

TALLINNA ÜLIKOOL  
Informaatika Instituut

# AGIILSE METOODIKA RAKENDAMINE PROJEKTI MY.SPACEMAN NÄITEL

Magistritöö

Autor: Janno Ketolainen

Juhendaja: Peeter Normak

Autor: ..... «.....».....2011. a.

Juhendaja: ..... «.....».....2011. a.

Instituudi direktor: ..... «.....».....2011. a.

Tallinn 2011

## **Autoridelkaratsioon**

Deklareerin, et käesolev magistritöö on minu töö tulemus ja seda ei ole ei minu ega kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....  
(kuupäev)

.....  
(magistritöö kaitsja allkiri)

# Sisukord

Sisukord .....	3
Mõisted.....	5
1. Sissejuhatus .....	6
1.1. Aktuaalsus.....	7
1.2. Eesmärk .....	8
1.3. Probleemi kirjeldus.....	8
1.4. Magistritöös kasutatav meetodika.....	11
1.5. Magistritöö struktuur .....	11
2. Agiilne meetodika.....	12
2.1 Projektijuhtimise meetodikate ajalugu .....	12
2.2 Koskmudelil põhinevad arendusprojektid.....	12
2.3 Koskmudelil põhinevate arendusprojektide probleemid.....	13
2.4 Agile Manifesto.....	14
2.5 Agiilse ja traditsioonilise meetodika võrdlus.....	15
2.6 Levinumad agiilsed meetodikad.....	16
2.6.1 Ekstreemprogrammeerimine .....	17
2.6.2 Scrum .....	18
2.7 Lepingute sõlmimine ning projektide eelarvestamine agiilses tarkvaraarenduses .....	19
2.7.1 Fikseeritud aeg, maksumus ja skoop.....	20
2.7.2 Fikseeritud kogumaksumus ja aeg .....	20
2.7.3 Aeg ja raha .....	20
2.7.4 Osade kaupa tellimine raamlepingu alt.....	21
2.8 Äripoole osalemine arendusprojektides .....	21
2.9 Iseorganiseeruv meeskond.....	22
2.10 Agiilsete meetodikate juurutamine organisatsioonis .....	23
3. Agiilse meetodika rakendamine ETK-s .....	24
3.1 Projekti My.spaceman tutvustus.....	24
3.2 Projektiplaani koostamine.....	25
3.2.1 Esialgne plaan .....	25
3.2.2 Üleminek Agiilsele projektiplaanile .....	26
3.3 Projekti esimene etapp: Connectivity andmebaasi loomine.....	27
3.3.1 Vajaduste kaardistamine Connectivity jaoks.....	27
3.3.2 Esimese etapi testimine.....	28
3.4 Projekti teine etapp .....	29
3.5 Projekti kolmas etapp.....	30

3.5.1	Kolmanda etapi planeerimine .....	30
3.5.2	Kolmanda etapi arendustööd ja testimine .....	31
4.	Analüüs .....	33
4.1	Arutelu ja järeldused .....	33
4.2	Soovitused .....	36
	Kokkuvõte .....	39
	Summary.....	41
	Kasutatud kirjandus.....	43
	Lisad .....	47

## Mõisted

**Eriväärtus** (*Specific value*) – Tooteinfo mis on erinev igas planogrammis.

**ETK** – Eesti Tarbijateühistute Keskühistu.

**My.spaceman** – ACNielseni planogramide haldamise tarkvara (ühtlasi ka projekti nimi).

**Kaardistuse fail** (*MAP file*) – Fail kus kirjeldatakse *connectivity* struktuur *spaceman* jaoks

**ODBC** - avatud andmebaasipöördus võimaldab pöörduda mistahes tüüpi andmebaasi poole mistahes rakendusprogrammist

**Pinnahalduse osakond** – Kaupluste väljapanekute kujundamise eest vastutav osakond.

My.spaceman projekti algataja.

**Planogramm** – Joonis toodete paigutusest müügiriulitel.

**RT** (*Request Tracker*) – avatud lähtekoodiga tehnilise toe tarkvara, mis aitab jälgida ja hallata kasutajate poolt antud ülesandeid või probleeme.

**Skeem** (*schema*) – andmebaasi struktuuri kirjeldus nii relatsioonandmebaaside kui ka objektorienteeritud andmebaaside puhul.

**Spaceman** – ACNielseni planogramide loomiseks kasutatav tarkvara.

**Tootekaart** – Infosüsteemides ühe toote juurde kuuluv info.

**Üldvärtus** (*Generic value*) – Tooteinfo mis on kõikides planogramides ühesugune.

# 1. Sissejuhatus

Konkurentsieelise säilitamiseks peab ettevõtte pidevalt kohanema turuga. See tähendab, et olemasolevaid äriprotsesse tuleb pidevalt ümber kujundada, asendada või töötada välja efektiivsemad protsessid. Et aga tänapäeval on ettevõtetes äriprotsesside tagajateks sageli erinevad infosüsteemid, on oluline, et ettevõtte oleks võimeline rakendama äriprotsesse maksimaalselt toetavaid infosüsteeme. Pahatihti ei osata aga arendusprojektide alguses lõplike vajadusi piisavalt adekvaatselt hinnata ning sellest tingituna luuakse lahendusi, mis ei vasta äriprotsesside nõudmistele. Konkurentsieelise suudavad säilitada ainult need ettevõtted, kes arendusprotsessi käigus on võimelised jooksvalt kohanema äriprotsesside nõudmistega [11]. See aga tähendab et ettevõtted peavad rakendama arendusprotsessides arendusmetoodikaid, mille abil on seda võimalik teha.

Traditsioonilise arendusmetoodika põhimõtetest lähtuvates projektides kirjeldatakse lahenduse spetsifikatsioonid projekti alguses. Projekti alguses pole aga äripoolel enamasti välja kujunenud selget visiooni lahendusest, mida soovitakse, vaid on ainult ettekujutus oma vajadustest ning seetõttu on väga raske tarkvarale esitatavaid nõudeid täpselt kirjeldada. Alles projekti jooksul tekib äripoolel ning arendajatel parem arusaamine äriprotsessidest ning nõuetest, millele loodav lahendus peab vastama. Lisaks võivad algselt paika pandud äripoolle nõuded muutuda projekti jooksul, seega vajadus erinevate muudatuste järgi võib tekkida alles projekti lõpus. Projekti lõpus on aga muudatuste tegemine on väga kulukas.

Traditsioonilistele arendusmetoodikate kõrval on aina enam populaarsust kogunud agiilsed tarkvaraarenduse meetodid. Agiilsed tarkvaraarenduse meetodid on paindlikud arendusmetoodikad, kus selle asemel, et lähtuda fikseeritust nõuete spetsifikatsioonist, töötatakse spetsifikatsioon välja projekti jooksul. Agiilsed meetodid rõhuvad eelkõige inimeste vahelisele suhtlemisele ning äripoolle pidevale kaasamisele arendusprotsessi.

Antud magistritöös keskendub autor teguritele, millega peab ettevõtte arvestama, kui soovitakse traditsioonilise arendusmetoodika asemel juurutada agiilset arendusmetoodikat. Magistritöös analüüsitakse Eesti Tarbijateühistute Keskühistus (ETK) läbi viidud arendusprojekti, kus rakendati agiilset metoodikat.

Käesoleva magistritöö eesmärgiks on määratleda agiilse arendusmetoodika rakendamisel esinevad probleemid ETK-s ning sellest lähtuvalt töötada välja soovitusi, mis aitaksid tulevikus parendada arendusprojektide läbiviimise protsessi.

## **1.1. Aktuaalsus**

Tarkvara arendusprotsessi muutmine ei tähenda ainult vahendite ja tehnoloogiate asendamist uutega, vaid ka muudatusi organisatsiooni tasemel. Sageli nii organisatsiooni struktuuris kui ka toimimise põhimõtetes. [19]

Sageli aga agiilse meetodika juurutamisel, metoodika rakendajad ei teadvusta kui mahukas ülesanne see on ja mis muutusi see endaga kaasa toob [15] ning paljude ettevõtete vastuvõtlikkus agiilsetele metoodikatele on alles veel algusjärgus [5]. Seetõttu on jätkuvalt vajadus teadmiste ning nõuannete järgi, mis aitaksid parendada agiilse metoodika juurutamise protsessi ning maandada sellega kaasnevaid riske.

Käesolev magistritöö on eelkõige oluline ETK seisukohalt, kuid samas võib pakkuda huvi ka kõigile teistele kes puutuvad kokku agiilse metoodikaga igapäevaselt või nendele kellel on tulevikus plaanis agiilset metoodikat juurutada.

## 1.2. Eesmärk

Antud magistritöö eesmärgiks on määratleda agiilse arendusmetoodika rakendamisel esinevad probleemid ETK-s ning sellest lähtuvalt töötada välja soovitusi, mis aitaksid tulevikus parendada arendusprojektide läbiviimise protsessi.

Eesmärgist lähtuvalt seab autor järgmised ülesanded:

- 1) Tuua välja koskmudeli puudujäägid võrreldes agiilse metoodikaga.
- 2) Anda ülevaade agiilsest metoodikast ning agiilse metoodika juurutamiseks olulistest teguritest.
- 3) Teha kindlaks võimalikud probleemid, mis raskendavad agiilse metoodika rakendamist ETK-s.
- 4) Leida soovitusi tulevaste projektide läbi viimiseks, kasutades agiilset metoodikat ETK-s.

## 1.3. Probleemi kirjeldus

Eesti Tarbijate Ühistute Keskühistu on 100% Eesti kapitalil põhinev Eesti Tarbijateühistute Keskühistu on vanim ja suurim Eesti jaekaubanduse vallas ühiselt tegutsev grupp. ETK Grupis tegutseb 19 tarbijate ühistut, millel on omakorda kokku ligi 83 000 klientomanikku ehk ühistu liiget. ETK Gruppi kuulub 366 müügi- ja teenindusüksust üle Eesti, sealhulgas 280 on toidu- ja esmatarbekauba kauplused, 256 on ketikauplused Maksimarket, Konsum, A ja O ning millele lisandub 20 E-Ehituskeskust. ETK on Eestis vanim ning suurim tegutsev jaemüügiga tegelev ettevõtte. Esimesed tarbijate ühingud asutati 1902.a. Antslas ja Sindis. ETK kui keskühistu asutati Eesti Tarvitajateühisuste Liiduna 7. mail 1917.a. 1919.a. muudeti Liidu põhikirja ja organisatsioon sai endale ka uue nime – Eesti Tarvitajateühisuste Keskühisus ehk lühendatult ETK. [9]



ETK-s on olnud mitmetes arendusprojektides probleemiks, et loodava IT-lahenduse puudujäävad funktsioonid tulevad alles projekti lõpus välja. Seetõttu tuleb juurde tellida täiendavaid arendustöid, kuid sageli ei suudeta ka lisaarendustega eemaldada puudujääke, mis on tekkinud juba projekti alguses. Mitmed infosüsteemid seetõttu ei toeta täielikult äriprotsesse ning on madala kasutajamugavusega.

ETK puhul on tegemist suure organisatsiooniga millel on pikk ajalugu, mistõttu ETK-s eksisteerib väga palju tööprotsesse, mis on ajalooliselt välja kujunenud. Enamus tööprotsessidest on väga keerukad ning mahukad. Kasutades traditsioonilisi meetodikaid arendusprojektides, on seetõttu väga raske projekti alguses analüüsida, mida täpselt vajatakse ning kuidas loodav lahendus täpselt peaks parandama äriprotsesside kvaliteeti. Seetõttu ei vasta sageli loodav lahendus projekti lõpus äripoole tegelikele ootustele.

Projekti jooksul areneb äripoole ning ka arendajate arusaamine sellest, kuidas loodav lahendus täpselt peaks parandama tööprotsesside kvaliteeti. Kui ETK-s oleks projekti jooksul võimalik täiendada spetsifikatsiooni vastavalt vajadusele, oleks ka võimalik luua lahendusi, mis vastaksid äripoole ootustele. Agiilsed tarkvaraarenduse meetodikad on paindlikud arendusmeetodikad, kus ei lähtuda fikseeritud nõuete spetsifikatsioonist, vaid töötatakse spetsifikatsioon välja projekti jooksul. Rakendades arendusprojektides agiilseid meetodikaid, oleks äriprotsesse võimalik parandada oluliselt rohkem kui traditsiooniliste arendusmeetodikatega.

Suur osa ETK äriprotsessidest on seotud jaemüügiga ning seetõttu on kasutusel väga mitmeid erinevaid IT lahendusi, mis on spetsialiseerunud konkreetsele tegevusele. Üheks selliseks lahenduseks on kaupluste väljapanekute loomise tarkvara, mille abil on võimalik kujundada visuaalselt toodete paiknemist müügiriivulitel. Seda on vaja selleks, et kogu müügipinda võimalikult efektiivselt ära kasutada.

Kauplustes kõige kallimad tooted paigutatakse tavaliselt silmade kõrgusele, kuna üldjuhul märgatakse ning jäädakse pidama just nendele toodetele, mis asuvad silmade kõrgusel. Soodustooted paigutatakse enamasti riivulite otsadesse, kust lihtsalt ei ole võimalik märkamatu

mööda minna. Samuti tooteid, mis hästi kokku sobivad, eksponeeritakse kaupluses sageli kõrvuti. Näiteks kreekreid ja juustu või õunakooki ja koort. [28]

Selliste strateegiliste paigutuste leidmine on mahukas ning keerukas töö. Oluline on et toodete paigutus oleks võimalikult optimaalne, säilitades esindusliku ning müüva paigutuse ja samas tuleks kogu müügipind maksimaalselt ära kasutatud. Toodete paigutuste kujundamiseks kauplustele on ETK-s kasutusel väljapanekute haldamiseks vajalik tarkvara ACNielseni *spaceman professional*.

*Spaceman* võimaldab koostada riiulite väljapanekuid, ehk planogramme [LISA 1], ilma et reaalselt peaks toodet kuskile riiulisse paigutama. Planogrammid võimaldavad analüüsida, milline kaupade paigutus näeb visuaalselt parem välja ning milline toodete paigutus võtab kõige vähem ruumi müügiriivil. Sedasi on võimalik koostada planogramm valmis kontoris ning edastada see kaupluse töötajatel, kelle ülesandeks jääb ainult kaupade riiulitele tõstmine.

Kuigi *spaceman* näol on tegemist äärmiselt olulise töövahendiga väljapanekute kujundamisel, siis paraku on tööprotsessid, alates üksiku planogrammi loomisest kuni erinevate planogrammidesse muudatuste sisse viimisest väga aeganõudvad. Kõige enam kulub planogrammide koostamisel aega toodete lisamisele, kuna seda tuleb teha käsitsi. Kõik planogrammid salvestuvad ainult faili kujul ning nendes sisalduv info on ka ainult seal. See tähendab, et uue planogrammi loomisel tuleb alati kõik otsast peale teha. Sealjuures ka sisestada kõik tooted planogrammi. Kui on vaja teha muudatusi mis hõlmab mitmeid erinevaid planogramme, näiteks kampaaniatoodete ümber paigutamine, siis tuleb need alati teha planogramm planogramm haaval.

Planogrammide paremaks haldamiseks ning loomiseks on võimalik *spaceman* rakendus liidestada andmebaasiga ning kasutada täiendavat haldamise moodulit *my.spaceman*. *My.spaceman* rakenduse abil oleks võimalik oluliselt lihtsustata planogrammide loomise ning haldamise tööprotsessi kuna *my.spaceman* võimaldab hallata mitut planogrammi korraga ning lisada planogrammidesse tooteid otse andmebaasist.

## **1.4. Magistritöös kasutatav metoodika**

Magistritöö põhineb juhtumi analüüsil ning tegemist on empiirilise kvalitatiivse uuringuga.

Magistritöö teoreetilises osas annab autor ülevaate ning analüüsib agiilset metoodikat artiklites, raamatutes, veebilehtedes ja õppematerjalides avaldatud teabe alusel.

Praktilises osas analüüsib autor magistritöös vaadeldava projekti My.spaceman jooksul läbi viidud tegevusi ja nende planeerimist lähtudes agiilse tarkvaraarenduse metoodika põhimõtetest. Lisaks võetakse arvesse ka projektiga seotud dokumentatsioon nagu: projektiplaanid, lähteülesande kirjeldused ning materjalid projekti jooksul läbi viidud esitlustelt. Projekti lõpus viidi läbi mittestruktureeritud intervjuud projektis osalejatega.

## **1.5. Magistritöö struktuur**

Magistritöö koosneb neljast peatükist. Eraldiseisvateks osadeks on kasutatud mõistete seletusi ning lisaidsid.

Esimene peatükk on sissejuhatus, mis annab ülevaate teema aktuaalsusest, töö eesmärgist ning ülesannete püstitusest.

Teine peatükk annab ülevaate agiilsest metoodikast, selle olemusest ja päritolust ning võrdleb seda koskmudeli metoodikaga, tuues välja agiilse metoodika eelised ning koskmudeli puudujäägid. Lisaks antakse ülevaate enamlevinudatest agiilsetest metoodikates ning tuuakse välja olulised tegurid, mis mõjutavad nende juurutamist.

Kolmas peatükk annab ülevaate My.spaceman projektist ning projekti raames toimunud tegevustest.

Neljandas peatükis arutletakse ning analüüsitakse projektis toimunud tegevusi ning tuuakse välja soovitused edaspidiseks.

## **2. Agiilne metoodika**

### **2.1 Projektijuhtimise metoodikate ajalugu**

Enamik projektijuhtimise metoodikaid on tekkinud 20. sajandi teisel poolel. Nende väljatöötamisel on olnud oluline roll suurorganisatsioonidel ja sõjatööstusel. Suurettevõtete tegemised on mastaapsed ning nende läbiviimiseks on vajalik põhjalik eelplaneerimine. Sõjatööstuses oli pidevalt kiire uute relvade väljatöötamisega sest tuli ennetada vastaspoolt. Samas pidi relvade kvaliteet olema ülikõrge. Näiteks töötas Henry Gantt oma ribagraafikute meetodi välja esimese maailmasõja ajal sõjalaevade ehitusprojektide jaoks. Samuti oli sõjatööstuse jaoks mõeldud ka PERT võrkdiagramm tehnikad ja kriitilise tee meetod. [23]

### **2.2 Koskmudelil põhinevad arendusprojektid**

Koskmudel on järjestikuline disainiprotsess, mida kasutatakse enamasti tarkvaraarendus projektides. Visuaalselt kujutatakse koskmudeli tüüpi metoodikat, nagu nimigi vihjab, astmete kaupa järk järgulise protsessina. Koskmudel koosneb erinevatest faasidest: projekti algatamine, analüüsimine, disain, arendus, testimine, juurutamine ning hooldus. [30]

Koskmudel oli esimene formaalne tarkvara arendamisprotsess, mis on siiani kõige populaarsem meetod, kui projektide nõudmised on fikseeritud [24]. Koskmudelipõhise metoodika kasutamine arendusprojektides tähendab seda, et iga järgmine faas projektis jagatakse erinevale meeskonnale, et oleks tagatud parem kontroll projekti tegevuste ning tähtaegade üle [2]. Projekti algatamise

faasis määratakse projekti põhieesmärgid ning kujundatakse selge veendumus projekti vajalikkusest ja teostatavusest [20]. Järgmises etapis ehk analüüsi etapi käigus üritavad analüütikud analüüsida intervjuude ning modelleerimise keelte nagu UML abil äriprotsesse, et paika panna vajadused ja nõuded [18]. Sellele järgneb disainietapp, kus ärivajadused kirjeldatakse ümber IT-lahendusteks [18]. Kui disain on paigas siis toimub arendustöö, millele järgneb tervikliku täislahenduse testimine [2]. Kõige viimasena toimub tarkvara üleandmine ning ka hooldus, veendumaks, et kõik töötab sujuvalt [2].

## 2.3 Koskmudelil põhinevate arendusprojektide probleemid

Kuigi koskmudeli metoodika on ennast tõestanud võimeka metoodikana, millega on võimalik luua töötavat tarkvara, on koskmudelil ka teatud nõrkused.

- Nõuete koostamise faas külmutab skoobi, mis innustab projekti tellijaid lisama funktsionaalsust, mida algselt ei olnud ette nähtud.
- Ärirakenduste väärtuste ja nende mõjude ette hindamine on väga raske, kuna seal on palju määramatuid ning tundmatuid tegureid.
- Äriprotsesside analüüsimine on keerukas ettevõtmine. Enamasti on kasutajate kirjeldused küllastunud mitteolulistest detailidest, kuidas kunagi minevikus on tööd tehtud ning tegevustest, mida on harjutud aja jooksul tegema.
- Konkreetsete nõuete kirjeldamisel on piiri raske tõmmata, siis, kui puudub selge joon selle vahel, mis on oluline ja mis mitte. Näiteks, kui palju detaile on vaja kirjeldada või kui palju peab eeldama, mida nõuete lugejad teavad.
- Paljud valdkonnad ettevõttes ei ole formaliseeritud, mistõttu intuiitivsete äriprotsesside tõlgendamine üldisteks ning formaalseteks nõueteks võib olla keerukas.
- Projekti sisemiste või välimiste mõjude tõttu võivad nõudmised muutuda projekti jooksul.
- Projekti edenedes, nõudmistest arusaamine ning lahenduste välja töötamine areneb edasi ning võivad tekkida uued võimalused, mida on raske lisada projekti skoopi.

- Arendusmeeskonnal võib märkamata jääda olulisi äripoole vajadusi, kui äripool ise ei suuda väljendada korralikult oma soove ja ootusi.
- Arendajad ja testijad peavad tõlgendama dokumentatsioonis kirjeldatud, kuid nende arusaamine sellest võib olla ebapiisav ja poolik.
- Dokumentatsioon ning nõuded kipuvad ajast maha jääma kui hiljem toimuvad muutused. Ärianalüütikud tavaliselt ei hinda ümber nõudeid ja vajadusi ning seda, kuidas loodav lahendus nendega sobitub. [18]

## 2.4 Agile Manifesto

Agiilne, sõna kui selline tähendab eelkõige paindlikust ja nõtkust. Metoodika on iseenesest eksisteerinud ligi kaks aastakümnet, kuid mõiste agiilne võeti kasutusele alles veebruaris 2001. [14]

Veebruaris 11-13, 2001 kogunes Utah osariigis Wasatch mägede suusakuurordis konverentsile kokku 17 inimesest koosnev grupp tarkvaraarenduse spetsialiste, kus oli esindajaid ekstreemprogrammeerimisest, SCRUM, DSDM, Chrystal, ASD ning teistest arendusmetoodikatest. Selle konverentsi käigus koostati agiilse tarkvaraarenduse ühiseid väärtusi ning printsiipe kokkuvõtva *Agile Manifesto*.

*Agile Manifesto* ülistab lihtust ja koostööd ning toob oma põhiväärtustes välja, et kõige olulisem on just kliendi rahulolu läbi töötava tarkvara ning seda on võimalik saavutada läbi pidevalt kohaneva ja areneva protsessi.

- Inimesed ja suhtlemine on tähtsam kui protsessid ja tööriistad
- Töötav tarkvara põhjaliku dokumentatsiooni asemele
- Kliendiga koostöö lepingute läbirääkimiste asemele
- Reageerimine muutustele plaani järgimise asemele

Järgnevalt on välja toodud *Agile Manifesto* aluseks olevad 12 printsiipi, mis aitavad paremini mõista, mida agiilne tarkvaraarendus endast kujutab. (allikas: <http://www.agilemanifesto.org>)

1. Tarkvaraarenduse põhieesmärk on rahuldada kliendi vajadusi läbi varajaste ning pidevate väärtustandvate tarkvara väljalasetega.
2. Olla valmis muutusteks projektis, ka selle lõppjärgus, see annab kliendile suurema eelise konkurentsis.
3. Vii kliendini töötav tarkvara võimalikult tihti, mõne nädala või kuu tagant.
4. Äripool ja arendajad peavad töötama koos päevast päeva kogu projekti vältel.
5. Projekti tuleb kaasata motiveeritud inimesed. Neile tuleb anda töökeskkond ning vajalik tugi ja usaldada neid, et nad oma töö ära teevad.
6. Kõige efektiivsem meetod info vahetamiseks arendusmeeskonnas on näost näkku vestlused.
7. Progressi saab hinnata ainult töötava tarkvara järgi.
8. Agiilne protsess propageerib ühtlast ja pidevat arendus tempot. Sponsorid, arendajad ja kasutajad peaksid suutma hoida ühtset tempot lõputult.
9. Pidev tähelepanu tehnilisele tippasemele ja heale disainile suurendab agiilsust.
10. Lihtsustamine on oluline kunst maksimeerida töö hulka, mida ei pea tegema.
11. Parim arhitektuur, nõuded ja disain luuakse iseorganiseeruvates meeskondades.
12. Meeskond leiab regulaarsete intervallide tagant mooduseid, kuidas kohaneda ja ümber organiseeruda, et meeskond oleks efektiivsem.

## **2.5 Agiilse ja traditsioonilise metoodika võrdlus.**

Agiilseid metoodikaid rakendavates projektides üritatakse projekti erinevates etappides keskenduda eelkõige funktsioonide loomisele. Traditsioonilistes metoodikaid rakendavates projektides on põhiline rõhk ülesannete järjestikusel täitmisel, mistõttu traditsioonilised metoodikad on väga bürookraatlikud ning mehaanilised. Et oleks võimalik lühikeste ajavahemike tagant luua funktsionaalsus, mis moodustab ühe osa lõplikust tarkvarast, peab olema agiilset

metoodikat rakendatavates projektides meeskond ja organisatsioon pigem orgaaniline ning paindlik. [7]

Traditsioonilisi meetodeid kasutavad arendajad üldjuhul eeldavad, et lõppkasutajad on lühinägelikud ning ei tea täpselt, mida nad vajavad. Seetõttu tekib arendajatel ettekujutus, et nemad teavad paremini kasutaja vajadusi. Seega üritavad arendajad lisada loodavale lahendusele juurde funktsioone, mida äripool realselt ei vaja. Agiilset arendusmetoodikat rakendavad meeskonnad lähtuvad seisukohast, et nii äripoolel kui ka arendajatel ei ole projekti alguses arusaama kõikidest vajadustest, mida süsteem peab rahuldama. Sedasi suudavad arendajad koostöös äripoolega välja töötada vajalikud detailid projekti jooksul. [5]

Traditsioonilisi meetodikaid rakendavas projektis juhitakse projekti plaanipäraselt, valitse ja käsuta stiilis. Agiilset metoodikat rakendavates projektides lähtub projektijuhtimine pigem koostöö põhimõttest ning projektijuhi eesmärk on koordineerida ning nõustada. [17][19]

## 2.6 Levinumad agiilsed metoodikad

Tarkvaraarenduses on mitmeid erinevaid arendusmetoodikaid mis arendavad tarkvara agiilselt, millest igaüks rõhutab erinevaid aspekte tarkvaraarenduses. Ekstreemprogrammeerimine keskendub iteratiivsele ning kiirele arendamisele, scrum fokuseerub projektide haldamisele, FDD (*Feature Driven Development*) on väga tugevalt keskendunud disainile ja planeerimisele ja ASD (*Adaptive Software Development*) on suunatud tugevalt kontseptsioonide ja suuniste rakendamisele, et soodustada kiiret arendust läbi prototüüpide loomise [24]. Ühiseks jooneks on kõigil agiilsetel metoodikatel on luua koostöös äripoolega järkjärguliselt töötav tarkvara. Eelmainitud agiilsetest metoodikatest on kõige populaarsemad ja enim rakendatud ekstreemprogrammeerimist ning scrum [4].



## 2.6.1 Ekstreemprogrammeerimine

Ekstreemprogrammeerimises suhtlevad programmeerijad pidevalt oma klientidega, hoides samal ajal kirjutatavat koodi võimalikult puhta ja lihtsana. Läbi testimise saadakse koheselt tagasisidet projekti esimesest päevast alates. Tekkinud muudatuste vajadused viiakse sisse vastavalt kliendi soovidele ning tarkvara toimetatakse kohale võimalikult kiiresti. [10]

Ekstreemprogrammeerimine on kõige populaarsem ning enim praktiseeritud agiilne tarkvaranduse meetod [25], mis on Kent Beck'i poolt loodud tarkvaraarendus metoodika [21]. Fokuseerides eelkõige kiirele ja järkjärgulisele arendusele [2] [23]. See meetod on eelkõige sobilik väikeste ja keskmist suurusjärku projektide jaoks [24]. Ekstreemprogrammeerimine on väga programmeerija keskne meetod ning on ainuke meetod agiilsetest metoodikates, mis üldse keskendub konkreetsele programmeerimisele ning väga tugevalt testimisele [25].

Ekstreem programmeerimine lähtub tarkvaraarendamisel 5-st reeglist. [10]

Lihtsus - arendusprotsess hoitakse võimalikult lihtsana. Tehakse ära ainult vajalikud tööd ja ei midagi üleliigset. Arendusprotsessi jooksul liigutakse edasi väikeste lihtsate sammude kaupa ning probleemidega tegeletakse siis kui need ilmnevad.

Suhtlus – igauks on meeskonna liige. Suhtlemine toimub meeskonnas näost näkku ning probleemidele leitakse ühiselt lahendusi.

Tagasiside – iteratsioonide tagasisidesse suhtutakse väga tõsiselt. Võetakse kuulda kõiki tellija ettepanekud ning vajadusel viiakse sisse muudatused.

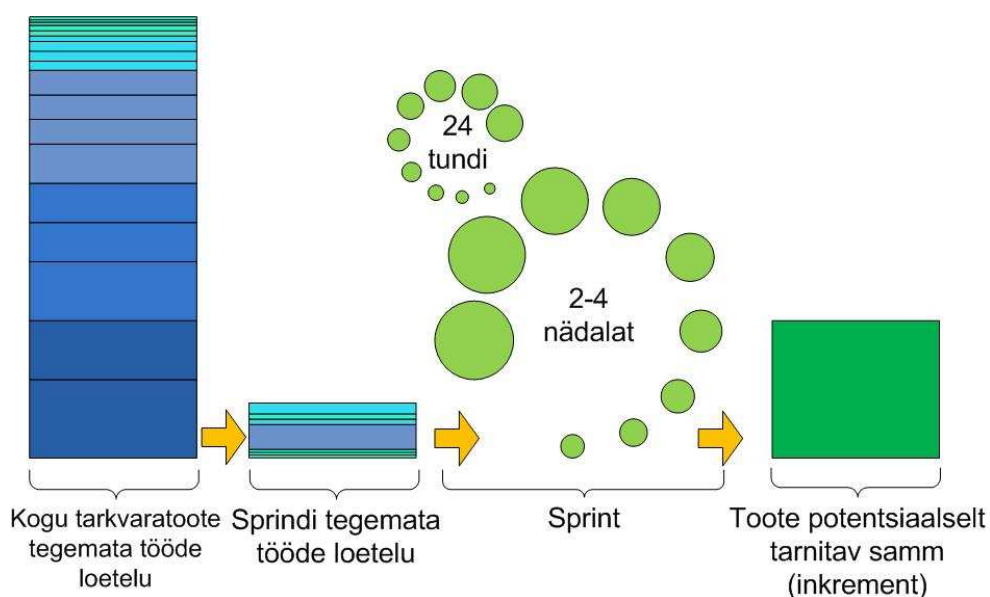
Tunnustus – oluline on meeskonnatöö. Kõik projektis osalejad tunnustavad ja julgustavad üksteist. Arendajad hindavad kliendi teadmisi ja vastupidi.

Julgus – progressi ja hinnanguid antakse ausalt ilma vabandusteta. Julgetakse tunnistada oma vigu, nendest õppida ja kohaneda olukorraga.

## 2.6.2 Scrum

Scrum arendusmeetodi pakkus välja esimest korda Ken Schwaber [29]. Scrum on eelkõige paindlik, kohanev, empiiriline, produktiivne ning iteratiivne meetod, kus tarkvara arendamiseks rakendatakse põhimõtteid industriaalsest protsessi kontrolli teoriast [24]. Scrum on eelkõige projekti haldamisele suunatud agiilne metoodika [25].

Scrumi keskne idee seisneb selles, et kogu projekti jooksul ei ole võimalik ette ennustada paljusid protsesse. Ainukesed konkreetsed planeeritud etapid scrum projektis on esimene ja viimane faas (planeerimine ja lõpetamine). Scrumi elutsükkel (joonis 1) koosneb mitmetest arendusetappidest, ehk sprintidest. Projekti jooksul võib sprintide sisu muutuda vastavalt uute nõudmiste tekkimisele. [29]



Joonis 1. Scrumi elutsükkel. [3]

Scrum projekti käivitamisel, esialgse planeerimise ning disaini käigus koostatakse *backlog*, mis on kogu tarkvaratoote tegemata tööde kirjeldus. *Backlog* koosneb kasutajate kui ka arendajate ettepanekutest, soovidest ja ideedest. Scrum projektis kasutatakse kahte erinevat tüüpi *backlog*. Üks on toote *backlog*, kus on kirjeldatud kliendi soovid ja funktsionaalsed nõudmised. Toote

*backlog* muutub omakorda arenduse ehk sprindi *backlog*. Sprindi *backlog* on kirjas meeskondade ülesanded ning arendustööde kirjeldused. Toote *backlog* haldajad loovad spetsifikatsioonid arenduse jaoks ning arendusjuhid loovad ühe töötava funktsionaalsusega toote sprindi *backlog* alusel. [29]

Sprinte on projektis tavaliselt 3-4 ning need võivad kesta kaks kuni neli nädalat [3]. Sprindid on projekti etapid millal toimuvad arendustööd ning sinna ei kaasata äripoolt [25]. Peale igat sprindi tegemist vaadatakse see üle koos äripoolega ning lepitakse kokku järgmise sprindi tegevused. Kui on tekkinud muutusi nõudmistes või vajaduses siis lisatakse need järgmisse sprindi backlog.

Projekti ajagraafiku jälgimiseks kasutatakse läbipõlemise graafikut (*burndown chart*). Läbipõlemise graafik koostatakse *backlog* põhjal. *Backlog* koostamisel hinnatakse ära kui kaua aega kulub konkreetsete funktsioonide loomiseks. Summeerides *backlog* ajahinnangud saadakse kogu projekti eeldatav kestvus. Kui *backlog* kirjeldatud funktsioon saab valmis, lahutatakse reaalselt töö peale kulunud aeg kogu projekti ajast maha ning saadakse teada, kui palju töötunde veel teha on. [27]

## **2.7 Lepingute sõlmimine ning projektide eelarvestamine agiilses tarkvaraarenduses**

Agiilses tarkvaraarenduses projekti alguses ei koostata detailset loodava disaini, mistõttu on väga raske hinnata milline lõpptulemus tegelikult on ning kui palju töötunde sobiva lahenduse loomisele kulub. Seetõttu on keeruline projekti alguses eelarvet koostada ning lepinguid sõlmida. Lepingutüübid mida on harjutud kasutama traditsiooniliste arendusmetoodikate puhul ei kaitse agiilses tarkvaraarenduses osapoolte huve piisavalt. Tulenevalt projekti skoobist, tähtaegadest ning ka osapoolte omavahelisest usaldusest, peavad osapooled jõudma kokkuleppele, kuidas ja mille alusel projektis toimub tasustamine.

### **2.7.1 Fikseeritud aeg, maksumus ja skoop**

Sõlmides lepinguid fikseeritud aja, maksumuse ja skoobiga projektides muutub probleemiks vajadus kogu skoop täpselt spetsifitseerida juba projekti alguses, mis on väga keeruline ja aeganõudev. Sellised projektid muutuvad sageli vaidlusteks skoobi tõlgendamise teemal, kus arendaja püüab läbi saada kõige lihtsamate lahendustega ning tellija omakorda tahaks viimseni välja nõuda kõik, mis hankedokumentides kirjas, isegi kui seda projekti käigus selgunu põhjal reaalselt vaja pole. Selliste projektide edu sõltub poolte võimest üksteist usaldama hakata ning projekti tegeliku eesmärgi saavutamise nimel tihedat koostööd teha, vormistades vajalikud muudatused tagantjärele.[1]

### **2.7.2 Fikseeritud kogumaksumus ja aeg**

Siin teadvustavad osapooled, et etteantud eelarve raames tuleb kindlaks tähtajaks süsteem valmis saada. Projekti alguses fikseeritakse süsteemi visioon, ärieesmärgid, põhifunktsionaalsused ja kasutajagrupid, aga funktsionaalsused täpsustuvad ja muutuvad töö käigus. Selle, tellijale rohkem kaitset pakkuva, lepingumudeli edukus sõltub sellest, kui hästi suudetakse viimati mainitud protsess paika panna. Kui poolte vahel on mõistlik usaldus, mõlemad peavad silmas projekti alguses paika pandud ärieesmärke ja töötavad skoobi täpsustamise ja iteratsioonidesse planeerimise nimel ühiselt (milleks agiilsetes meetodikates, nt Scrumis, on õnneks üsna täpsed juhised olemas), töötab see lepinguvorm väga hästi. [1]

### **2.7.3 Aeg ja raha**

Aeg ja raha tüüpi lepingute puhul tasustatakse ainult tehtud töö eest. Enamasti maksab tellija arendajale perioodiliselt tehtud töötundide eest. Aeg ja raha tüüpi lepingud on kõige sobivam lepinguvorm, kui nõudmised on pidevalt muutuvad [6]. Miinuspoolelt eeldab antud

lepinguvorm suurt tellijapoolset usaldust täitja suhtes, samuti ei ole siin täitjal välist motivatsiooni oma tööd efektiivsemaks muuta [1].

#### **2.7.4 Osade kaupa tellimine raamlepingu alt**

Osade kaupa tellimisel raamlepingu alt on võimalik kasutada siis, kui tellija ja arendaja jõuavad kokkuleppele funktsioonides, mis on vaja luua ning mis nende maksumused on. Erinevate funktsionaalsuste maksumuste liitmisel on võimalik arvestada kogu projekti eeldatavat kulu. Peale iga funktsiooni valmimist hindab selle üle audiitor ning tellija maksab realselt loodud funktsioonide eest, mitte selle järgi, mis algselt plaaniti. Selline lepingu tüüp võimaldab kasutajal muuta oma nõudmisi ning motiveerib arendajat looma võimalikult kasuliku ning töötava tarkvara.[6]

Probleemid võivad tekkida erimeelsuste tõttu selle osas, kas osadesse mineva töö mahtude hindamine on tasustatav töö või mitte [1].

### **2.8 Äripoole osalemine arendusprojektides**

Projektides, kus rakendatakse traditsioonilist tarkvaraarendusmetoodikat piirdub arendaja ja arenduse tellija koostöö enamasti projekti alguses nõuete kirjeldamisega ning lõpus tagasiside andmisega. Kontrastina on agiilises tarkvaraarenduses tellija koostöö väga oluline ning vajalik edutegur projektide õnnestumiseks. Agiilsete metoodikate rakendamisel on oluline, et tellija annaks arendusmeeskonnale pidevalt tagasiside lühikeste intervallide tagant. Seetõttu agiilseid metoodikaid rakendades kaasatakse tellija kogu projekti vältel, näiteks nii kasutuslugude koostamisel kui ka toote omaduste analüüsimisel. Pahatihti võib aga juhtuda, et realses elus tellija ei ole nii aktiivne kui arendaja sooviks. Seda võivad põhjustada mitmed erinevad tegurid. Projekti tellija on skeptiline agiilsete metoodikate efektiivsuses. Arendajad paiknevad tellijast kaugel, kas teises linnaosas või üldse teises linnas ning see muudab raskeks näost näkku

kohtumiste korraldamise. Tellijal ei ole piisavalt aega tegeleda projektiga või puuduvad piisavalt võimekad tellijate esindajad kes suudaksid kirjeldada arendajatele oma vajadusi. Kui tellijat ei ole võimalik kaasata projekti piisavalt aktiivselt, tekib probleeme arendajatel tagasiside saamisega, nõuete ja ohukohtade kirjeldamisel, mis võib põhjustada languse arendajate motivatsioonis. [12]

## 2.9 Iseorganiseeruv meeskond

Agiilises arendusmetoodikas püüeldakse ise oma tööd korraldavate ning koordineerivate meeskondade ehk iseorganiseerivate meeskondade poole. See on samm eemale traditsioonilisest käsuta ja valitse mentaliteediga keskkonnast. Iseorganiseerival meeskonnal kui ühtse tervikuna on omad teadmised, motivatsioon ja oskused. Meeskonda koheldakse kui juhtkonna ning tellija partnerit. Sedasi on meeskonnal võime anda protsessidest ülevaadet, mõjutada otsuseid ning läbi rääkida kohustuste jaotamise üle. [14]

Kui arendajad on harjunud traditsiooniliste käsuta ja kamanda metoodikatega, siis see raskendab iseorganiseerivate meeskondade tekkimist. Inimeste kogemused ja suhtumine ei muutu hetkega ning töökorralduse muutumine avatumaks ning suhtlemisele orienteerituks võib põhjustada agiilse metoodika põhimõtete väärkasutamist. Agiilsete metoodikate rakendamise puhul on oluline, et projektimeeskonna liikmete vahel toimuks avatud suhtlemine. Sedasi on võimalik anda üksteisele tagasisidet ning vajadusel ka tuge kui projektis peaks tekkima inimressurssidest puudus. Kui arendajad tunnevad igapäevaste koosolekute jooksul, et nad peavad projektijuhile pidevalt ette kandma seda, kas töö on tehtud, võib tekkida olukord, kus arendajad sulguvad endasse ning ei räägi avameelselt probleemidest ning plaanidest. See aga tähendab, et inimesed keskenduvad oma tegemistele ning meeskonna terviklik koordinatsioon on halvatud. [17]

Arendajad ei ole agiilsetes projektides enam ainult koodikirjutajad vaid nad peavad läbi rääkima ning lahendusi leidma, mis tähendab nende vastutusala laieneb. Samas regulaarsetest

koosolekutest võib arendajale jääda mulje, et nad on pideva jälgimise all, seetõttu võib tekkida hirm, et nende töö on läbinähtav ning nende puudujäägid ja saamatused tulevad ilmsiks. [8]

## **2.10 Agiilsete metoodikate juurutamine organisatsioonis**

Organisatsioonikultuur on möödapääsmatu kontekst, milles organisatsiooni elu käib [13]. Organisatsioonikultuur iseloomustab organisatsiooni psühholoogiat, suhtumist, uskumisi ning väärtushinnanguid, olles eriline kogum normidest, millele tuginedes inimesed ja grupid suhtlevad omavahel ning väliste partneritega [21]. Agiilsete metoodikate juurutamiseks peab organisatsioon olema paindlik, toetama koostöötamiseks vajalikku keskkonda ning samas ei tohiks olla liiga bürokraatlik [16]. Organisatsiooni paindlikkus tuleneb kolmest faktorist: oskusest avastada uusi võimalusi konkurentsieelise tõstmiseks, oskusest olemasolevaid varasid ja teadmisi rakendada ning võimest kohaneda järskude äritingimuste muutustega [26].

Agiilse metoodika juurutamises on oluline inimeste suhtumine. Nokias läbi viidud uuringus [15] selgus, et mida enam on inimestel kogemusi agiilsete metoodikatega, seda positiivsemalt suhtutakse agiilsesse metoodikasse. Isegi kui inimesed algselt ei soovinud juurutada agiilset metoodikat, siis mida enam osaleti projektides, kus rakendati agiilset metoodikat seda vähem sooviti tagasi pöörduda traditsiooniliste metoodikate juurde.

Tarkvara arendusprotsessi muutmine ei tähenda ainult vahendite ja tehnoloogiate asendamist uutega, vaid keerukat muutust organisatsiooni tasemel. Muutus peab hõlmama struktuuri, kultuuri ning juhtimise organisatsiooni juhtimise võtteid. Agiilne metoodika ning organisatsioonid hõlmavad endas väga palju muutujaid ning üheselt on raske soovitada lahendusi, mis sobiksid kõigile organisatsioonidel. Seetõttu iga organisatsioon peab ettevaatlikult liikuma agiilse arenduse poole valides endale selleks sobivad meetodid. [19]

### 3. Agiilse metoodika rakendamine ETK-s

#### 3.1 Projekti My.spaceman tutvustus

Toodete paigutuse kujundamiseks kaupluste müügiriulitel on ETK-s kasutusel ACNielseni kommertstarkvara *spaceman professional*, mille abil on võimalik luua planogramme [LISA 1]. Planogrammide alusel paigutatakse kaubad müügiriulitele kõikides ETK kauplustele ning planogramme loovad ETK pinnahaldus osakonna töötajad. Kuna aga ETK-s on igas kaupluses müügisaalid erineva suurusega, siis iga kaupluste jaoks koostatakse eraldi planogrammid. Planogrammide koostamine on aga pikk ja keerukas protsess, millest põhiline aeg kulub toodete lisamisel planogrammi. Kui planogrammid on vaja hiljem muudatusi teha, näiteks toodete asendamisel, siis tuleb seda teha planogrammide kaupa, kasutades *spaceman*.

*My.spaceman* on kommertstarkvara mis on ACNielseni poolt loodud ning mille abil on võimalik hallata juba loodud planogramme ning uutesse planogrammidesse tooteid lisada.

*My.spaceman* rakenduses pakutavad lisavõimalused.

- Toote atribuutide muutmist, kustutamist ning asendamist (sh. terve toote asendamist).
- Toodete importimist planogrammidesse.
- Erinevate planogrammide ja toodete grupeerimist ning kategooriatesse jagamist.
- Planogrammide kasutajatele rolle määrata.
- Eksportida planogramme.
- Raporteid koostada.

*My.spaceman* lahenduse kasutamiseks on vajalik *connectivity*. *Connectivity* on avatud disainiga andmebaas, kuhu on võimalik salvestada planogramme ning planogrammid kasutatavaid tooteid. Tulenevalt erinevates äri vajadustest kohaldatakse *connectivity* struktuur igale ettevõttele vastavaks, sest vastavalt äriprotsesside erinevusele ettevõtetes on *connectivity* vaja säilitada osa tooteinfot *specific* andmeväljades.



Projekti *My.spaceman* eesmärgiks oli planogrammide haldamiseks vajaliku rakenduse *my.spaceman* juurutamine ning *connectivity* liidestamine ETK infosüsteemidega.

ETK poolt osalesid projektis pinnahalduse osakond, kelle poolt oli kaasatud projekti pinnahalduse osakonna juhataja ning 2 töötajat. IT-osakonna poolt oli kaasatud projektijuht ning kaks arendajat. ACNielseni poolt osaleb projektis ACNielseni esindaja.

*My.spaceman* projekti puhul lähtutakse üldistest agiilsetest põhimõtetest tarkvaraarenduses tuginedes, *Agile Manifesto* printsiipidele: inimesed ja suhtlemine on tähtsam kui protsessid ja tööriistad, töötav tarkvara põhjaliku dokumentatsiooni asemele, reageerimine muutustele plaani järgimise asemele.

## **3.2 Projektiplaani koostamine**

### **3.2.1 Esialgne plaan**

Pinnahalduse osakond algselt soovis juurutada ainult *connectivity* andmebaasi. Algne projektiplaani koostati selle põhjal, et *connectivity* andmebaas luuakse ETK arendusmeeskonna poolt. Kuna esialgse hinnangu põhjal ei paistnud ülesanne olevat keeruline, sest *spaceman* ühendub *connectivity* andmebaasiga ODBC kaudu ning andmebaas ise on avatud disainiga ning koosneb viiest tabelist kus planogrammide infot hoitakse.

Üsna pea peale esialgse plaani koostamist selgus, et ACNielsenil on olemas ka *my.spaceman* nimeline tarkvaralahendus, mis annaks väga tugevat lisaväärtust pinnahalduse osakonna töö optimeerimises. *My.spaceman* rakenduse kasutusele võtmise eelduseks oli aga see, et see peab ühilduma *connectivity* andmebaasiga. Kuna pinnahalduse osakond hakkas selle vastu huvi tundma, siis otsustati ka *my.spaceman* juurutamine projekti skooopi lisada. Selle tulemusena selgus, et kuigi ilmselt ei tekitaks suuremaid probleeme *spaceman* liidestamine *connectivity* siis

*my.spaceman* täislahenduse jaoks andmebaasi loomine võib osutada tunduvalt keerukamaks, kuna ükski ETK arendaja ei olnud antud tarkvaraga enne kokku puutunud mistõttu ei olnud võimalik hinnata adekvaatselt ohukohti. Sellisele järeldusele jõudmine tähendas seda, et kindlaim moodus töötava tarkvara saamiseks on *connectivity* andmebaasi arendus tellida ACNielsenilt. Seega siis otsustati, et ACNielsen loob ise andmebaasi ning disainib selle ETK ärivajadustele vastavaks kuid *connectivity* liidestamine ETK olemasolevate infosüsteemidega jääks ETK enda kanda. See aga tähendas, et esialgne projektiplaan tuli täielikult ümber teha mille tulemusena projekt jaotati kolme etappi.

### 3.2.2 Üleminek Agiilsele projektiplaanile

Projekti jaoks koostatud esialgse plaani järgi ei olnud enam võimalik edasi minna, siis tundus kõige mõistlikum läheneda sellele etapiliselt ning lisaks käesoleva magistritöö autoril tekkis huvi agiilsete meetodikate vastu, siis otsustati projekt viia läbi lähtudes agiilsetest põhimõtetest.

Projektis plaaniti etappide kaupa järgnevad tegevused:

1) *Connectivity* andmebaasi loomine:

Koostöös ACNienseni, äripoolle ning IT vahel kaardistatakse ära ETK *connectivity* andmebaasi eripärad ning disainitakse see ETK ärioloogikale vastavaks. Leitakse kokkupuutepunktid ETK infosüsteemide vahel tulevikus liidestamiseks ning võttes aluseks ETK soovid teostab ACNielsen vajalikud arendustööd. Lahenduse vastuvõtmisel viib ETK läbi omapoolse vastuvõtu testi.

2) *My.Spaceman* rakenduse juurutamine:

Juurutatakse *my.spaceman* lahendus. Viiakse läbi vastuvõtu testimine.

### 3) Olemasolevate infosüsteemidega liidestamine:

Kui on eelnevad etapid on läbitud ning töötab terviklik lahendus *spaceman* ning *my.spaceman* koos *connectivity* siis ETK arendusmeeskonna eesmärgiks jääb liidestada *connectivity* teiste infosüsteemidega, kus on vajalik tooteinfo.

Sellisel kujul plaan koostati 2010 aasta novembris. Projekti kestvuseks plaaniti kolm ja pool kuud, millest kaheksa nädalat moodustasid ACNielseni poole ETK vajaduste kaardistamine ning arendustöö *connectivity* andmebaasi loomiseks ning testimiseks. Samas kuna oli teada, et Eestis toimub 01.01.2011 valuutavahetus ning käibelt kaob ära Eesti kroon mille asemele tuleb Euro. Seetõttu oli hõivatud väga suur hulk IT ressursist, My.spaceman projekti alguses leiti, et fikseeritud projekti tähtajad ei pruugi olla reaalsed. Eesmärgiks seati siiski novembri jooksul ära kirjeldada vajalik informatsioon selleks et ACNielsen saaks alustada oma arendustöödega ning siis vastavalt jätkata veebruarist võimalustele järgmiste etappidega.

## **3.3 Projekti esimene etapp: Connectivity andmebaasi loomine**

### **3.3.1 Vajaduste kaardistamine Connectivity jaoks**

Projekti esimene etapp koosnes põhiliselt kohtumistest ETK IT, pinnahalduse ja ACNielseni vahel. Kohtumiste jooksul üritati eelkõige fikseerida planogrammide haldamiseks vajalikku ärioloogikat ning millist informatsiooni on ETK jaoks vaja salvestada planogrammid.

Esimene etapp käivitus peale ACNielseni poolset projektiplaani kinnitamist ning samaaegselt kohe novembri alguses 2010 alustasid ETK IT- ning pinnahalduse osakond oma esialgsete vajaduste kirjeldamist ACNielseni esindajale. Kohtumiste organiseerimise osas võttis suure initsiatiivi enda peale pinnahalduse osakond.

Vajaduste vajadused kaardistati kahe nädala jooksul kohtumiste jooksul pinnahalduseosakonna juhataja, Nielseni esindaja ja IT projektijuhi vahel. Koosolekud kestsid enamasti 1-1,5 h. ning

esimesel nädalal toimunud kahel koosolekul tutvustas ACNielsen eelkõige süsteemi loogikat ning eripärasid, andes ülevaate *My.spaceman* ja *connectivity* võimalustest. Teise nädala jooksul toimusid kaks koosolekut ETK sisemiselt pinnahalduse ja IT vahel. Nende jooksul vaadati läbi põhjalikult ACNielseni poolt tutvustatud tarkvara lahenduse *my.spaceman* pakutavaid võimalusi ning vastavalt nendele leida kokkupuutepunktid nii ETK äriprotsessidega kui ka neid toetavate IT-süsteemidega. Kohtumiste jooksul analüüsiti näiteks erinevaid võimalusi, kuidas oleks kõige parem koostada ja hallata planogramme kaupluste sortimendi, tootepuu ning ketitunnustest lähtuvalt ja milliste ETK infosüsteemidega peab olema *connectivity* liidestatud. Pinnahaldus pani paika ka millised andmeväljad on *connectivity specific* või *generic* tüüpi väljad ning selle info alusel alustas Nielsen *connectivity* arendustööde juurde.

### 3.3.2 Esimese etapi testimine

Kogu *connectivity* andmebaasi ning *my.spaceman* süsteemse andmebaasi arenduse ning esialgse testimise teostas ACNielsen. Seejärel andis ACNielsen üle ETK-le andmebaasi skriptid, *connectivity* andmebaasi kaardistuse faili (*MAP file*) ning juhendid andmebaaside ülesesseadmiseks.

Seejärel alustati ETK poolse testimisega, kuid paraku ei suudetud testimist alustada õigeaegselt, kuna pinnahalduse osakonnal muutusid teised pooleli olevad projektid kriitiliseks ning seetõttu oli pinnahaldus enamuses tööajast hõivatud teiste projektidega. Kuna testimine pidi toimuma ETK IT ja pinnahalduse koostööna, siis ei olnud võimalik projektiga jätkata seni kuni pinnahaldus ei olnud võimeline leidma aega testimiseks.

*My.spaceman* projekti esimeses etapis sai testimist alustada poolteist nädalat hiljem, kui algelt oli planeeritud. Kuigi pinnahaldusel ei olnud ka selleks hetkeks teised projektid lõppenud, ei olnud need enam nõnda koormavad ning IT ja pinnahaldus leppisid kokku, et iga päev

pinnahaldus viib paar tundi läbi teste ning peale testimist vaadatakse tulemused koos IT-ga üle. Testimisel lähtus pinnahaldus tüüpilistest tegevustest planogrammide koostamisel. Kõige parem viis testimise tulemuste, probleemide ning järgmiste tegevuste, arutamiseks arendajate ja pinnahalduse vahel, kujunes 10-15 minutit kestvad kohtumised, mis toimusid enamasti pinnahalduse tööruumides. Enamus osalejatest seisis püsti, seda puhtalt sellepärast, et tegemist oli tavaliste tööruumidega, kus ei olnud kõigile piisavalt istumise kohti. Samas andis see hea võimaluse alati *spaceman* kasutajatel kohe tööarvutist näidata, milliseid muudatusi peaks tegema või probleemide esinemisel kohe näidata, kuidas probleem tekib ning, mida see põhjustab.

Testimise käigus avastati mõned tehnilised vead andmebaasi salvestamisel, mis suudeti ära parandada kiiresti ETK IT ja ACNielsen koostööna. Lühikeste, nii öelda testimise koosolekute jooksul käidi alati peale paranduste tegemist uuesti üle testlood ning kontrolliti, kas viga õnnestus parandada või kas on avastatud ka uusi probleeme peale vigade parandamist. Kuigi lühikesi koosolekuid testimise jooksul oli planeeritud viia läbi iga päev, siis paraku osade testimisel avastatud vigade põhjuste välja selgitamine ning parandamine võttis kauem aega ja nii ei olnud otstarbekas iga päev koosolekuid läbi viia. Pinnahaldus oli osaliselt hõivatud teiste projektidega ning seetõttu ei olnud võimalik koosolekuid läbi viia kindla rutiini järgi, kuna erinevatel päevadel oli pinnahaldusel erinevad ajad vabad. Seetõttu tuli sageli leida ja sobitada aegu, et arutada testimise tulemusi.

### **3.4 Projekti teine etapp**

Teise etapi eesmärgiks oli võtta kasutusele *my.spaceman* rakendus. Selle jaoks oli vajalik ülesse seada *my.spaceman* süsteemne andmebaas ning paigaldada kasutajate arvutitesse *my.spaceman* rakendus ja vastavalt seadistada. Peale seda oli plaanitud viia läbi järgmine testimine, kus testitakse *my.spaceman* funktsionaalsus.

Teises etapis jätkati testimist, kasutades sama metoodikat, mis esimeses etapis oli välja kujunenud. Lühikeste ja kiirete koosolekutega, kus käidi läbi erinevate testlugude tulemused. Otseseid vigu teise etapi testimisel ei avastatud, pigem tulid välja kasutajate lahtised küsimused *my.spaceman* funktsionaalsuste ja võimaluste kohta, millele aitas vastata Nielsen esindaja.

## 3.5 Projekti kolmas etapp

### 3.5.1 Kolmanda etapi planeerimine

Kolmandas etapis oli planeeritud teostada *connectivity* andmebaasi liidestamine ETK infosüsteemidega ning kogu liidestamist hõlmav arendus toimus ETK oma arendusmeeskonna poolt.

Enne koodi kirjutamise juurde asumist, toimus koosolek IT ja pinnahalduse vahel, mille jooksul käidi üle kõik andmeväljad, mis pidi ETK infosüsteemidest importima *connectivity* ning lepiti kokku tegevused, mis etapi jooksul on vaja teha. Suures plaanis olid andmeväljad kaardistatud täiesti projekti alguses, kui ACNielsenilt telliti *connectivity* arendus. Samas taaskord oli vajalik need andmed üle käia, kuna projekti jooksul oli pinnahaldusel tekkinud parem arusaam mida nad tahavad, mis järjestuses ning mis tingimustele mõned andmed peavad vastama. Näiteks kas hinnad on käibemaksuga või ilma, millist sortimenti puuduvat infot soovitakse jne. Valdava osa andmetest mida loodava liidese kaudu vaja edastada oli, moodustasid tootekaardil olevad erinevad andmed nagu nimetus, tootekoodid, tootja, päritolumaa jpm. Samuti soovis pinnahaldus, et võimalusel lisatakse juurde toodetele ka müügistatistika (kui mitu tükki nädalas on müüdüd, jne.) Tegemist oli üsna suure hulga erineva infoga, mis oli kohati erinevate infosüsteemide vahel ära jaotatud ja tuli leida tehniline lahendus, mis oleks võimeline sünkroniseerima infot erinevatest andmebaasidest. Teisest küljest oli olemas teatav risk, et andmeid on võimalik andmebaasi tasemel sünkroniseerida, kuid võib tekkida mingeid anomaaliaid kasutajapoolses rakenduses, ehk *spaceman* või *my.spaceman*. Seetõttu plaaniti hakata edastavat info hulka suurendama järk

järguliselt, saates algul ainult baasinformatsioon ühe toota kohta, mis on *spaceman* vaja planogrammide loomiseks: tootekood, nimetus ning mõõtmed. Eesmärgiks oli välja töötada andmebaasi tasemel protseduur, mis oleks võimeline edastama infot olemasolevatest infosüsteemidest *connectivity*.

### 3.5.2 Kolmanda etapi arendustööd ja testimine

Kogu arendustööde jooksul üritas arendusmeeskond jätkata juba välja kujunenud koosolekute vormi vahega, et pinnahaldus ei osalenud koosolekutel. Koosolekuid üritati korraldada võimalikult sagedasti vastavalt selle, mis töö arendajatel täpsemalt käsil oli. Etapi alguses oli vaja koostada andmebaasi skript, mis andmeid pärib ning eraldi tabelitesse paigutab *connectivity*. Kuna tegemist oli programmeerimise mõttes lihtsa ja lühikese ülesandega ning meeskonna liikmed asusid lähestikku siis suheldi omavahel jooksvalt vastavalt vajadusele. Samas kui arendajatel oli vaja andmebaasi tasemel ära kaardistada kõik andmed, mida liides vahendab ning viia need vajalikule kujule, siis oli otstarbekam koosolek planeerida 3-4 päeva pärast. IT projektijuht otsustas sellise vabama graafiku kasuks arendustöodes seetõttu, et pinnahaldus oli suuresti hõivatud teiste projektidega, mistõttu ei olnud võimalik kokku leppida konkreetset aega testimiseks. Seetõttu ei olnud ka arendustöödega kiirustamisel olulist mõtet, kuna niikuinii oli vajalik pinnahaldusega koos need üle testida.

Kui esimene liidese prototüüp oli valmis, mis importis ainult põhi andmed *connectivity*, siis tuli jooksvalt leida sobivaid aegu, millal oli võimalik testlugusid arutada koos pinnahaldusega. Koosolekuid õnnestus pidada kahel korral ning nende jooksul tõdesid arendajad, et importimise lahendus töötab.

Seejärel alustasid arendajad ülejäänud tootekaardi andmete lisamist liidesesse. Paraku selgus, et müügistatistika andmevälju on väga keerukas importida kuna *connectivity* on müügiinfot

kirjeldavad väljad on *specific* tüüpi. Seetõttu suudeti realiseerida ainult liides, mis impordib kogu tootekaardi infot kuid mitte müügistatistikat.

Kuna ETK-s oli selleks ajaks alustatud ühe infosüsteemi välja vahetamisega, mis oleks mõjutanud müügistatistika importimise protseduuri, siis otsustati et ei ole otstarbekas seda enam My.spaceman projekti raames juurde lisada, kuna paari kuu pärast oleks pidanud selle ümber tegema. Et aga selleks hetkeks oli juurutatud *my.spaceman* lahendus ning liidestatud ära toodete andmebaasiga, loeti My.spaceman projekt lõpetatuks ning otsustati, et alustatakse uus projekt, peale infosüsteemide vahetamist, mille eesmärgiks oleks toodete müügistatistika importimine *connectivity*.



## 4. Analüüs

### 4.1 Arutelu ja järeldused

My.spaceman projektis oli suurimaks probleemiks, et pinnahaldusel olid pooleli samaaegselt teised ärikriitilisemad projektid. Samuti pidid planogrammide koostajad tegema oma igapäevaseid töid ning seetõttu ei olnud võimalik tegelda lisaks My.spaceman projektiga. Agiilse metoodika rakendamiseks on vajalik äripoolega aktiivne suhtlemine, kuid antud projektis pidid arendajad sageli ootama pikalt äripoole tagasisidet. Näiteks nii *connectivity*, kui ka *my.spaceman* üle kontrollimise ja testimise pidid läbi viima planogrammide koostajad. Sageli olid aga pinnahalduse osakonna töötajad majast väljas teiste projektide tõttu ning seetõttu oli võimatu leida sobivat aega kohtumisteks, kus arendajad oleksid saanud koos pinnahaldusega testülesandeid läbi vaadata. Projekti jooksul õnnestus pinnahaldusel ning arendajate koosolekuid pidada väga ebaregulaarse režiimiga. Enamasti õnnestus koosolekuid testimise faasis läbi viia poolikute tsüklite kaupa. Koosolekud võisid toimuda kahel kolmel päeval järjest, mõnikord ka ülepäeviti, enne kui tekkis nädalane paus, mille jooksul pinnahaldus oli täielikult hõivatud teiste projektidega. Projekti lõpus läbiviidud vestluse käigus selgus, et kõik projektis osalejad pidasid pinnahalduse teisi projekte põhjuseks, miks My.spaceman projekt lõppes kaks kuud hiljem kui algselt oli planeeritud. Mitmete projektide kokkulangemine oli tingitud sellest, et juba varsemalt plaanitud ning alustatud projektid ületasid tähtajad oodatust rohkem.

Üheks võimalikuks lahenduseks sellises olukorras oleks olnud ehk kokku leppida regulaarsed kohtumised pikema ajaperioodi tagant, kasvõi kord nädalas fikseeritud kellaajal, kuhu kumbki osapool midagi muud ei planeeri. Äripoole ja arendajate aktiivse koostöö osakaalu vähendamine toimib aga teatud piirini, kuna agiilse metoodika juures on kliendi koostöö üks olulisi tegureid, mis tagab projekti edu. Seetõttu tuleks iga projekti jaoks leida aktiivse koostöö jaoks võimalused, mis tagavad koostöö stabiilsuse, kuid samas ei lahutaks äripoolt ja arendust liiga palju.

Kuigi projekti jooksul on võimalik vähendada probleemi mõju, oleks oluline ennetada probleeme, mis võivad tekkida agiilse metoodika rakendamisel. Ennetamiseks probleeme projektides kus

rakendatakse agiilset metoodikat, mis võivad tekkida äripoole vähesest aktiivsusest, tuleks hinnata enne projekti algust kui suurt mõju projektis avaldab äripoole osalemine. Kui projekti nõuded ja eesmärgid on väga konkreetselt ja üheselt paigas, siis ei pruugi äripoole aktiivne osalemine vajalik olla ning projekti läbiviimiseks oleks võimalik kasutada ka traditsioonilist arendusmetoodikat. Kui aga on teada, et nõudmised võivad projekti jooksul muutuda, peab äripool olema valmis aktiivselt osalema projektis. Kui see ei ole mingil põhjusel võimalik, siis tuleks analüüsida, millist mõju avaldab projekti tulemile kasutatav arendusmetoodika. Agiilset metoodikat rakendades on aga tunduvalt parem võimalus luua kvaliteetsem lahendus kui traditsiooniliste metoodikatega, kuid kui agiilse metoodikaga kaasnevate riskide, nagu äripoole osalus, maandamine ning projektist saadav lisaväärtus ei õigusta projektis tehtavaid kulutusi, tuleks rakendada traditsioonilist metoodikat.

Selleks, et otsustada kas on otstarbekas rakendada agiilset metoodikat projektis tuleks esitada järgnevad küsimused:

- 1) Kas, ning kui palju on äripool võimeline osalema aktiivselt projektis?
- 2) Kui tõenäoline on, et projekti jooksul muutuvad vajadused?
- 3) Kui palju lisaväärtust on võimalik saada muudatuste sisseviimisel?

Positiivseks teguriks, mis aitas projektil edasi liikuda, pidasid nii arendajad kui ka pinnahaldus sagedasi lühikesi koosolekud. Neid ei õnnestunud läbi viia regulaarsete intervallidega, sest pinnahalduse töörutiin oli segatud teiste projektide tõttu, kuid siiski toimusid koosolekud võimalikult tihti. Projekti erinevates etappides testimisel õnnestus koosolekuid viia läbi mõnikord kas iga päev või kahe päeva tagant. Samas oli testimisel ka pikemaid, kuni nädala pikkuseid perioode kus ei toimunud koosolekuid. Nii pinnahaldus kui arendajad leidsid, et sagedased lühikesed koosolekud olid väga efektiivsed, kuna sai kiiresti üle vaadatud, mis on tehtud, kas on probleeme ning ideid vahetada ja vastavalt kokku leppida järgmine samm. Arendajad leidsid, et koosolekutel osalemine andis väga palju motivatsiooni juurde. Koosolekutel oli võimalik paremini aru saada, mida äripool vajab ning samal ajal kaasa mõelda, kuidas oleks tehniliselt võimalik lahendus kõige paremini teostada. Käesoleva magistr töö autor leiab, et projekti My.spaceman jooksul läbi viidud koosolekud olid efektiivsed seetõttu, et need toimusid püstijalu. Autor täheldas, et püstijalu läbi viidud koosolekutel olid osalejad fokuseeritumad ning ei

kulutanud nii palju aega üldisteks aruteludeks, kui koosolekuteruumis istudes, vaid keskendusid konkreetsetele tegevustele või probleemidele. Sedasi oli osalejatel ka lihtsam ruumis ringi liikuda kasutada erinevaid töövahendeid nagu kirjutustahvel, et selgitada oma seisukohti või arvutit näitamaks milliste tegevuste peale tekib programmis veateade. Sellist koosolekute vormi samas ei ole hea kasutada pikkade koosolekute jaoks, kuna tõenäoliselt osalejad väsiks püstijalu seismisest ära.

ETK-s on juurutatud tugevalt RT (*Request Tracker*) keskkond, mida kasutatakse nii kasutajate probleemide haldamiseks kui ka IT osakonna sisemisteks ülesannete jagamiseks nõ. ülesande „piletitega“. My.spaceman projekti jooksul jagas IT projektijuht ülesandeid arendajatele mõnikord ka kasutades RT keskkonda, seda eriti projekti esimeses ja teises etapis, kui tegemist oli tehniliselt lihtsa ja ühekordse tegevusega, nagu andmebaasi ülesseadmine. Kuigi RT on vajalik keskkond IT probleemide ja tööde haldamiseks, siis My.spaceman projektis tekitas see mõningaid väiksemaid tagasilööke. Näiteks projekti esimeses etapis oli vaja üles seada *connectivity* andmebaas ning projektijuht edastas selleks vajalikud skriptid ning juhendid RT keskkonna vahendusel. Kuigi arendaja seadis üles andmebaasi edukalt vastavalt juhenditele otsustas ta *schema* nimetuse ära vahetada, mistõttu algul ebaõnnestus ODBC kaudu *spaceman* andmebaasiga ühendada. Hiljem selgus, et arendaja oli küll mõelnud kontrolliks üle küsida kas andmebaasi skeemi (*schema*) nimetus on oluline või mitte, kuid et RT keskkonnas pileтите kommenteerimine ja edasi tagasi suunamine võtab aega, siis leidis arendaja, et on lihtsam skeem nimetada ümber sedasi, et arendajal endal on seda hiljem lihtsam eristada ülejäänud skeemidest. Selline väike eksitus demonstreeris hästi seda, kui oluline on agiilses projektis suhtlemine ja info vahetamine näost näkku. Et RT on kasulik vahend tööülesannetel järke peal hoida, siis projekti kolmandas etapis rääkisid projektijuht ja arendaja enne täpselt läbi, mida on vaja teha ja kuidas vaja teha on ning alles siis avas projektijuht RT-s pileti.

My.spaceman projekti lõpus selgus magistr töö autorile projektis osalejatega vestluse käigus, et igapähe oli tekkinud omamoodi arusaamine agiilses meetodikast. Pinnhalduse osakonna töötajad kirjeldasid agiilset meetodikat, kui projektide läbi viimise vormi kus arendus toimub koostöös äripoolega. Arendajad pidasid agiilseks meetodikas oluliseks aga sagedasi iteratsioone. Kuigi My.spaceman projektis ei tekitanud probleeme meetodika põhimõtted vaid pinnahalduse vähene

aktiivsus, peab magistritöö autor oluliseks, et arendusmeeskonnal oleks üks ühtne arusaamine mida tehakse ja kuidas seda tehakse. Kui meeskonnaliikmetel tekivad erinevad arusaamised sellest kuidas projekte läbi viia või mis roll igal juhul projektis on, võivad osad meeskonnaliikmed hakata liikuma ülejäänud meeskonnaga vastupidises suunas. Agiilse meetodika põhimõtte järgi on äripool see kes annab pidevalt tagasisidet arendajale. Sedasi arendaja mõistab paremini äriprotsessi, mida loodav lahendus peab toetama ning seeläbi suudab luua vajaliku ning kasuliku tarkvara. Kui arendaja arvab aga et äripool peab tagasisidet andma selleks, et välja tuua arendaja enda vead ja puudused, võib arendaja kaotada projektis motivatsiooni. Seetõttu on oluline, et arendusmeeskond ainult ei järgi agiilse meetodika põhimõtteid vaid mõistab nende sisu ja eesmärki. Sellest tulenevalt leiab käesoleva magistritöö autor, et agiilsete meetodikate rakendamiseks peab arendusmeeskond mõistma, mis asi on agiilne meetodika ning mis selle eesmärk on. Seda oleks võimalik teha, läbi koolituste, kuna varasemad uuringud on ka näidanud, et koolituste läbiviimine mõjub positiivselt agiilse meetodika juurutamisele [5][16].

## 4.2 Soovitused

Magistritöös käsitletud projekti analüüsi põhjal annab autor järgnevalt soovitusi tulevaste projektide jaoks ETK-s, kus plaanitakse rakendada agiilset meetodikat.

- Kasutades projekti läbiviimiseks agiilset meetodikat, tuleks eelnevalt hinnata, kui palju lisaväärtust on võimalik saada agiilse meetodikaga ning kas see kaalub üle agiilse meetodika kasutamise puhul tekkivad ohud nagu äripoole ebapiisav osalus. Kui riskid lähevad liiga suureks, tuleks kaaluda kas traditsiooniliste meetoditega oleks võimalik saavutada parem tulemus, arvestades projekti tulemist saadavat lisaväärtust ning riskide suhet.

Selleks on vajalik, et IT projektijuht enne projekti algust kaardistab projektiga kaasnevad riskid ning riskide mõjud, et oleks võimalik analüüsida, kui palju suureneksid või väheneksid projekti riskid ning projekti tulemist saadav lisaväärtus kui agiilse meetodika asemel

kasutatakse traditsioonilist meetodikat. Kogutud info põhjal oleks võimalik koos äripoolega otsustada millist meetodikat oleks otstarbekas rakendada.

- Sage koosolekute läbiviimine projekti jooksul äripoole ja arendusmeeskonna vahel on väga oluline. Sedasi on võimalik tehtud tegevused ja tekkinud probleemid üle vaadata ning leppida kokku järgmised sammud. Arendajatel on kõige parem saada tagasisidet näost näkku kohtumistel, et oleks võimalik luua äripoole vajadustele vastav lahendus. Kui äripool ei anna arendajatele piisavalt sagedasti tagasisidet, ei pruugi funktsioonide puudujäägid või tekkinud vead välja tulla.

Maandamaks riske ETK-s, mis võivad kaasneda äripoole vähesest aktiivsusest, tuleks paika panna regulaarsed koosolekud, mis toimuvad kõigi projektis osalejate vahel olenevalt vajadustest projektis, näiteks igal nädalal. Kui projektis on aga arendajatel vaja sagedamini saada tagasisidet, tuleks projekti alguses määrata äripoolel isik kes vastutab arendusmeeskonnale tagasiside andmise eest. Nii oleks tagatud, et olukorras, kus arendajad ootavad tagasisidet loodud funktsioonide kohta, oleks äripoolelt olemas üks töötaja, kes tunneb äriprotsesse piisavalt põhjalikult ning vajadusel on võimeline täpsustama oma osakonna siseselt arendajate jaoks vajalikke detaile. Sedasi saaks garanteerida, et arendusmeeskonna ning äripoole vaheline infovahetus ei oleks puudlik ning arendajatel on võimalik stabiilselt saada tagasisidet kogu projekti vältel. Selleks on vajalik eeldus, et äripool soovib ning suudab enda töötajate hulgast leida vastava inimese. Samas ei tohiks selline suhtlusvorm asendada täielikult regulaarseid kohtumisi ülejäänud projekti kaasatud äripoole esindajate vahel.

- Projektis on RT keskkond projektijuhile vajalik abivahend, pidamaks järge tegevuste üle, mis meeskonnas hetkel pooleli on.

Kui projektis kasutatakse RT keskkonda ülesannete jagamiseks, on oluline need meeskonnas enne näost näkku läbi arutada, enne kui RT keskkonnas pilet luuakse. Sedasi saaks innustada meeskonnaliikmeid omavahel rohkem suhtlema ning soodustada iseorganiseeruva meeskonna

tekkimist, et vajadusel oleks lihtsam tööjaotust ümber korraldada. Samuti on siis väiksem tõenäosus, et ülesannet tõlgendatakse valesti.

- Agiilse metoodika rakendamisel on vajalik, et arendusmeeskond mõistab, mis on agiilse metoodika eesmärgid ning vahendid. Vastasel juhul ei pruugi meeskonnaliikmed täpselt aru saada, mis nende roll on. Selleks on vajalik koolituste läbi viimine kui tulevikus soovitakse jätkata agiilse metoodika rakendamist. Koolituste läbi viimiseks oleks otstarbekas leida väline partner kellel on kogemusi agiilse metoodikaga.

## Kokkuvõte

Tänapäeval on ettevõtetes äriprotsesside tagajateks sageli erinevad infosüsteemid. ETK arendusprojektide puhul aga ei ole osatud analüüsida keerukaid äriprotsesse ning vajadusi piisavalt detailselt, seetõttu on probleemiks, et paljud infosüsteemid ETK-s ei vasta kasutajate ootustele ning ei toeta äriprotsesse nõutud tasemel. Et tulevikus vältida äriprotsessidele mittevastavate süsteemide loomist, oleks üheks võimaluseks kasutada agiilset arendusmetoodikat. Agiilne metoodika on tarkvaraarenduse metoodika, kus äripool kaasatakse aktiivselt arendusprotsessi kogu projekti vältel. Tegemist on paindliku arendusmetoodikaga, kus selle asemel, et järgida fikseeritud nõudmisi, kohandatakse nõudmised vastavalt projekti jooksul tekkinud vajadustele.

Antud magistritöö eesmärgiks on määratleda agiilse arendusmetoodika rakendamisel esinevad probleemid ETK-s ning sellest lähtuvalt töötada välja soovitusi, mis aitaksid tulevikus parendada arendusprojektide läbiviimise protsessi.

Magistritöös käsitletud My.spaceman projekti eesmärgiks oli juurutada *my.spaceman* nimeline planogrammide haldamise lahendus ning liidestada see ETK infosüsteemidega. My.spaceman projekt viidi läbi kolmes etapis ning projekti jooksul lähtuti agiilse arendusmetoodika põhimõtetest: inimesed ja suhtlemine on tähtsamad kui protsessid ja tööriistad, töötav tarkvara põhjaliku dokumentatsiooni asemele, reageerimine muutustele plaani järgimise asemel.

Magistritöö tulemusena pakub autor järgnevad soovitusel agiilset metoodikat rakendavate projektide jaoks.

- Enne projekti alustamist tuleks IT projektijuhil kaardistada võimalikud riskid ning saada lisaväärtus, mis võiksid tekkida võrreldes traditsiooniliste meetodite kasutamisega. Kogutud info oleks aluseks otsusele, millist metoodikat kasutada.

- Soodustamaks suhtlust meeskonnaliikmete vahel ning vältimaks arusaamatusi, tuleb RT keskkonnas loodavad ülesanded meeskonnas enne omavahel läbi arutada.
- Maandamaks riski, mis kaasneb äripoolle vähesest aktiivsusest, tuleks projekti alguses äripoolle määrata vastutav isik, kes suhtleb arendusmeeskonnaga. Samas ei tohiks see asendada täielikult regulaarseid kohtumisi ülejäänud projekti kaasatud äripoolle esindajate vahel.
- Vajalik on koolituste läbiviimine välispartnerite poolt, et arendusmeeskond mõistaks agiilse metoodika eesmärke ning oskaks paremini kasutada selle vahendeid.



## Summary

### APPLYING AGILE METHODOLOGY. THE CASE OF MY.SPACEMAN PROJECT

Business processes in today's businesses are usually backed by information systems. However with ETK's development projects there hasn't been skills necessary to analyze complicated business processes and needs in necessary detail. For that reason many of the ETK's information systems have failed to meet the demands of users and therefore don't provide enough support for business processes. In order to avoid future mistakes by creating systems that don't accord with the business processes one of the solutions could be to use agile developing methodology. Agile methodology is a software development methodology where users are actively taking part in the development process throughout the entire project. It's a flexible development methodology in which instead of following fixed requirements it addresses to create requirements during the project's when new requirements and needs arise.

The goal of this Master's thesis is to pinpoint necessary factors for applying agile methodology in ETK's context and propose recommendations which would help carry out future projects based on agile methodology.

My.spaceman project's goal was to implement my.spaceman planogram managing solution and create link between it and ETK's information systems. The project was carried out in three stages and all the decisions stemmed from the principles of agile development methodology: people and communication are more important than processes and tools, working software instead of documentation, reacting to change rather than following plan.

As a result of Master's thesis, author offers the following recommendations for future projects using agile methodology:

- Before project begins IT project manager should map possible risks and profits that could emerge in comparison to using traditional methods. Gathered information would contribute to users decisions regarding which method to use.
- To promote communication between team members and to avoid misunderstandings, tasks set in Request Tracker (RT) should be discussed within team beforehand.
- To mitigate risks resulted from low user activity in project, users should point person who is responsible for communication with development team.
- Training is necessary to help development team to understand purpose of agile methodology and to have skills to use its tools. For that reason ETK should hire trainers who are skilled using agile methodology in software development process.

## Kasutatud kirjandus

[1] Agiilne arendus ja lepingud

URL: <http://www.usabilitykitchen.com/agiilne-arendus-ja-lepingud/> (28.04.2011)

[2] *Agile Introduction For Dummies.*

URL: <http://agileintro.wordpress.com/2008/01/04/waterfall-vs-agile-methodology/> (28.04.2011)

[3] *Agile and scrum.*

URL: <http://scrum.ee/en/node/13> (28.04.2011)

[4] *An Overview of Lean-Agile Methods.*

URL: <http://www.executivebrief.com/agile/overview-lean-agile-methods/> (28.04.2011)

[5] Chan, F. K.Y., Thong, J.Y.L. 2009. *Acceptance of agile methodologies: A critical review and conceptual framework. Decision Support Systems. Volume 46, Issue 4, 803-814. ISSN: 01679236*

URL: <http://www.sciencedirect.com/> (28.04.2011)

[6] Cockburn, A. *Agile Contracts*

URL: <http://alistair.cockburn.us/Agile+contracts> (28.04.2011)

[7] Conboy, K., Morgan, L. 2010. *Beyond the customer: Opening the agile systems development process. Information and Software Technology,*

URL: <http://www.sciencedirect.com/> (28.04.2011)

[8] Conboy, K., Coyle, S., Wang, X., Pikkarainen, M. 2010. *People Over Process: Key People Challenges in Agile Development.* IEEE Computer Society

URL: <http://www.computer.org/portal/web/csdl/doi/10.1109/MS.2010.132> (28.04.2011)

[9] ETK koduleht.

URL: <http://www2.etk.ee/et/ettevottest> (28.04.2011)

[10] *Extreme programming*

URL: <http://www.extremeprogramming.org/> (28.04.2011)

[11] Highsmith, J. 2010. *Agile Project management. Creating Innovative products, Second Edition. Addison-Wesley, Pearson education* ISBN-10: 0-321-65839-6

[12] Hoda, R., Noble, J., Marshall, S. 2010. *The impact of inadequate customer collaboration on self-organizing Agile teams. Information and Software Technology*

URL: <http://www.sciencedirect.com/> (28.04.2011)

[13] Iivari, J., Iivari, N. 2010. *The relationship between organizational culture and the deployment of agile methods. Information and Software Technology,*

URL: <http://www.sciencedirect.com/> (28.04.2011)

[14] Koch, A. 2004. *Agile Software Development : Evaluating the Methods for Your Organization. Artech House, Incorporated.* ISBN: 1-58053-842-8.

URL: <http://site.ebrary.com/lib/tallinn/Doc?id=10082017&ppg=31> (28.04.2011)

[15] Laanti, M., Salo, O., Abrahamsson, P. 2011. *Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. Information and Software Technology, Volume 53, Issue 3, 276-290.* ISSN: 09505849

URL: <http://www.sciencedirect.com/> (28.04.2011)

- [16] Misra, S.C., Kumar, V., Kumar, U. 2009. *Identifying some important success factors in adopting agile software development practices. Journal of Systems and Software, Vol. 82 Issue: Number 11. 1869-1890. ISSN: 01641212*  
URL: <http://www.sciencedirect.com/> (28.04.2011)
- [17] Moe, N.B., Dingsøyr, T., Dybå, T. 2010. *A teamwork model for understanding an agile team: A case study of a Scrum project. Information & Software Technology, Vol. 52 Issue 5, 480-491*  
ISSN: 0950-5849  
URL: <http://www.sciencedirect.com/> (28.04.2011)
- [18] Mugridge, R. 2008. *Managing Agile Project Requirements with Storytest-Driven Development IEEE Software, Vol. 25 Issue 1, 68-75. ISSN: 07407459*  
URL: <http://www.computer.org/portal/web/csdl> (28.04.2011)
- [19] Nerur, S., Mahapatra R., Mangalaraj, G. 2005. *Challenges of migrating to agile methodologies. Communications of the acm. Vol. 48, No. 5, 72-78.*  
URL: <http://www.sciencedirect.com/> (28.04.2011)
- [20] Normak, P. 2008. Projektijuhtimise konspekt. Tallinna Ülikooli õppematerjal
- [21] Organisatsiooni kultuur.  
URL: [http://en.wikipedia.org/wiki/Organizational\\_culture](http://en.wikipedia.org/wiki/Organizational_culture) (28.04.2011)
- [22] Piho, G. 2003. XP-metoodika juurutamisest väikeses eesti tarkvarafirmas. Magistritöö.
- [23] Projektijuhtimise käsiraamat. Juuni 2002. Äripäeva kirjastus
- [24] Qumer, A., Henderson-Sellers, B. 2008. *An evaluation of the degree of agility in six agile methods and its applicability for method engineering. Information and Software Technology, Volume 50, Issue 4, March 2008, 280-295. ISSN: 0950-5849*  
URL: <http://www.sciencedirect.com/> (28.04.2011)

[25] Schuh, P. 2004. *Integrating Agile Development in the Real World*. Charles River Media. ISBN: 9781584503644

URL: <http://site.ebrary.com/lib/tallinn/Doc?id=10078511&ppg=37> (28.04.2011)

[26] Setia, P., Sambamurthy, V., Closs, D.J. 2008. *Realizing business value of agile IT applications: antecedents in the supply chain networks*. *Information Technology and Management*, Vol. 9 Issue: Number 1, 5-19. ISSN: 1385951X; 15737667

URL: <http://www.springerlink.com/> (28.04.2011)

[27] Shojaee, H. *SCRUM in Under 10 Minutes*.

URL: [http://www.youtube.com/watch?v=Q5k7a9YEoUI&feature=player\\_embedded](http://www.youtube.com/watch?v=Q5k7a9YEoUI&feature=player_embedded) (28.04.2011)

[28] Tarbija24.

URL: <http://www.tarbija24.ee/?id=233122> (28.04.2011)

[29] Vlaanderen, K., Jansen, S., Brinkkemper, S., Jaspers, E. 2011. *The agile requirements refinery: Applying SCRUM principles to software product management*. *Information and Software Technology*, Vol. 53 Issue: Number 1, 58-70 ISSN: 09505849

URL: <http://www.sciencedirect.com/> (28.04.2011)

[30] *Waterfall model*.

URL: [http://en.wikipedia.org/wiki/Waterfall\\_model](http://en.wikipedia.org/wiki/Waterfall_model) (28.04.2011)

## **Lisad**

PÄHKLID, SOOLAPÄHKLID, SEEMNED				KUIVATATUD PUUVILJAD, ROSINAD				
47421500000 Kõrgend nõudlis 100g nää Näma	590057170003 Maapähklid rõõst. soolatud 50g Felix	47421500000 Pistaani soola 100g na0	47421500000 Pistaani soola 400g	47421500000 Teevilja segu 100g	47421500000 Keevika ja pärlilise segu 200g	47712495130 Nõudis puuviljasegu 200g Arimax	47712495133 Lõõnise 200g Arimax	474005300 Põhiline segu 150g na0
47421500000 Soolapähklid 100g	47421500000 Soolapähklid 100g	474005300342 Maapähkel soolatud Säätku 150g	5610361092114 Soolapähke tsallimaitse 150gTafel	4750195010549 Tudangise 150g Naka Naka	47400530034 Pähkide segu 100g	4740053004117 Pähkide ja puuvilja segu Säätku 150g		
47501950100 Kreem pähkel 50g Naka Naka	474005300388 Kreem pähkel 100g	4742150000002 Mandel magus CaliComis 100g	4742150000008 Pähkide segu 200g	474005300388 Ploom kuivat. 150g NAO	474005300065 Aprikoos kuivat. 150g NAO	474005300 Banaan kuivat. segu 100gna0		
474005300133 India pähkel 100g NAO	474005300102 Sarapuupähkel 100g NAO	474005300119 Mandel 100g NAO	474005300096 Kreem kuivat. 150g NAO	474005300333 Põhiline kuivatatud 100g	474005300 Puurilja Nix 150g NAO	Põhiline segu 100g		
4740053004193 Sarapuupähkel Säätku 100g	4740053004179 Mandel Säätku 150g	4740053004094 Maapähkel Säätku 150g	4740053004131 Ploom Säätku 150g	4740053004155 Aprikoos Säätku 150g				
474005300531 Maapähkel 100g	474005300571 Maapähkel 100g	474005300034 Maapähkel koostatud 150g NAO	474005300027 Maapähkel 150g NAO	4750195010525 Rosin Sultana 300g Naka Naka	474005300010 Rosin 150g NAO	4771249513015 Rosin Sultana 500g Arimax		
4742150000000 Põhiline seemned 135g	4740053000000 Põhiline seemned 100g	4740053000000 Põhiline seemned 100g	4740053000000 Põhiline seemned 100g	4750195010000 Seemne 150g Naka Naka	4750195010000 Seemne 150g Naka Naka	47712495126 Rosin Jumbo 200g Arimax	4740053004070 Rosin Säätku 150g	4742150001094 Rosin Sultana 200g
4740053000007 Põhiline seemned 50g	4740053000072 Põhiline seemned puhast. 150g NAO	4770286126450 Põhiline seemned trübitseeritud 50g Y.K.S.	4770286126268 Põhiline seemned maitse 170g Y.K.S.		4770286126122 Põhiline seemned trübitseeritud 150g Y.K.S.			

Lisa 1- Planogramm, mis on koostatud kasutades *spaceman*