kriitilise tarkvaravea (*bugi*), kuid ilma funktsionaalsust testimata ehk unustanud objekti koodi kompileerimata ja seetõttu vastav funktsioon ei tööta
 - c. Navi ongi katki, andmebaas maas, serveriga probleemid, võrguprobleemid
- b. On lihtsalt arusaamatus:
 - a. Kasutaja on end mingite väljade peale „lukku“ ajanud
 - b. Kasutajal pole numברי lukustusklahv (*numlock*) sisse lülitatud, seetõttu numbrikliaviatuurilt (*numpad*) ei tule numbrid, vaid kursor hüpleb kaootiliselt ekraanil ringi
 - c. Trükkimisele saadetud dokument ei jõua mingil põhjusel printerisse
 - d. Mingi funktsioon ei tööta ootuspäraselt, kuna laosaldo pole piisav, õigused ei luba funktsiooni käivitada, tehakse ebaloogiline tehe, nt. üritatakse tagastada renditooteid mineviku kuupäevaga vms, kasutajale väljastatakse süsteemiteated, ent neid ignoreeritakse
- c. Litsentsjärgne kasutajalimiit on täis ja rohkem kasutajaid süsteemi ei lubata
- d. Arendatud on vigane komponent, mis põhjustab programmi iseeneslikku sulgumist
- e. Jne jne jne.

Sellest tulenevalt järeldub, et tarkvara arendusprojektis oleks tarvilik koolitada kliendiorganisatsiooni neid liikmeid kes projektis otseselt osalevad.

4.2.4 Läbirääkimised

Juhul kui vastuseis muutusele organisatsioonis on ikkagi tugev, tasub kaaluda läbirääkimiste taktikat. Läbirääkimiste eesmärgiks saab olla kompromiss. Võimalik, et plaanitud muudatus etteantud mahus polekski reaalne ja osaliste nõudmised on seetõttu õigustatud. Kompromiss on reeglina kahepoolne, mis peaks rahuldama mõlemat osapoolt (kõiki osapooli).

Teine põhjus läbirääkimisteni jõudmisel on lisakoormus. Võib juhtuda, et ettevõtte töötajad leiavad, et süsteemiarendusega seotud tegevused võtavad neilt ära palju ressursi, alluvate tööaega, nende enda tööaega, tõstavad organisatsiooni stressitaset ja teevad inimesed närviliseks, kuna aega napib ja kohustusi on palju. See omakorda võib põhjustada häireid igapäevatoös, kannatada võivad nii konkreetse töö kvaliteet kui ka suhted klientidega jms.

4.2.5 Sundimine

Sundimise taktikat kasutatakse juhul kui muutustega organisatsioonis on väga kiire ja vastuseis on väga tugev. Sundimise alla kuulub näiteks juhtkonna selgesõnaline käsk, st, sundimist saab kasutada ainult siis, kui muutusele on selge tugi ja huvi olemas ka organisatsiooni kõige kõrgemates kihtides. Sundimise strateegiat saab kasutada ka kaudselt, nimelt ähvardades, näiteks, töölt vabastamine, üleviimisega teisele kohale, ametikõrgendusest ilmajätmisega jms. Sundimisstrateegia rakendamisel tasub arvestada, et „ohvritele“ jääb see meelde ja vastava muutuse algataja võib jääda organisatsioonis ebasoosingusse. Isiklikust kogemusest võin öelda, et harvad pole juhud, kus sellised muutuste läbiviijad oma alluvate poolt hiljem organisatsioonist „välja aidatakse“. See võib toimuda väga kiiresti (kuude ja isegi nädalatega).

4.3 Juhtimisstrateegiate analüüs

Järgnevalt analüüsitakse strateegiaid individuaalselt, leides küsimused vastustele:

Kuidas strateegiat kasutada – analüüsitakse ja kirjeldatakse protsessi võimalike strateegia rakendustega

Millal strateegiat kasutada – analüüsitakse, millal muutuste protsessis oleks kõige otstarbekam strateegiat rakendada

Oodatav tulemus – milliseid otseseid tulemusi peaks strateegia rakendamine andma

Tugevused – analüüsitud strateegia tugevused

Puudused – analüüsitakse antud strateegia puuduseid

Järgnevalt on esitatud organisatsiooni juhtimise strateegiate analüüsid.

Suhtlusstrateegia

Kuidas kasutada: koosolekud, e-post, intranet, näost-näku, posterid

Millal kasutada: muutuse planeerimisel, võimalusel juba enne tarkvaraprojekti algust aga ka projekti käigus

Oodatav tulemus: informeeritud töötajad elavad muutusesse sisse kergemini, tarkvaraprojekt ei tule üllatusena, ollakse progressiga kursis

Tugevused: vähese vaevaga suur tulemus

Puudused: kommunikatsiooni sisu peab muutuma vastavalt projekti edenemisele, st nõuab päris palju aega ja tööd

Kaasamisstrateegia

Kuidas kasutada: andmete korrastamine, nõuete kogumine, testimine, koolitused

Millal kasutada: muutuste/projekti käigus

Oodatav tulemus: protsessis osalenud inimesed aitavad õiget asja teha, ise osaledes ollakse muutusega meelsamini nõus ja ilmselt nakatatakse entusiasmiga ka teisi organisatsioonis

Tugevused: kaasamine töötab erakordselt tõhusalt, kuna veendunud kaasatutest saab projekt tohutult kasu ning muudab elluviimise efektiivsemaks

Puudused: võib tekitada probleeme aja planeerimisel, igapäeva tööülesanded *versus* muutustega seotud projektiülesanded, prioritseerimine väga oluline

Juhendamisstrateegia

Kuidas kasutada: tuleb identifitseerida puudujäägid mida juhendamisega katta, seejärel vastavad koolitused organiseerida

Millal kasutada: projekti alfaasis või isegi enne projekti algust

Oodatav tulemus: juhendatud on muutusega meelsamini nõus ja teavad mida neilt oodatakse

Tugevused: ilmselt kõige efektiivsem muutuste juhtimise strateegia, kui plaanitud muutused muudavad paljusid protsesse, st inimesed peavad õppima teistmoodi töötama

Puudused: võib osutada väga kulukaks ja ei garanteeri tingimata head tulemust

Läbirääkimiste strateegia

Kuidas kasutada: erimeelsuste lahendamine kompromissi teel

Millal kasutada: kui on reaalne oht, et muutuste protsessis osalemine häirib tavapäraste tööülesannete täitmist

Oodatav tulemus: leitakse kompromiss igapäevaste tööülesannete ja muutuste protsessi ülesannete vahel, võimalik, et osad positsioonid organisatsioonis muutuvad

Tugevused: küllalt tõhus meetod jäiga vastuseisu murdmiseks

Puudused: kuna inimeste valulävi ja iseloomud on erinevad, siis võib tekkida olukord, kus keegi saab läbirääkimiste käigus väiksema töökoormuse, keegi teine, kes pole kurnud, peab aga tegema rohkem tööd

Sundimise strateegia

Kuidas kasutada: juhtkonna tasemel jäik ja konkreetsete tähtaegadega otsus, käskkiri, ähvardamine või nõ *näidis*poomine, suurima vastuseisja vallandamise näol.

Millal kasutada: kui soovitatav muutus organisatsioonis suure vastuseisu leiab ja eelnevad strateegiad ei tööta või kui muutus on vaja sisse viia koheselt, kuna iga venitatud hetk tähendab olulist majanduslikku kahju

Oodatav tulemus: muutused saavad sisse viidud

Tugevused: erakordselt tõhus viis igasuguse vastuseisu murdmiseks

Puudused: muutuste põhjustajad võivad sattuda organisatsioonis ebasoosituks ning võivad ootamatult avastada, et nende vastu pöörduvad ka teised organisatsiooni osad

4.4 Kliendi organisatsiooni ettevalmistamise ja juhtimise kokkuvõte

Eelnevalt selgub, et organisatsiooni ettevalmistamine tarkvaraprojekti eel ning selle pidev juhtimine tarkvaraprojekti kestuse ajal on möödapääsmatu, vastasel juhul on suur oht projekti ebaõnnestumiseks. Selline järeldus on sügavalt autori isiklik järeldus, mis on kinnistunud aina enam proportsionaalselt tarkvaraprojektide arvule, milles allakirjutanul on õnnestunud kaudselt või otseselt osaleda. Organisatsioonil on nimelt mitmeid hoobasid, mille abil soovitud muutuste rakendamist kas tahtlikult või kogemata takistada, seetõttu on terve organisatsiooni informeerimine, tööle rakendamine ja jälgimine arendusprojekti edus kriitilise tähtsusega.

Korralikult juhtimata organisatsioonis uue ERP juurutamise kogemusi ja tulemust saab lugeda ka Internetist, näitena lisan ZDNet'i artikli Marin'i maakonna vastavast projektist [13].

Tavaliselt ei ole olemas ühte konkreetset ja õiget strateegiat organisatsiooni juhtimiseks muutuste ajal, vaid tuleks rakendada erinevaid strateegiaid, erinevates muutuste protsessi etappides. Eeldatavasti aitavad sobivate valikute tegemisel eeltoodud strateegiate iseloomustus, kirjeldused ja analüüs. Käesoleva töö viimases osas, peatükis 5.8, leheküljel 59 on esitatud analüüs ka näidissettevõtte ERP juurutamise ajal kasutatud juhtimisstrateegiatest.

5. Arendusprotsess

Töös näitena toodud ettevõttes ei kasutatud algselt mingit metoodikat tarkvara tellimisel. Lähim vaste arendusmetoodikale oli „kaootiline“, mis piltlikult tähendas seda, et need vähesed arendused mis tehti olid väga pika ja piinarikka protsessi tulemus. Mahukaid arendusi ei tehtud kunagi, kõik mis tehti olid keelelises võrdluses nõ lihtlased. Iga arendust juhtis arendajaga otse see inimene, kes tellija organisatsioonis sellest arendusest kõige enam huvitus. Selline lähenemine põhjustas ülipikad arendustähtajad ja harva saadi lõpptulemuseks see, mida tegelikult sooviti.

Näiteks klientide autoriseeritud kontaktisikute funktsionaalsus ehk kliendi poolt volitatud inimeste nimekiri, kes antud kliendi nimel seadmeid rendile said võtta, oli lahendatud hiiglasliku tekstiväljaga kliendikaardil, kuhu oli võimalik vabas vormis kliendi kontaktisikuid lisada. Selline lähenemine tingis olukorra, kus kontaktisikuid oli nimekirjast keeruline leida (otsingufunktsiooni ei olnud), andmed olid seal täiesti ebastruktureeritud ja kirjas oli täpselt nii palju kui sinna viitsiti antud ajahetkel sisestada. Kuna süsteemil puudusid kasutajatasemed, siis said neid nimekirju muuta kõik kel süsteemis kasutajakonto. Loomulikult puudus igasugune automaatika nende kontaktide märkimiseks rendilepingule, seetõttu ei saanud rakendada süsteemset kontrolli, et rendileping tõesti antud ajahetkel volitatud isikule koostatakse. Lõpplahenduseks oli täiesti kasutu arendus, mis tööd mingil moel ei hõlbustanud.

Samas, kõiki eeltoodud ja ka muid ebapiisava funktsionaalsusega programmilõike kokku liites, formuleerus selge arusaam, mida ühelt efektiivselt ERP süsteemilt tegelikult nõudma peab. See teadmine tingiski projekti majandustarkvara vahetamiseks sellise vastu, mis olemasolevatele probleemidele lahendused pakuks.

Lühidalt arendusmetoodikatest üldiselt. Põhimõtteliselt eksisteerib kahte tüüpi arendusmetoodikaid.

Esimese tüübi alla kuuluvad nn kosk-tüüpi arendusmetoodikad, kus kõik tegevused, analüüs, disain, arendamine, testimine, juurutamine, hooldus on järjestikused ning tulemuseni jõutakse protsessi lõpus. Klient teeb ise tööd peamiselt analüüsi etapis ja näeb valmis tööd alles testimise faasis juba sellisel kujul nagu see oli analüüsifaasis kirjeldatud.

Teine tüüp arendusmetoodikaid on nõ. agiilsed (inglise keeles 'agile'), teise nimega paindmetoodikad. Painsidmetoodikate omapäraks on tarkvara arendamine lühikeste iteratiivsete tsükklitena, mis meenutavad spiraalprotsessi [14]. Sellist tüüpi metoodikate rakendamisel on arvestatud faktiga, et nõuded ei ole täielikud ega lõplikud vaid võivad iga hetk muutuda. Spiraalprotsess toetab sellist lähenemist. Spiraalprotsessiga probleemidele lähenemine on teatud perspektiivis kliendile mugav. On pigem reegel, et arenduse käigus tekib tellijal uusi ideid, mis alguses visioonis puudusid ja tingivad juba esitatud nõuete muutmise vajaduse. Ent samas toob selline lähenemine ka kliendisuunalise ebakindluse, kuna projekti on võimatu täpselt eelarvestada või täpselt projekti lõppu prognoosida - kuna töö maht pole projekti alguses lukus, vaid elab ja muutub projekti käigus. Positiivse külje pealt saab klient tarkvara oma kasutusse varem, tõsi küll jupikaupa, teoreetiliselt võimalusega seda ka jupikaupa rakendada.

Milline metoodika on arenduspartneril kasutusel, on tellija vaatevinklist enamasti ebaoluline. Tellija jaoks on oluline, et tarkvaraarendusega tegelev partner kasutab metoodikat, mida ta kõige paremini tunneb. Seega esitada omaltpoolt nõudmisi mingi konkreetse arendusmetoodika järgimiseks ei ole reeglina õigustatud.

Allakirjutanu hinnangul on kliendi jaoks kõige olulisemad punktid tarkvara väliselt partnerilt tellimisel hoopis alljärgnevad:

- Korralikult testitud, juurutatud ning aktsepteeritud tarkvaralahendus tähtaegselt ja eelarves.
- Tarnitud tarkvara juhend-dokumentatsioon, kogu funktsionaalsuse eesmärgipäraseks kasutamiseks. Dokumentatsioonist peaks eelistatult alati olema kaks versiooni, kõikehõlmav ja tellija protsesse järgiv.
- Arenduste analüüsi / disaini dokumentatsioon.
- Standardlahendusse integreeritud arendused ehk nn. „*custom coding*“ arendused standardlahenduse lähtekoodis peavad olema korrektselt dokumenteeritud.
- Olenemata arendusmetoodikast, tuleks tellijat kaasata arendusprotsessi nii palju kui võimalik – garanteerimaks õige asja tegemine ja välistamaks kahetimõistetavusi.

Kindlasti peaks arvestama võimalusega, et mingitel põhjustel tuleb teha tarkvaraprojekti käigus muutusi algselt esitatud nõuetesse, mis tõenäoliselt muudab nii tarneaega kui ka eelarvet, seega on oluliseks punktiks ka:

- Õigus nõuete muutmiseks projekti käigus, st kasutatav arendusmetoodika peab toetama nõuete muutmist.

5.1 Kas arendada või kasutada standardset funktsionaalsust

Töö iseloomust lähtuvalt käsitletakse peaausjalikult ERP toodete arendamist. ERP arendamisel on teatud iseärasused, nimelt pakitoodetes juba sisaldub teatav funktsionaalsus erinevate moodulite näol. Reeglina on pakitootes väga hästi realiseeritud finantsjuhtimine koos juurdekuuluvaga, aga ka laohaldus ja ost/müük, tihti ka tootmine ja ressursihaldus ning mõningane standardne aruandlus. Olenevalt tootest, on mooduleid võimalik omavahel kombineerida ning osta endale selline kompleksus tootest, mis kõige paremini ettevõtte äriprotsesse toetab. Võib osta ka kõik moodulid, ent see poleks otstarbekas, kuna tuleks maksta ka nende eest mida reaalselt ei kasutata. Lisaks ERP tootjale ja pakitootele eksisteerivad nõ vertikaalide tootjad oma vertikaallahendustega, mis täiendavad kas olemasolevat funktsionaalsust või lisavad täiesti uusi funktsioone, nt mingi spetsiifilise tootmisprotsessi täitmiseks või kasvõi realiseerides kassalahenduse.

Kas arendada või mitte? Kuidas leida tasakaalupunkt, kus arendada pole otstarbekas vaid pigem muuta oma äriprotsesse või vähendada nõudmisi nii, et saaks sama otstarvet täitva tööriista näiteks vertikaalmooduli näol tehtud minimaalselt väikese kuluga?

Spetsiifilise arenduse võimalused-plussid:

- Tellija saab rätseplahenduse, mis järgib väljakujunenud ning efektiivseid äriprotsesse ja toetab nende protsesside realiseerimist igakülgelt.
- Igakülgelt optimeeritud lahendus võimaldab unikaalseid võtmemõõdikuid (*KPI*) aruandluse tarbeks.
- Rätseplahendusel puuduvad litsentsitasud, mis õigustavad kallimat juurutust.

Spetsiifilise arenduse ohud-miinused:

- Eksisteerib oht, et ebaselgete või ebapiisavate nõuete tõttu arendatakse kasutuskõlbmatu komponent.
- Tellija organisatsiooni koostöövõime ja ebapädevus ülesandepüstituse ja nõuete kooskõlastamise faasis on suur risk arenduse ebaõnnestumiseks.
- Mõni erilahendus võib suurema tõenäosusega olla pakitootest ebaturvalisem, võib põhjustada andmete leket.
- Spetsiifiliste arendusega ERP süsteemi uuendamine uuele versioonile on oluliselt töömahukam kui standardtarkvara puhul. Korrektselt dokumenteerimata arenduste puhul võib versiooniuuendus osutada võimatuks ja otstarbekas võib olla uus tarkvara nullist paigaldada ja teha nullist uus juurutus – mis on väga töömahukas ja seetõttu pikk ning kallis projekt.
- IT-süsteemi arenduste oskusteave (*know-how*) on ainult ühe firma, võib-olla ühe inimese käes.

Vertikaallahenduse võimalused ja plussid:

- Vertikaallahendus on pakitoode, hangid sellise, mis toetab olemasolevaid protsesse maksimaalselt või on muul moel sobiv, nt hind, keel, lisaomadused.
- Vertikaallahenduse standardprotsessid võivad olla optimaalsemad kui ettevõttes seni kasutatud.
- Vertikaallahendus tuleb reeglina koos kasutuskõlbliku aruandepakiga.
- Vertikaallahendusele on reeglina lihtsam leida tootetuge, kuna seda kasutavad erinevad ettevõtted ja võib eksisteerida mitmeid spetsialiseerunud ettevõtteid seda vertikaallahendust toetama.
- Vertikaallahenduse juurutamine on eriprojektist kiirem, kuna puudub mahukas ja aeganõudev arendusfaas.

Vertikaallahenduse ohud ja miinused:

- Vertikaallahendused enamasti ei toeta kõiki unikaalseid äriprotsesse, protsesside mugandamine võib aga mõjutada ettevõtte kasumlikkust.

- Vertikaallahendus võib osutuda kallimaks algselt planeeritust:
 - o Vertikaallahenduse tootja võib nõuda tarkvara pidevat uuendamist, ettekäändel, et vanadele versioonidele ei pakuta tootetuge, iga versiooniuuendus on aga kulukas projekt.
 - o ERP kasutajaliidesega integreeritud vertikaallahendustel on reeglina kasutajapõhine litsentsihind, võib-olla kasutab vertikaali funktsionaalsust 25 inimest 100st ERP süsteemi kasutajast, litsentsitasu vertikaallahenduse eest peab maksma aga ikka kõigi süsteemi kasutajate eest.
- Ka vertikaallahenduste puhul eksisteerib oht, et firma, kes selle loonud on, lõpetab tegevuse. Sellises olukorras võib tarkvara kasutav klient jääda ilma kasutajatoeta. Halvemal juhul võib tekkida olukord, kus kasutaja on lahenduse „pantvang“, keegi sellest andmeid välja lugeda ei oska, kuna andmebaas on krüpteeritud ja läbi kasutajaliidese nt andme-eksporte teostada ei saa.

Ülaltoodud valikute vahel valimisel on selgelt tegemist kulude ja funktsionaalsuse (*cost vs functionality*) vahelise probleemiga ehk siin tuleks valikute selgitamiseks vajadusel kasutada otsustustööriista, näiteks tulude-kulude analüüsi või *SWOT* [15] analüüsi või, miks ka mitte, juba eespool kasutatud Thomas L. Saaty *AHP* [4] meetodit, et suhteliselt subjektiivsete kriteeriumite pealt langetada võimalikult objektiivne otsus. Kasutades näiteks Web-HIPRE [9] keskkonda, on sellise otsustusmudeli loomine küllaltki lihtne

Praktikas võib valik olla kombineeritud, valitakse nii vertikaallahendus kui ka tellitakse veel spetsiifilised arendused sinna juurde. Selliselt tegutses ka näiteks kasutatud ehitusseadmete rendiettevõtte. Selliselt käituses on loogiline, et päritakse mõlema variandi head küljed, samas ei maksa unustada, et niimoodi käituses päritakse ka mõlema lahenduse halvad küljed.

Antud osa meelepea oleks, et võimalusel tuleb vältida standardtoote põhifunktsionaalsuse muutmist, selle asemel tuleks lisada vertikaallahendusi, kasutada võimalikult palju olemasolevat koodi – rakendada võimalikult efektiivselt objektorienteeritud disaini.

5.2 Nõuete kogumine ja esitamine

Lõpuks ometi oleme jõudnud tarkvara tellimise sisulise osani, milleks on nõuete kogumine ja nõuete dokumendi koostamine. Tarkvara arendaja jaoks algab ju tegelikult protsess alles siin.

Mahukamate tarkvara tellimise projektide puhul on otstarbekas esitada nõuded eraldi dokumendina, ERP arenduste puhul on otstarbekas, kui selline dokument valmib kliendi ja arenduspartneri koostöös ning võiks kanda nimetust analüüsi dokument. Maksimaalselt efektiivne on töötubade meetodika, kus tellija esindaja ja tarnija konsultant saavad mängida läbi kõik nõutava arenduse kasutuskohad. Ainult nõnda toimides saab toote esindaja ehk arenduspartner kindla veendumusega öelda, kas nõutud funktsionaalsus on standardis olemas või mida ja kui palju tuleks juurde arendada.

ERP vallas väikeste arenduste korral saab nõuded enamasti esitada mitteformaalselt, kas näiteks kohtumise käigus koostatud dokumendina või siis otse kantuna arenduspartneri veebipõhisesse klienditoe süsteemi.

Nõuete peatüki selgitamiseks esitatakse reaalne näide. Näitena on toodud ühe keskmise suurusega arenduse nõuded (vt. Arendusvajaduste liike peatükis 2.1, leheküljel 15). Rendisüsteemi tellitakse arendus, mis võimaldab luua kaubakomplekte. Esitatakse järgnevad nõuded.

Funktsionaalsed nõuded:

- Kaubakomplekti peab saama luua iga erineva tellingutorni kohta, kusjuures iga tellingutorn sisaldab konkreetse hulga konkreetseid kaupu. nt
 - 4 ratast, kaubakood TT00020
 - 4 jalga, kaubakood TT00021
 - 6 raami, kaubakood TT00022
 - 2 horisontaali, kaubakood TT00023
 - 12 diagonaali, kaubakood TT00024
 - 2 tööplatvormi, kaubakood TT00025
 - 2 küljepiiret, kaubakood TT00026
 - 2 otsapiiret ja kaubakood TT00027
 - 4 teleskooptugijalga kaubakood TT00028

- Kaubakomplekti kaardil peab saama määrata komplekti hinna, nt 10.- €, mille süsteem jagab ise detailidele, proportsionaalselt vastavalt detailide müügihinnale ja kogusele.
- Nimetatud kaubakomplekti peab saama määrata lepingule samaselt, kui hetkel käib kauba või seadme valimine, selle vahega, et kaubakomplekti valimisel kannab süsteem automaatselt komplektiga seotud kaubad õiges koguses lepingule.
- Juhul kui rendilepingul komplekti kogust või hinda muudetakse, peab süsteem ümber arvutama ja uuendama ka rendilepingule kantud kaubaread.
- Kaubakomplekti arveldamisel, peab käibe konteerima detailidele ehk kaupadele, mitte komplektikoodile.
- Arvetele peab saama näidata valikuliselt kas ainult komplektikoodi koos nimega või siis komplektikood koos nimega + detailide nimekiri, kusjuures arveldatavate detailide hinnad on väljatrükil alati summeeritud komplektikoodi reale.

Mittefunktsionaalsed nõuded:

- Täielik dokumentatsioon uue arenduse efektiivseks kasutamiseks.
- Tellingukomplekti sisaldavate rendilepingute ridade sisestamisele kuluv aeg väheneb minimaalselt 75%.

Näites esitatud nõuete alusel teostatud arenduste tulemuseks on oluliselt mugavam ning töökindlam viis tellingukomplekte rentida, lisaboonusena oskab laotöötaja alati välja anda korrektse komplekti, klient saab turvalise toote, mida on võimalik kokku panna ja turvaliselt kasutada.

5.3 Leping

Leping on õige hetk püüda maksimeerida enda jaoks kummagi variandi, standardtoote vs rätseplahendus kasud ja võimalused ning minimeerida riskid ja ohud. Selleks tasub kasutada antud valikute näiteid ja analüüsi, mis toodud peatükis 5.1.

Lepingus tuleb selgelt sätestada, kellele kuulub programmi lähtekood, reeglina on see tarnija omand, ent tellija huvides peaks rakendama klauslit, mis võimaldaks tellijal lähtekoodi teatud tingimustel omandada, nt tarnija pankrotistumisel. Kuna ilma lähtekoodita on võimatu

olemasolevat modifitseerida või edasi arendada. Juhul kui kogu platvorm on mingil moel suletud, näiteks sisaldades tootja komponente, mille lähtekoodi ei avaldata, ei pruugi selline lähenemine tulemust anda. Siis on otstarbekas tõsiselt kaaluda teisi platvorme, millel taolist piirangut ei oleks.

Muu tavapärane tuleb loomulikult samuti lepinguga reguleerida, tähtajad, hinnad, poolte vastutus ja muu.

5.4 Arendamine

Peale nõuete esitamist, teeb arenduspartner tavaliselt pakkumise, ehk annab mõista, palju soovitud arenduse tegemiseks töötunde kulub, ehk, mis on planeeritava arendusse hind. Siin on veel võimalik tellimisest taganeda ning kõige õigem hetk vajadusel viimased täpsustused nõuete osas teha.

Olenevalt tellitavast arenduse mahust, sõltub edasine tellija roll projektis:

Mikroarenduse puhul üldiselt tellija esitab tellimuse ja seejärel lihtsalt ootab, kuni tellitud töö ette võetakse ja ära tehakse.

Miniarenduse puhul on seis üldjoontes sama, tellijal pole teha muud kui tellimus sisse anda ja oodata, välja arvatud juhul, kui arendajal tekib täpsustavaid küsimusi, siis konsultant tavaliselt selgitab vastused ebaselgustele ja peale arenduse lõppu antakse tellijale üle valmis toode.

Miniarenduse puhul, kuna se on keerulisem kui mikroarendus, samas tegelikku arenduse tellijat (nt raamatupidaja, müügimees) otseselt arendusprotsessi ei kaasata, siis eksisteerib risk, et ülesande püstituse käigus lähenetakse teemale liialt pealiskaudselt ning arendus tellitakse valede või puudulike nõuete järgi, lõpptulemusena on oht, et valmib kasutu toode. Seetõttu peaks just selle arendusliigi puhul pöörama erilist tähelepanu nõuete defineerimisele.

Midiarenduse puhul on otstarbekas nõuda korrapärast tagasisidet tehtud tööle, nt staatuse aruannete näol, kuna reeglina ühe hooga töö valmis ei saa, vaid valmib mitmes etapis. Tagasiside ja kommentaarid käivad enamasti veebipõhises arendaja portaalis.

Midiarendus on esimene arenduse liik, mille korral tuleb pea alati panustada ka organisatsiooni juhtimisele. Kuna tehtav arendus puudutab reeglina mitmeid kasutajaid, võib

olla õigustatud suhtlusstrateegia, vt peatükk: 4.2.1, leheküljel 39, kasutamine „projekti“ algfaasis ning vajadusel ka muud organisatsioonijuhtimise strateegiad.

Kesksuurele arendusele tuleks alati läheneda kui projektile, koos projektiplaani, projektijuhi, projektitiimi ja tellijaga esindajaga või kogu tellija tiimiga. Projektijuht esitab korrapäraselt vahearuandeid, nii dokumendina kui ka ettekannetena telefonikonverentsi või füüsilise kohtumise raames. Koosolekute käigus otsustatakse tekkinud probleemide lahendused, võimalusel leitakse väikestele küsimärkidele kiiremad vastused ka veebisuhtluses arendaja portaalis. Sellistel arendustel on tellijal oluliselt suurem roll, peamiselt siiski analüüsi ja testimise faasis. Otstarbekas on kaasata organisatsiooni juhtimiseks projekti alguses suhtlusstrateegia ning hilisemas faasis kaasamisstrateegia, vt peatükid: 4.2.1, leheküljel 39 ja 4.2.2, leheküljel 40. Projekti lõpuks on reeglina tarvis organiseerida kasutajatele koolitused.

Maksiarendust või uue tarkvara juurutamist tuleb kindlasti käsitleda kui projekti, koos projektijuhi, projekti tiimi, projekti plaani ja tellija tiimiga. Tavaliselt on arendajatel mingi konkreetne meetodika millega tarkvaralahendusi ellu viiakse, Microsofti äritarkvara juurutusmeetodikaks on tavaliselt Microsoft Dynamics Sure Step [16] meetodika, seda rakendatakse nii juurutamistel kui ka versiooniuuendustel. Sure Stepi meetodikat kirjeldav protsessi slaid on näha töö lisas: Lisa 4. Microsoft Dynamics Sure Step ülevaade, leheküljel 70.

Maksiarenduse korral on organisatsiooni juhtimine hädavajalik, kuna organisatsioonil on võtmeroll projekti õnnestumiseks. Kindlasti tuleb korralikult panustada suhtlusstrateegiale, samuti kaasata võimalikult suur osa organisatsioonist projekti ellu viima ning vajadusel tuleb läheneda inimestele individuaalselt, kasutades nt juhendamisstrateegiat või läbirääkimiste strateegiat. Viimases hädas on maksarenduse korral õigustatud organisatsiooni ettevalmistamiseks ka sundimisstrateegiate kasutamine. Vaata organisatsiooni juhtimisstrateegiad, peatükk 4.2, leheküljel 38.

Ülaltoodud erinevad lähenemised ja meetodikad erinevate arendusmahtude puhul on näidissettevõtte peal töötanud hästi. Miski ei garanteeri täielikku edu, alati jääb arendusse mingi kogus teadmata faktoreid, mis tarkvaraprojekti kulgemist negatiivselt võivad mõjutada. Seetõttu on ka ülaltoodud meetodikaid kasutades aeg-ajalt ikka ette tulnud arenduste venimisi, kuid täielikke ebaõnnestumisi on õnnestunud vältida.

5.5 Testimine ja aktsept

Arendaja poolt tarnitud tarkvara tuleb alati testida. Testimine on vajalik veendumaks tarnitud tarkvara kvaliteedis. Selleks on tavapärane, et arendust tellival organisatsioonil või siis arendaja organisatsioonil on oma arendus-testkeskkond, kuhu arendus algselt luuakse. Antud testkeskkonnas tuleks läbi viia ka ennetav testimine, enne arenduse tööbaasi tõstmist. ERP arenduste puhul on Eestis tavapäraseks saanud, et testimise vastutus delegeeritakse arendaja poolt tellijale. Arendaja küll testib lahendust, aga mitte rohkem kui hädapärastel tarvilik, piltlikult öeldes vaatab arendaja, kas uus lisatud nupp midagi teeb või aruande puhul, et kas mingid andmed veergudes jooksevad. Sisuline testimine jääb ikka tellija mureks. Pole harvad olukorrad, kus arendaja poolt testitud lahendus vajab veel päeva-paar tööd, kuna palju muutujaid olemasolevas süsteemis on arendajal jäänud kahe silma vahele.

Põhimõttelisi testimise liike ERP arenduste alal, mida tellija ise saab korraldada, on kaks, esimene neist, konkreetse arenduse testimine, ehk kontrollimine, kas tehtud arendus vastab tellitule ehk, kas esitatud nõuded on tarkvaras funktsionaalsuse ja omadustena realiseeritud.

Teine testimise liik on regressioonitestid ehk kogu kasutatava funktsionaalsuse läbitestimine. Õige oleks iga uue arenduse puhul teha kogu funktsionaalsuse regressioonitestid. Kahjuks meie näites kasutatud ERP regressioonipraktika rakendamist reaalselt siiski ei võimalda, kuna teste peaks tegema käsitsi inimene, vastavat automaatikat süsteemi sisse ehitatud pole. Miks võimatu? Peasjalikult arenduste koguhulga tõttu, reaalselt tehakse väikeseid muudatusi iganädalaselt, peale iga 'mikro-muudatuse' tegemist käsitsi kõiki protsesse läbi testida poleks majanduslikult mõttekas ja ajaliselt võimalik, piltlikult öeldes peaks olema tööl inimene kes kogu-aeg testiks. Ka selline protsess on kaheldava väärtusega, kuna uut arendust ei tohiks luua enne, kui kõik testid eelmise arendusega lõpetatud. Samas tasub rakendada tervet mõistust ning saada aru, et aruande arendamine, mis ainult pärib andmeid, ei vaja tagantjärgi kõikide funktsioonide läbitestimist. Mahukam test aga on suure tõenäosusega õigustatud, kui muudetakse mingit süsteemset funktsionaalsust, nt seoses arveldamisega.

Eelöeldu põhjal, iga arendus vajab testimist, samas testide maht sõltub arendusest, pimesi aktsepte jagada ei tohi.

5.6 Käivitamine ehk *Go-Live*

Käivitamine eh *go-live* on tarkvaraarenduse nõ finaal. Üldiselt käivitamine on natuke pidulik termin, mida kasutatakse nt ERP juurutamisel või versiooniuuendusel ja tähistab seda hetke, kui ettevõtte uue tarkvara ametlikult käiku võtab. Nõ igapäevaste väikeste ERP arenduste tööbaasi tõstmiseks käivitamise terminit üldiselt ei kasutata, siin kasutatakse pigem väljendit: 'töö lõpetamine'. Käivitamine pole ainult nimetaja esimesele kasutuskorrale, vaid on protsess, mis tegelikult on kogu eelneva tarkvaraprojekti jooksul hoolikalt ette valmistatud. Käivitamine on tarkvaraprojekti viimane tähis ehk, kui on näha, et projekt venib, siis võib ka käivitamise kuupäeva rahulikult edasi lükata. Tavapäraselt tehakse umbes, olenevalt projekti liigist, nädal enne planeeritud käivitamise päeva koosolek, kus langetatakse olemasoleva info põhjal otsus, kas käivitamine tegelikult toimub või tuleb edasi lükata. Palju käivitamise otsuseid tehti Eestis ilmselt 2011 aasta alguses, kus toimus Eesti riigi raha vahetus Eesti kroonilt euro vastu. Paljud infosüsteemid tehti eelnevalt ümber ja aasta lõpus ning uue alguses tehti suure tõenäosusega sadu, kui mitte tuhandeid andmete konverteerimisi, eks iga ettevõtte pidi siis konverteerimise järgselt langetama käivitamise või tagasi kerimise (*rollback*) otsuse.

Igale vähegi mahukamale uue funktsionaalsuse käivitamisele peaks eelnema koolitus, tulles töö aluseks oleva näidisettevõtte juurde ja tema ERP juurutamise juurde, siis üks saadud kogemustest oli, et kasutajate lõppkoolitus tuleks alati läbi viia tegelike andmetega. Kui kasutajad on harjunud, et nende osakonna nimi on 'Tallinn, Mustamäe', mitte 'London BD1', siis olgu see nii ka koolituskeskkonnas. Sama pädeb ka muude parameetrite kohta, näiteks seadmete rendiga tegeleva ettevõtte tootebaasis ajavad tooted nagu 'jalgratas' või 'lego komplekt' kasutajad koolituse ajal segadusse, samuti hajub tähelepanu. Õige on kasutada igapäevaseid termineid, nagu näiteks 'maatihendaja' ja 'akutrell'.

5.7 Tarkvaraprojekti järeltegevused

Igale juurutusele, aga ka versiooniuuendusele ja muidu mahukamale lisafunktsionaalsuse arendusele peaksid järgnema mõned järeltegevused. Allpool on välja toodud neist kõige olulisemad, mis mõnikord kipuvad tähelepanuta jääma.

5.7.1 Hooldusleping:

Mahukama ning vähegi olulisema arenduse puhul, eriti aga ERP juurutamise korral, on oluline sõlmida arenduspartneriga hooldusleping. Hoolduslepingul on üldiselt kolm olulist eesmärki:

- Pakkuda igapäevast kasutajatuge olemasolevale lahendusele
- Arendada olemasolevat lahendust edasi
- Kõrvaldada tehtud arenduste tarkvaravead ehk *bug'id*

Hoolduslepingu osas tasub tarkvara tarnijaga samuti juba algse lepingu ajal kokku leppida. Otstarbekas on esialgu leppida kokku fikseeritud igakuises tasus, mida kord aastas või vajadusel tihemini üle vaadatakse, et kas rahuldab mõlemat osapoolt. Fikseeritud tasu pluss arendajale on, et neil on garanteeritud tulu, tellijal aga kindel teadmine, kui suur igakuine kulu saab olla, seega lihtne eelarvestada. Loogiliselt kuuluvad arenduskulud eraldi tasustamisele ja neid ilmselt fikseeritud tasu sisse ei õnnestu määrata. Lepingusse võib aga määrata kompromissi, et kui nt 3 kuud järjest on kasutajatoe tunde kulunud nt kolmandik fikseeritud tasu põhisummast, siis ülejäänuga kaetakse arenduste kulud.

Üldiselt tasub ERP maailmas arendajale, kes pakub ka kasutajatuge, delegeerida igasugused süsteemi ülalpidamisega kaasnevad rutiinsed tegevused, kuna nemad teavad kuidas ja oskavad neid kõige paremini rakendada. Siin peetakse silmas nt andmebaaside hooldust ja indekseerimist, tabelite optimeerimist ja muid perioodilisi tegevusi, mida saab automatiseerida, kuid milleks tellijal endal reeglina piisavalt teadmisi pole.

Vastutuse jagamine on lepingutes tihti ebapiisavalt kirjeldatud, selle all peetakse eelkõige silmas tarkvaras ilmnenu vigade parandamist. Tasuks kokku leppida ilmnenu vigade liigid ja parandamise pakilisus, näiteks kategooriad A, B, C ning määrata neile parandamise tähtajad ja rakendada sanktsioone vigade paranduse ületähtaegseks venimise vältimiseks.

Täpsemaid juhiseid hoolduslepingu SLA sisu ja ka muude osade koostamise kohta saab näiteks infotehnoloogia haldamise standardist ITIL (*Information Technology Infrastructure Library*) [17].

5.7.2 Koolitused

Kuigi käivitamisele eelnesid kasutajakoolitused, ei saa unustada, et koolitusi on vaja ka tulevikus. Näiteks kipub kasutajail ununema harvemini kasutatav funktsionaalsus, mis vajab aeg-ajalt järelkoolitust. Näitena toodud ettevõttes on järelkoolitused lülitatud reeglina aastakoosolekute raamesse, kitsaskohad käiakse läbi ja tuletatakse meelde, on nõ küsimustevastuste voor. Kasutajate poolt on tagasiside väga positiivne. Samuti esineb süsteemis vähem vigu, mis tähendab väiksemat koormust kasutajatoele.

Koolitust vajavad ka uued töötajad. Antud ettevõttes lahendatakse uute töötajate koolitus majasiseselt, piltlikult öeldes toimub väljaõpe kohapeal, oma osakonna sees. Näiteks uus klienditeenindaja leiab end kohe, esimesel tööpäeval arvuti tagant, kogenud töötaja kõrvalt ja päeva lõpuks on ta juba ise mõnedki rendilepingud ja tagastused teinud. Meie kogemuste põhjal toimub selline lähenemine efektiivselt.

5.7.3 Vana süsteemi väljalülitamine / arhiveerimine

Kui uus süsteem on lõpuni juurutatud, kasutajad koolitatud ja käivitamine toimunud, on otstarbekas midagi teha ka vana süsteemiga, põhimõttelisi variante on kolm:

- Kustutada (kõige turvalisem, kuna midagi ei saa lekkida; kõige ohtlikum, kuna kui avastatakse puudujääk mingites andmetes, on raske minevikku kontrollida ja olukorda parandada)
- Välja lülitada ja arhiveerida (mõõdukalt kulukas ja kindlapeale minek, vajadusel on süsteem uuesti käivitav)
- Mitte midagi teha, jätta nii nagu on – kõige kulukam, piltlikult öeldes on ettevõtte kanda nüüd paralleelselt kahe süsteemi kulud; ebaturvaline, kuna töös olevat süsteemi on lihtne rünnata; riskantne, kuna keegi võib ajaloolistes andmetes pahatahtlikke tegevusi teha

Kindlasti pole ka halb idee salvestada (näiteks PDF [18] kujul) ning digitaalselt arhiveerida dokumente, millel on seadusejärgne säilitamiskohustus, nt raamatupidamisdokumendid, säilitamiskohustusega 7 aastat [19]. Selliselt käitudes pole vaja arhiveeritud süsteemi uuesti käivitada, juhul kui mõni klient küsib müügiarveid, mis moodustatud vanas süsteemis.

5.8 Juhtimisstrateegiad arendusprotsessis

Peatüki lõpetuseks analüüsin kasutatud juhtimisstrateegiaid meie põhinäite ehk ERP juurutusprojekti varal.

Kui näidissettevõttes oli langetatud otsus majandustarkvara vahetuseks, põhjustas see uudis organisatsioonis parasjagu segadust. Peamine segaduse põhjustaja oli hirm, et uus „rendiprogramm“ on keeruline ja pole kasutajasõbralik. Osaliselt usuti, et uus majandustarkvara juurutatakse ainult töötajate parema kontrollimise eesmärgil. Kuigi tegelikult just olemasolev rendiprogramm oli see, mille kasutajasõbralikkus soovida jättis – kuidagi kanti olemasoleva programmi probleemid automaatselt üle ka uude programmi, teadustamata, et just olemasolevate probleemide lahendamine ongi majandustarkvara vahetamise põhjuseks.

Meie näites aitas vastuseisu murda peamiselt tihedam ja avatud infojagamine, töötajate kaasamine ja hilisemas etapis ka juhendamine.

Suhtlusstrateegiat kasutati:

- kasutajatele tehti uue programmi tutvustusi
- näost-näku koosolekutel (nii müügi kui ka juhtkonna tasemel) oli majandustarkvara vahetuse hetkeseis tihti koosoleku osa
- Suhtluseks kasutati palju e-posti teel teavitusi

Kaasamist korraldati mitmeti:

- võtmekasutajad kaasati uue funktsionaalsuse arendusprotsessi
- kasutajaid kaasati andmekorrastamise protsessi, koostöös korrigeeriti olemasolevad andmed
- tootetiimi kaasamisel loodi täiesti uus toodete hierarhia, mis rahuldaks emattevõtte aruandlusnõudlust
- loodi uus töökoht, krediidikontroller, kelle hallata jäid kliendid ja kontaktisikud ning kes lõi esialge versiooni kliendibaasist koos korrektselt struktureeritud volitatud isikute andmebaasiga

- finantsosakonna initsiatiivil loodi kulude jagamise funktsionaalsus, mis kulud automaatselt osakondade vahel jagas ning müügiarvete kliendile kättetoimetamine usaldati allhankijale
- kasutajad kaasati testimisprotsessi, sellega sai testitud süsteem ja ka kasutajad said uue tarkvaralahendusega tuttavamaks

Juhendamisstrateegiat kasutati järgmiselt:

- testkeskkond tehti kasutajale avalikuks võimalikult varakult, juba mitu kuud enne käivitamist
- koostati detailsed juhendid, mis järgisid konkreetseid protsesse, nt kuidas teha rendilepingut / sularahaarvet / osalist tagastust jms
- vahetult enne käivitamist tehti kogu kasutajaskonnale nende igapäevategevuste põhised koolitused
- käivitamise ajal organiseeriti füüsiline kohapealne kasutajatugi suurematesse klienditeeninduspunktidesse
- peale käivitamist külastas peakasutaja kõiki osakondi üle Eesti ja lahendas kohapeal tekkinud küsimused

Lõppkokkuvõttes, käivitamine õnnestus hästi, tänu organisatsiooni juhtimisele toetati muutuste ja tarkvara juurutamise protsessi igakülgsest ning muutused võeti vastu juba üdini positiivselt.

6. Kokkuvõte

Antud töö põhieesmärkideks oli pakkuda välja lahendused, kuidas selgitada välja tarkvara arendusvajadus, leida arendaja, kes sobiks soovitud arendusi ellu viima ning juhendada, kuidas soovitud tarkvaraarendus edukalt lõpule viia ning tulemuslikult tööle panna.

Töö olulisemate tulemustena võiks ära märkida:

- Arendusvajaduse väljaselgitamise meetoodika, kus mõne lihtsa parameetri teadmisel on võimalik otsustada arendusvajaduse põhjendatuse üle ning langetada otsus, kas arendada või mitte. Ühtlasi saades meetoodikatest juhtnööre, kuidas arendusele läheneda.
- Arenduspartneri valiku meetoodika, mis aitab kergesti kättesaadavate ettevõtte näitajate baasil langetada otsuse võimalike kandidaatide vahel. Esitatud mõõdikud/-näitajad ei ole kahtlemata ainuvõimalikud, praktiliselt võib kasutada mistahes kättesaadavaid näitajaid.
- Organisatsiooni juhtimise strateegiad ja taktikad, kasutades ära Schlesingeri ja Kotteri tööde [12] põhjal tehtud järeldusi, on antud töös analüüsitud võimalikke strateegiad, kuidas organisatsiooni tarkvara arenduse- ja juurutamisega tekkivate muutuste keerises tulemuslikult juhtida. Seejuures eriliselt rõhudes vajalikkusele antud olukorras organisatsiooni juhtida.
- Arendusprotsessi peatükis on selgitatud võimalikult lihtsalt ERP arendamise protsessi ja iseärasusi kliendi ehk tellija perspektiivist. Olulisemaks tulemuseks võiks mainida analüüsi, mis kaardistab standardtoote ja „rätsepalahenduse“ erinevused ning plussid-miinused. Lõpus antakse ülevaade milliseid suhtlusstrateegiaid kasutati näidisettevõtte ERP juurutusprojekti käigus.

Tööst tulenevalt võib väita ning järeldada, et allhanke korras tarkvaraarendus, tarkvaraarenduse kompetentsita ettevõttes ei ole kindlasti võimatu. Järjekindluse ja pideva eneseharimise abil on võimalik välja arendada täiesti töötavad meetodid tarkvara arenduste tellimiseks ning käivitamiseks.

Tulemustest ja järeldustest tulenevalt on allakirjutanul julgust väita, et käesolevale tööle osundatud eesmärk saavutati. Välja on pakutud täiesti toimiv ja rakendatav kogum tegevusi, kuidas tarkvaraarendust, vastava kompetentsita ettevõttes, välise pakkuja kaudu teostada. Põhimõtteliselt võiks seda tegevuste kogumit isegi nimetada tarkvara arendusmetoodikaks, mis ei nõua tellija organisatsioonilt liigselt eriteadmisi ning pühendumist.

Loomulikult on võimalik antud tööd ka edasi arendada, üks võimalik edasiarenduse suund oleks kirjutatud kõrvutada teistsuguste allhanke korras lähenemistega tarkvaraarendusele. On ka mitmeid muid võimalusi, nt võrdlus organisatsiooniga, kus on loodud oma majasisene arendusosakond või vastupidi, võrdlus organisatsiooniga, kus on veelgi enam panustatud kolmanda osapoole esindajate kaasamisele. Kus näiteks lisaks arendamisele oleks allhanke korras korraldatud ka majasisesed tarkvaraarenduse protsessi tegevused, konsultantide kaasamise näol.

Käesolev töö ei pretendeeri ainuõigeks näiteks, kuidas allhanke korras tarkvaraarendust korraldada. Käesolev töö püüab pigem näitlikustada, kuidas on võimalik vähete vahenditega, vähese bürokraatia ning mõistliku pühendumisega toota optimaalne tulemus, mis on kasutatav, töökindel, edasiarendatav ja samas efektiivne ning kasutajasõbralik. Ainsaks eelduseks on, et nende väärtuste nimel tehakse tööd, metoodika ei garanteeri tulemust – see on allakirjutanu kogemusest tulenev tõekspidamine.

Summary

The main purpose of the thesis was to come up with methods to identify the need for development, to find a developer for coding the required functionality and finally to give ideas, how to successfully implement the developed functionality.

The main results obtained were:

- Methodology to identify the needs for development. All it takes, is to identify couple of details and then it would be possible to decide on the feasibility of the desired change. The developed methodologies also guide how to approach a development.
- Methodology for choosing a development partner. The methodology uses easily available information to come up with feasible decision between the given alternatives. The used metrics and indications are certainly not the only ones to be used on such cases, it is justified to use all and any available metrics.
- Communication strategies and -tactics for managing organization during change process were inspired by Schlesinger's and Kottler's work [12]. Analysis of possibilities to steer an organization during the change process often associated with software implementation projects was carried out in the chapter. The importance of steering the organization during such projects was specifically stressed on.
- Development process chapter has explanations and analysis of ERP development specific features from customer perspective. Chapter also has analysis of common off the shelf software packages versus tailor-made software, listing strengths and weaknesses of each, and giving some tips and recommendations for decision making. An overview of organization management communication strategies is also given.

It can be concluded from the analysis and results, that it is feasible to run outsourced software development projects even in organizations that lack specific experience in the field. Consistent commitment and openness to self-education are keys to develop sustainable methods for software development and implementation.

The author is bold enough to suggest that the goals assigned to the work were achieved. A complete set of applicable methodologies to run outsourced software development have been developed, and analyzed. In principle, the set of methodologies introduced could be labelled as development methodologies that require little dedication and know-how to be implemented successfully in any organization.

It would be possible to further elaborate on the issues analyzed in this thesis. For example to bring in another perspective, for example compare to other approaches of outsourced software development. Or compare with different kind of organizations, with ones that have also outsourced internal processes to consultants, or an opposite example, where development is also done in-house.

This thesis does not pretend to be the sole and only way to run outsourced software projects. The main aim of the thesis instead, is to introduce and prove that a light approach, with little bureaucracy and reasonable dedication could be successful in achieving optimum results, that are usable, dependable yet effective and user-friendly. The most important prerequisite would be the willingness to commit. In the end, it's not the methodology that delivers – it's the people that do, this is the opinion and experience of the author.

Kasutatud kirjandus

- [1] „Wikipedia: ERP,“ [Võrgumaterjal]. Available: http://en.wikipedia.org/wiki/Enterprise_resource_planning. [Kasutatud 25 4 2014].
- [2] „Microsoft Dynamics NAV,“ Microsoft, [Võrgumaterjal]. Available: <http://www.microsoft.com/en-us/dynamics/erp-nav-overview.aspx>. [Kasutatud 26 4 2014].
- [3] M. Krigsman, „ZDNet: California abandons \$2 billion court management system,“ 2 4 2012. [Võrgumaterjal]. Available: <http://www.zdnet.com/blog/projectfailures/california-abandons-2-billion-court-management-system/15363>. [Kasutatud 30 4 2014].
- [4] „Wikipedia: AHP,“ [Võrgumaterjal]. Available: http://en.wikipedia.org/wiki/Analytic_hierarchy_process. [Kasutatud 27 4 2014].
- [5] „Wikipedia: Business case,“ [Võrgumaterjal]. Available: http://en.wikipedia.org/wiki/Business_case. [Kasutatud 30 4 2014].
- [6] „BPMN,“ Object Management Group, Inc, [Võrgumaterjal]. Available: <http://www.bpmn.org/>. [Kasutatud 30 4 2014].
- [7] „Wikipedia: RFP,“ [Võrgumaterjal]. Available: <http://en.wikipedia.org/wiki/RFP>. [Kasutatud 25 4 2014].
- [8] „Wikipedia: Issue tracking system,“ [Võrgumaterjal]. Available: http://en.wikipedia.org/wiki/Issue_tracking_system. [Kasutatud 27 4 2014].
- [9] R. P. Hämäläinen ja J. Mustajoki, „WEB-Hipre,“ [Võrgumaterjal]. Available: <http://hipre.aalto.fi/>. [Kasutatud 27 4 2014].
- [10] „Wikipedia: SLA,“ [Võrgumaterjal]. Available: http://en.wikipedia.org/wiki/Service-level_agreement. [Kasutatud 25 4 2014].
- [11] A. Virovere, R. Alas ja J. Liigand, Organisatsiooni käitumine, Külim, 2008.
- [12] L. A. Schlesinger ja J. P. Kotter, „Choosing Strategies for Change,“ *Harvard Business Review* 57, no. 2, March-April 1979.
- [13] M. Krigsman, „ZDNet: Understanding Marin County's \$30 million ERP failure,“ 2 9 2010. [Võrgumaterjal]. Available: <http://www.zdnet.com/blog/projectfailures/understanding-marin-countys-30-million-erp-failure/10678>. [Kasutatud 12 5 2014].
- [14] J. Tepandi, „Tarkvara kvaliteet ja standardid (IDX5721, IDX5722),“ [Võrgumaterjal]. Available: <http://deephought.ttu.ee/users/tepani/we-tk.html>. [Kasutatud 23 4 2014].
- [15] „Wikipedia: SWOT,“ [Võrgumaterjal]. Available: http://en.wikipedia.org/wiki/SWOT_analysis. [Kasutatud 13 5 2014].
- [16] Microsoft, „Microsoft Sure Step,“ [Võrgumaterjal]. Available: <https://www.microsoft.com/dynamics/support/implementation/success.aspx>. [Kasutatud 1 5 2014].
- [17] J. v. Bon, ITIL v3 - A Pocket Guide, Van Haren Publishing, 2009.
- [18] „Wikipedia: PDF,“ [Võrgumaterjal]. Available: http://en.wikipedia.org/wiki/Portable_Document_Format. [Kasutatud 13 5 2014].

[19] „Raamatupidamise seadus,“ 23.3.2014. [Võrgumaterjal]. Available:
<https://www.riigiteataja.ee/akt/125052012016?leiaKehtiv>. [Kasutatud 25.4.2014].

Lisa 1. Meetodid muutuste juhtimiseks organisatsioonis

Meetodid muutuste juhtimiseks organisatsioonis [12]

Method	How to Use	When to Use	Advantages	Drawbacks
Education	Communicate the desired changes and reasons for them	Employees lack information about the change's implications	Once persuaded, people often help implement the change	Time consuming if lots of people are involved
Participation	Involve potential resisters in designing and implementing the change	Change initiators lack sufficient information to design the change	People feel more committed to making the change happen	Time consuming, and employees may design inappropriate change
Facilitation	Provide skills training and emotional support	People are resisting because they fear they can't make the needed adjustments	No other approach works as well with adjustment problems	Can be time consuming and expensive; can still fail
Negotiation	Offer incentives for making the change	People will lose out in the change and have considerable power to resist	It's a relatively easy way to defuse major resistance	Can be expensive and open managers to the possibility of blackmail
Coercion	Threaten loss of jobs or promotion opportunities; fire or transfer those who can't or won't change	Speed is essential and change initiators possess considerable power	It works quickly and can overcome any kind of resistance	Can spark intense resentment toward change initiators

Lisa 2. Hierarhilise analüüsimeetodi kriteeriumite kaalud

Kogemuste ja kvaliteedi alamkriteeriumite kaalude leidmiseks kasutatud arvutustabel:

	Vanus	Konsultandid	Arendajad	Projuhid	Projektid	Suurim	Tunnihinne	Geom kesk.	Kaalud
Vanus	1,00	0,33	0,33	1,00	0,33	0,33	3,00	0,62	0,084
Konsultandid	3,00	1,00	1,00	1,00	1,00	1,00	0,33	1,00	0,135
Arendajad	3,00	1,00	1,00	3,00	3,00	1,00	1,00	1,60	0,216
Projuhid	1,00	1,00	0,33	1,00	1,00	0,33	1,00	0,73	0,099
Projektid	3,00	1,00	0,33	1,00	1,00	1,00	1,00	1,00	0,135
Suurim	3,00	1,00	1,00	3,00	1,00	1,00	3,00	1,60	0,216
Tunnihind	0,33	3,00	1,00	1,00	1,00	0,33	1,00	0,85	0,115
Summa:	14,33	8,33	5,00	11,00	8,33	5,00	10,33	7,41	1,00

Kommunikatsiooni alamkriteeriumite kaalude leidmiseks kasutatud arvutustabel:

	Suhtluskanal	Suhtluskeel	Asukoht	Geom kesk.	Kaalud
Suhtluskanal	1,00	0,33	1,00	0,69	0,200
Suhtluskeel	3,00	1,00	3,00	2,08	0,600
Asukoht	1,00	0,33	1,00	0,69	0,200
Summa:	5,00	1,67	5,00	3,47	1,00

Lisa 3. Hierarhilise analüüsimeetodi grupiotsuse kaalud

Tegevjuht	Partner			Geom kesk.	Kaalud
	A	Partner B	Partner C		
Partner A	1,00	0,33	0,20	0,41	0,105
Partner B	3,00	1,00	0,33	1,00	0,258
Partner C	5,00	3,00	1,00	2,47	0,637
Summa:	9,00	4,33	1,53	3,87	1,00

Finantsjuht	Partner			Geom kesk.	Kaalud
	A	Partner B	Partner C		
Partner A	1,00	0,20	0,20	0,34	0,091
Partner B	5,00	1,00	1,00	1,71	0,455
Partner C	5,00	1,00	1,00	1,71	0,455
Summa:	11,00	2,20	2,20	3,76	1,00

IT juht	Partner			Geom kesk.	Kaalud
	A	Partner B	Partner C		
Partner A	1,00	1,00	0,20	0,58	0,156
Partner B	1,00	1,00	0,33	0,69	0,185
Partner C	5,00	3,00	1,00	2,47	0,659
Summa:	7,00	5,00	1,53	3,74	1,00

Grupi kaal	Normeeritud grupikaalud
0,114	0,118
0,279	0,288
0,576	0,594
0,969	

Lisa 4. Microsoft Dynamics Sure Step ülevaade

