

TARTU ÜLIKOOL  
Pärnu kolledž  
Ettevõtluse osakond

Marco Peedo  
EP3

**PAINDLIKE MEETODITE RAKENDAMINE  
VÄIKESEMAHULISTE  
TARKVARAPROJEKTIDE PUHUL**

Lõputöö

Juhendaja: dotsent Arvi Kuura

Pärnu 2014

# SISUKORD

Sissejuhatus .....	3
1. Paindlikud projektijuhtimismeetodid .....	5
1.1. Väikesemahulise tarkvaraprojekti määratlus .....	5
1.2. Paindlike projektijuhtimismeetodite võrdlus traditsiooniliste lähenemistega.....	8
1.3. Ülevaade paindlikest meetoditest.....	11
1.3.1. Kristallmeetodid.....	11
1.3.2. Dünaamiline tarkvaraarendusmeetod.....	13
1.3.3. Omaduspõhine tarkvaraarendusmeetod .....	16
1.3.4. Scrum meetod .....	18
1.3.4. Ekstreemprogrammeerimine .....	19
1.4. Paindlike meetodite võrdlus.....	21
2. Paindlike meetodite Võrdlus näidisprojekti abil .....	24
2.1. Metoodiline alus.....	24
2.2. Tegevusuuringu läbiviimine ja uuritava objekti kirjeldus .....	28
2.3. Paindlike meetodite rakendamise planeerimine.....	33
2.3. Projekti elluviimise tegevusuuring.....	39
2.4. Uuringu järeldused ja soovitused .....	48
Kokkuvõte .....	55
Viidatud allikad.....	58
Lisad.....	62
Lisa 1. Pärnu töökeskkonda iseloomustav probleemipuu .....	62
Lisa 2. Projektipäevik.....	63
Lisa 3. SA Ida- Virumaa Tööstusalade Arenduse ankeet ja selle kokkuvõte .....	67
Lisa 4. Personaliotsingu andmebaasi esimene prototüüp.....	70
Lisa 5. Erinevate töövahendusportaalide ülevaade ja võrdlus .....	75
Lisa 6. Personaliotsingu andmebaasi lõppversioon .....	76
Summary .....	81

## SISSEJUHATUS

Paindlikke projektijuhtimismeetodeid peetakse tarkvaraga seotud projektide teostamisel tihti sobivaimaks valikuks. Seoses nüüdisajale iseloomuliku tehnika arenguga suureneb ka pidevalt tarkvaraliste projektide osatähtsus ja muutuvad seonduvad nõuded. Klassikalist lähenemist seotakse põhjaliku planeerimisega ning on paremini rakendatav suhteliselt muutumatus keskkonnas. Sellest tulenevalt leiavad üha rohkem tarkvaraloojaid, et klassikaliste projektijuhtimise meetodite asemel on kasulik rakendada paindlikke ehk väledaid meetodeid.

Lõputöö eesmärgiks on katsetada erinevaid meetodeid väikesemahuliste tarkvaraprojektide juhtimiseks ning töötada välja juhised sarnaste projektide läbiviimiseks. Töö eesmärgist lähtuvalt saab välja tuua järgmised uurimisülesanded:

- leida tunnused, mis iseloomustavad väikesemahulisi tarkvaraprojekte;
- analüüsida võrdlevalt paindlikke ja traditsioonilisi meetodeid;
- kirjanduse baasil analüüsida võrdlevalt erinevate paindlike meetodite põhimõtteid ja rakendusviise;
- testida erinevate meetodite kasutatavust näidisprojekti abil;
- analüüsida läbiviidud näidisprojekti ja hinnata kasutatud meetodite sobilikkust;
- töötada välja juhiseid sarnaste väikesemahuliste tarkvaraprojektide läbiviimiseks.

Teoreetilise tausta loomiseks kasutatakse projektijuhtimist, eeskätt rakendatavaid meetodeid kajastavaid raamatuid, artikleid ja internetiallikaid. Erinevad teoreetilised käsitlused annavad ülevaate väljakujunenud lähenemiste erisustest, sarnasustest, põhimõtetest ning rakendusviisidest,

Empiiriliste uurimisülesannete teostamiseks viiakse läbi konkreetse juhtumi süvaanalüüs, kus vaadeldakse näidisprojektis kasutatavaid meetodeid. Vaatluse alla seatakse näidisprojekt, milleks on töövahendusportaali loomine Pärnumaal. Projekt on väikesemahuline ning olemuselt sõltuv tarkvaralisest lahendusest ja selle läbiviimisel katsetatakse erinevaid meetodeid, mis paindlike meetodite pioneerid loonud on. Läbi juhtumiuuringu hinnatakse tehtud otsuseid, selgitatakse nende tulemusi ja kaalutakse alternatiive. Tehtud uuringu põhjal tehakse järeldused erinevate meetodite rakendatavuse kohta väikesemahulistest tarkvaraprojektides ning antakse juhised sarnase ulatusega projektide elluviimiseks.

Töö koosneb kahest peatükist, millest esimene keskendub teoreetilise tagapõhja loomisele. Defineeritakse väikesemahulised tarkvaraprojektid ja analüüsitakse paindlike ehk väledate meetodite erinevusi traditsiooniliste suhtes. Analüüsivalt võrreldakse ka erinevate paindlike meetodite põhimõtteid ja rakendusviise. Teine peatükk on suunatud erinevate väledate meetodite katsetamisele, kus hinnatakse teoorias uuritud lähenemisviise. Seejärel analüüsitakse tehtud otsuste tulemusi ning antakse hinnang tehtud projektiosadele, mille põhjal tehakse järeldused erinevate meetodite kasutamise kohta.

# 1. PAINDLIKUD PROJEKTJUHTIMISMEETODID

## 1.1. Väikesemahulise tarkvaraprojekti määratlus

Tarkvaraprojektid on projektijuhtimise üks paljudest liikidest. Konkreetse projektiliigi juhtimisviiside väljaselgitamiseks tuleb selgitada nende olemus ja ja nende toimimispõhimõtted. Väikesemahuliste tarkvaraprojektide teostamisel eelistatakse meetodeid, mis seostatakse paindlikkusega. Edaspidi kasutatakse töös nende kirjeldamiseks mõisteid nagu paindlikud või väledad (*agile*) meetodid. Järgnevalt määratletakse väikesemahulise tarkvaraprojekti olemus läbi definitsioonide ning teoreetilise lähenemise.

Tarkvara omab arvutiajastul keskset rolli kõikides igapäevaelu valdkondades ja selle usaldusväärsusest sõltub paljude inimeste igapäevaelu. Selle tulemusena kulutatakse tarkvaraarendusele üha enam aega ja raha, selle tähtsus kasvab ning suureneb erinevate arendustegevuste ulatus. (Software... 2004 :6-8) Tarkvaraprojekti olemust ning ideed aitab paremini mõista projekti definitsioon. Järgnev tabel 1 toob ülevaate projekti erinevate mõisteselgituste kohta.

**Tabel 1.** Mõiste 'projekt' erinevad tähendused (CDO 2014; EKSS 2014; project 2014; Artto jt 2011: 17; Turner, Müller 2003; autori koostatud)

Allikas	Seletus
Artto, Martinsuo, Kujala	Projekt on unikaalne üksus, mis on moodustatud keerukate ja omavahel seotud tegevustega, millel on eelnevalt määratletud eesmärgid, mis peavad olema täidetud kindla aja, eelarve ning spetsifikatsiooni piires.
Cambridge Dictionaries Online	Osa planeeritud tööst või tegevus, mis on teostatud teatud aja jooksul ning mõeldud saavutama kindlat eesmärki.
Eesti keele seletav sõnaraamat	Plaan, kavand.
Eesti keele seletav sõnaraamat	Tehniliste dokumentide komplekt, mis sisaldab ehitatava, remonditava, rekonstrueeritava v. restaureeritava objekti kirjelduse, joonised, põhjendavad arvestused ja vahel ka maketi.

Allikas	Seletus
Oxford English Dictionary	Plaan, kavand, skeem, disain või muster, mille järgi midagi teostatakse.
Oxford English Dictionary	Planeeritud või kavandatud ettevõtmine, skeem, ettepanek, eesmärk, ülesanne.
Turner, Müller (2003)	Projekt on ajutine organisatsioon, mille ressurssidega saavutatakse unikaalne, muutust kaasatav eesmärk.

Läbivaks ideeks tabelis toodud mõisteselgituste puhul on plaan, kavand ning tegevus mingi eesmärgi saavutamiseks. Oluliseks kriteeriumiks on aga ka piiratud aeg, mille jooksul määratletud tulemuste saavutamiseks. Projekt on defineeritav ka kui ajutine organisatsioon (nt Turner, Müller 2003). Sellisele lähenemisele panid suuresti aluse Lundin ja Söderholm oma 1995 aasta artikliga, mille järgi keskendub ajutine organisatsioon tegevusele ja lähtub selle teostaja ressurssidest. Artto jt (2011: 16) järgi on ajutine organisatsioon projektiorganisatsioon, mis luuakse kindla ülesande täitmiseks.

Artto jt (2011: 17) sõnul on nende kajastatud definitsioon projektist, kui unikaalsest üksusest laialdaselt aktsepteeritud. Kuna see on ka projekti eri aspekte kõige konkreetsemalt kokkuvõttev, lähtub töö autor sellest tähendusest. Samas suhtutakse projekti ka kui seda teostatava organisatsiooni poolt sõltuvasse ajutisse organisatsiooni.

Selleks, et paremini mõista tarkvara arendusprojekti, tuleb uurida seda iseloomustavaid tegureid. Järgnev tabel annab ülevaate tarkvaraarendusega seonduvast terminoloogiast koos nende tähendustega.

**Tabel 2.** Tarkvaraarendusega seonduvad tähendused (tekstis viidatud allikate põhjal; autori koostatud)

Mõiste	Selgitus
tarkvaraarendus	Professionaalne süsteemi tarkvara arendamine, tootmine ja haldamine (software engineering 2014).
tarkvaraarendus	Sellist tüüpi arendus, mis allub arvutiteaduse ja matemaatika põhimõtetele, et saavutada kuluefektiivne lahendus tarkvaralistele probleemidele (Ford 1990: 6).
tarkvaraarendus	Süsteemiarenduse allharu, mis tegeleb suurte ja keerukate tarkvaranõudlike süsteemidega (Finkelstein, Kramer 2000: 3).
tarkvaraarenduse meetod	Süstemaatiline protseduur või tehnika tegemaks tarkvaraarendust, mille eesmärgiks on jõuda kindla eesmärgini ja/või jõuda määratletud tarkvara osade kogumikuni. (Engels, Sauer 2010: 414)

Mõiste	Selgitus
tarkvaraarenduse protsess	Protsess, läbi mille kasuta ja vajadused muudetakse tarkvaratooteks (IEE Standard 610.12 1990 :67).
tarkvaraarenduse protsess	Inseneriteaduse stiilis eesmärgipõhine tarkvara arendamine (Münch jt 2012: 8).

Autor leiab, et nende mõistete kombineerimisel on tarkvara arendusprojekt defineeritav kui tegevus, mis on teostatud kindla aja jooksul, kasutab tarkvaralisi lahendusi ning on mõeldud lahendama kindlat tarkvaralist probleemi.

Projekti juhtimiseks parimate ja kõige sobilikumate vahenditega on oluline teada selle suurust ning ulatust. Projekti väikesemahulisuse piiritlemiseks on Rowe (2007: 5-8) toonud välja põhitunnused, mis on tavapärased just väikeste projektide puhul. Kõik väikesed projektid on idee poolest lihtsad, kuid ühene viis määramaks, kas projekt on väike või mitte, puudub. Siiski on autor välja selgitanud mitmeid tunnuseid, mis on omased väikesemahulistele projektidele. Olulisimad kriteeriumid on esitletud tabelis 3.

**Tabel 3.** Väikesemahulise projekti tunnused (Rowe 2007: 5; autori koostatud)

Tunnus	Kriteerium
Kestvus	Lühiajaline, tüüpiliselt kestab kuni kuus kuud .
Tööaeg	Peamiselt osaline tööaeg.
Meeskonna suurus	Kümme või vähem liiget.
Nõutav oskuste hulk	Vaja on suhteliselt vähe erinevaid oskusi.
Eesmärk	Üksikeesmärk ja lahendus.
Maksumus	Kuni 55 000 eurot ning vajaminevad vahendid projekti teostajal olemas
Tulemused	Tekivad üheselt mõistetavad tulemused, mis sõltuvad erinevatest oskusaladest vähesel määral.

Lin ja Liu (2011: 1355) leiavad, et väikeprojekti definitsioon sõltub konkreetse isiku vaatepunktist ning ettevõtte varadest, mille kontekstis projekti teostatakse. Kuura jt (2013: 219) on leidnud, et projektijuhtimine on suures osas võrreldav ettevõtlusega ning valdkondade vahel kehtivad mitmed sarnasused. Projekti käsitletakse ka kui ajutist organisatsiooni (vt ka tabel 1 lk 5), mis sõltub ressursside poolest selle ellu kutsunud alalisest organisatsioonist. Hinnang, kas projekt on väikesemahuline või mitte sõltub seega seda teostava organisatsiooni suurusest.

Euroopa komisjoni järgi on väikeettevõtte organisatsioon, kus töötab vähem kui 50 inimest ning mille käive või bilansimaht on kuni kümme miljonit eurot. Mikroettevõttes on aga töötajaid alla kümne ning käive või bilansimaht ei ületa kahe miljoni euro piiri (Commission... 2003)

Töö autor järeldeb, et kuna projekti suuruse määratlemine sõltub kontekstist, on selle hinnangud erinevad. Arvestada tuleb projekti investeeringu suurusega, projektimeeskonna arvukuse ja projektipõhiste tegevuse osakaaluga teiste tegevuste suhtes. Näiteks kui projekti teostab väikeettevõtte, kus on alaliselt tööl 45 inimest ning teostamiseks läheb vaja kuut inimest kolme kuu jooksul, on see pigem väikeprojekt. Sama projekt mikroettevõttes aga enam väikeprojektina ei liigitu. Kindlasti tuleb arvestada ka teisi projektiks vajaminevaid ressursse ja võrrelda neid olemasolevate vahenditega. Väikesemahuline tarkvaraprojekt on seega suhteliselt väheste vahenditega teostatav eesmärgistatud tegevuste kogum, mis kasutab tarkvaralisi võimalusi konkreetse probleemi lahendamiseks.

## **1.2. Paindlike projektijuhtimismeetodite võrdlus traditsiooniliste lähenemistega**

Traditsiooniline projektijuhtimine on läbi aegade olnud tavapärane viis uuenduslike ideede elluviimisel ning see on suuresti eelistatud ka tänapäeval. Üha kiirenev informatsiooni liikumine toob aga kaasa vajaduse olla paindlik ning avatud erinevatele muudatustele, mis projekti elluviimise eri etappides muutuda võivad. Kui projekti algaasis oodatakse ning planeeritakse teatud kindlat tulemit, siis aja jooksul võivad nendes ootustes toimida suured muutused. Selline käitumine on eriti tõenäoline tarkvara valdkonnas, kus tehnoloogia arengud ning uuenduslikud ideed võivad iga hetk muuta projekti tellijate soovi. Eelmainitud muutustega toimetulemiseks ongi välja töötatud paindlikud projektijuhtimismeetodid.

Stoica jt (2013: 65-70) arutlevad, kuidas suurenenud paindlikkus annab kiire ja tõhusa kohanemise erinevate turumuutustega ning on erinevate organisatsioonide jaoks üha suureneva tähtsusega. Lisaks toob paindlik ehk kohanemisvõimeline projektijuhtimine kaasa väiksema ajakulu uute protsesside loomises ning olemasolevate muutmises, kus



modifitseerimis- ja täiendusvõimalus on edukuse eelduseks. Keerukas ja pidevalt muutuv keskkonnas ei ole organisatsiooniline paindlikkus enam lihtsalt vajadus vaid üheks tingimuseks konkurentsipüsimiseks. Kohanemisvõimeline ettevõtte suudab kiirelt reageerida klientide nõudmistele ning keskkonna muutustele, mis suurendab võimalust saavutada turueelist. Järgnev tabel 4 annab ülevaatliku võrdluse traditsiooniliste ning paindlike meetodite eelistest ja puudustest.

**Tabel 4.** Traditsiooniliste ja paindlike meetodite võrdlus (Stoica jt. 2013: 73; Dybå, Dingsøy 2008: 834-837; Manifesto... 2014; autori koostatud)

Meetodi liik	Traditsioonilised meetodid	Paindlikud meetodid
Suhtlus	ametlik	mitteametlik
Organisatsiooniline korraldus	mehaaniline, bürokraatlik	orgaaniline, paindlik
Kliendi vajadused	fikseeritud ja defineeritud enne teostamist	interaktiivne ja pidevalt täienev
Kvaliteedi tagamine	range kontroll ja keerukas planeerimine	pidev nõuete, disaini ja lahenduste kontroll
Katsetamine	pärast koodi kirjutamist	igal vaheetapil
Kliendi kaasatus projekti kestel	kaasatakse vähesel määral	kaasatakse võimalikult palju
Suurus	suured meeskonnad ja projektid	väikesed meeskonnad ja projektid
Põhiline väärtushinnang	maksimaalselt määratletud probleemistik, kõikehõlmav planeerimine, dokumenteerimine	inimesed ja suhtlemine ning toimiv toode on olulisemad kui dokumenteerimine
Muudatuste elluviimine	raske, muudatused ei ole eeldatud ega soovitud	süsteem eeldab muudatusi ning võimaldab nende elluviimist

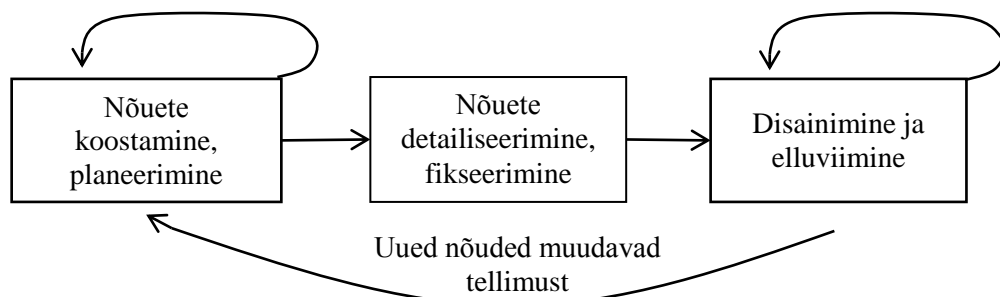
Traditsioonilised meetodid on loomult ranged ja eeldavad struktuurset lähenemist. Kliendi vajadused ning projekti teostamiseks vajalikud tegevused tuleb planeerida ning fikseerida enne selle algatamist. Paindlikud lähenemised ei eelda aga detailset planeerimist ning võimaldavad loomingulisemat lähenemist planeerimises, katsetamises ning testimises. Kohanemisvõimelised meetodid võimaldavad projekti pidevat täiendamist ning dünaamilisemat suhtlust nii organisatsiooni sees kui ka klientidega.

Paindlikud meetodid on osutunud väga tõhusaks tarkvaratoodete arendustes, kus tarkvaratööstus arendab väikest või keskmise suurusega toodet. Kasulikud on need ka siis, kui on tegemist süsteemi arendamisega organisatsiooni sees, kus projekti tellijal on

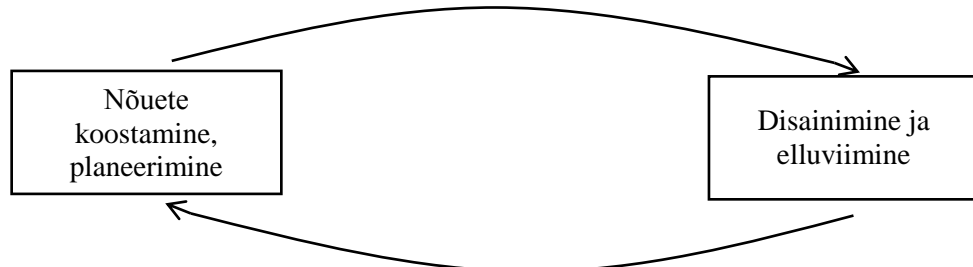
selge soov olla seotud arendustegevusega ning kus ei ole palju väliseid reegleid ja regulatsioone, mis lõpplahendust mõjutada võivad.. (Sommerville 2011:59).

Nimetatud projektitüübid võib liigitada peatükis 1.1 määratletud väikeprojektide hulka, kui need hõlmavad väikese suurusega toodete loomist ja puuduvad suurorganisatsioonile omased regulatsioonid. Väledad meetodid on head vahendid väikeprojektide elluviimiseks kuna soodustavad kliendi kaasamist ja vajalike muudatuste tegemist. Järgnev joonis 1 annab ülevaate plaanipõhise ja paindliku projektijuhtimismeetodi protsesside erinevustest.

Planeerimis põhine arendus



Paindlik arendus



**Joonis 1.** Plaanipõhise ja paindliku arenduse kirjeldus (Sommerville 2011: 63)

Plaanipõhine tarkvaraarenduse protsess toetab järk-järgulist arendamist, kus iga iteratsioon koosneb planeerimisest, detailiseerimisest ja elluviimisest. Arendustegevus jaotatakse osadesse ning igal astmel luuakse eelnevalt valminud osale mingisugune lisandväärtus. Paindlik protsess ei pööra nii suurt tähelepanu dokumentatsioonile vaid on rohkem erinevate versioonide loomisele ning pidevale täiendamisele. (Sommerville 2011: 63-64)

Paindlikel meetoditel on mitmeid olulisi erinevusi võrreldes traditsiooniliste lähenemistega. Olulisemad põhimõtted on avatus muutustele, inimestele ja suhetele

orienteeritus ning keskendumine kliendi muutlikele soovidele. Klassikalised meetodid eeldavad detailset ja struktureeritud planeerimist ning etapiviisilist teostust, mis peab viima algselt määratletud eesmärkide täitmiseni. Väledad meetodid sobivad eelkõige väikesemahuliste tarkvaraprojektide jaoks, mille meeskond on väike ning oodatavad tulemused võivad aja jooksul muutuda.

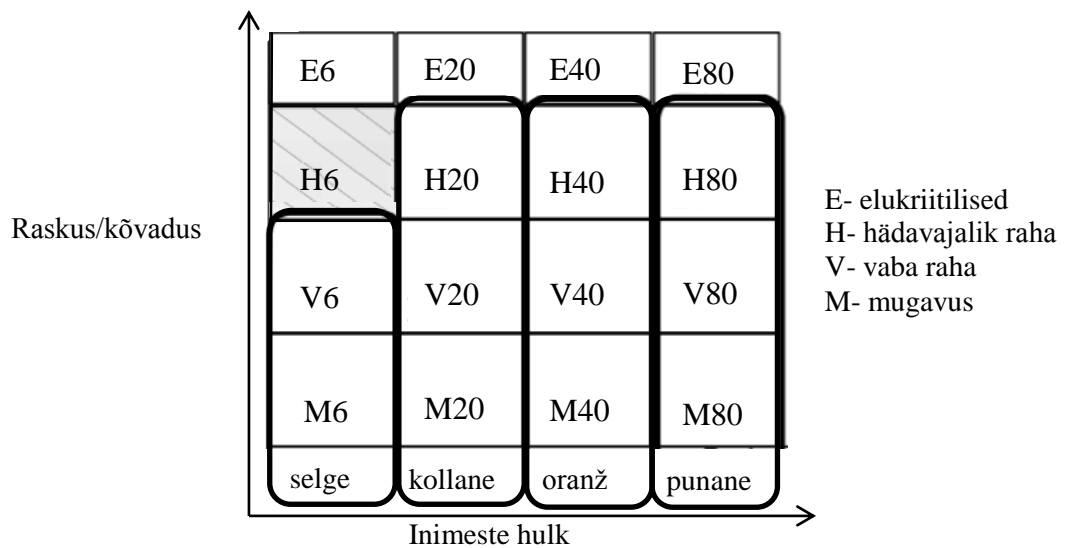
Paindlike meetodite hulk on mitmekesine ning tihti kasutatakse nende erinevaid kombinatsioone. Kuigi kõigil neist on unikaalseid vaateid ja lähenemisi, lähtuvad nad siiski paindliku projektijuhtimise manifestist. Meetodeid kasutatakse tarkvaraarenduse kvaliteedi tõstmiseks läbi tõhusa suhtlemise, testimise ja erinevate töökeskkonna elementide integreerimise. (Stoica jt 2013) Kõige enam viidatud meetodid on kristallmeetodid, dünaamiline tarkvaraarenduse meetod, omaduspõhine tarkvaraarendus, *Scrum* ja ekstreemprogrammeerimine (Dybå, Dingsøy 2008: 835; Stoica jt 2013: 73). Järgnevas alapeatükis selgitatakse mainitud meetodite ülesehitust ja põhimõtteid. Selgitatakse välja erinevate meetodite ühised jooned ning nende rakendusviisid.

## **1.3. Ülevaade paindlikest meetoditest**

### **1.3.1. Kristallmeetodid**

Tarkvaraarenduses on paindlikkus ja kiire otsustusvõime oluline kuna võimaldab kliendikeskset lähenemist ja optimeeritud arendustegevust. Nüüdisaja kiire infovahetus ja pidevalt muutuvad soovid ning võimalikud lahendused ei võimalda tarkvaraprojekte pikalt ja fikseeritult planeerida. Esimene vaadeldav väle meetod pakub tarkvaraprojektide teostamiseks mõtteviise ja suhtumist. Nende rakendamisel on projekti-meeskonnal võimalik välja kujundada oma metoodikad ja süsteemid eri suuruse ja keerukusega projektide teostamiseks.

Cockburn (2006: 335-366) on kristallmeetodite pioneer ning tema sõnul on see lähenemisviis, mis sobib erineva suurusega meeskondade jaoks. Organisatsioon saab ise kujundada oma projektijuhtimise meetodid vastavalt projektide keerukusele ja suurusele, lähtudes seejuures oma tavadest. Järgnev joonis annab ülevaate kristallmeetodite ülesehitusest.



**Joonis 2.** Kristallmeetodite ülesehitus projekti keerukuse järgi (Cockburn 2006: 336)

Kristallmeetodid on jaotatud valgusomaduste kaupa: selge, kollane, oranž, ja punane. Mida tumedamaks valgusspektris liigutakse, seda rohkem on seonduvaid inimesi. Mida ülespoole, seda elutähtsam on tegevus. Organisatsiooni jaoks väiksed ja kerged projektid on seega kristallselged, mis keskenduvad suhtlusele väikeste meeskondade vahel, kes töötavad välja mittekriitilist tarkvara. Kõige keerukamad meetodid on punased ning neilt eeldatakse rangemat ja traditsioonilisemat lähenemist. Süsteemi rakendamisel on oluline riskide maandamine ja tõhusa arendustegevuse säilitamine. (Cockburn 2006: 336-338)

Antud struktuuri järgides on projekti teostajatel võimalik selgete kriteeriumite alusel eristada projektitüüp ning vastavalt sellele ka sobivat lähenemist rakendada. Meetod on seega kasulik organisatsioonides, kus esineb mitmeid erineva suurusega ja erineva tevaldkondade projekte, kuna võimaldab iga alaliigi jaoks rakendada ühtset ja kindlat lähenemisviisi.

Järgnev tabel 5 annab ülevaate kristallmeetodi seitsme põhiomaduse kohta, mis iseloomustavad edukaid tarkvaraprojekte ning selgitab nende tähendust.

**Tabel 5.** Kristallmeetodi seitse põhimõtet selgitustega (Cockburn 2004: 33-60; autori koostatud)

Põhimõte	Selgitus
Sagedane väljastamine	Eeldab iteratiivset lähenemist, kus seni valminud lahendust esitletakse kliendile iga teatud aja tagant- sisuliselt tegemist testimisega.
Peegeldav täiendamine	Tegemist meeskonnasisese suhtlemisega, mille käigus saadakse ülevaade asjadest mis töötab, mis ei tööta ja mida parendada.
Osmootne suhtlus	Ideeks taustamüra ärakasutamine, kus inimesed paigutatakse töökohas selliselt, et informatsioonivahetus ühe inimesega leviks ka teisteni.
Isiklik turvalisus	Inimestevaheline usaldus rääkida muredest ja häirivatest asjaoludest, ilma et peaks kartma vaenulikku tagasisidet.
Fookus	Projektiliikmete teadlikkus vajalikest töödest ning piisav vaba aeg nende teostamiseks.
Oskuskasutajate kaasamine	Eeldab testimist oskuslike ja asjast huvitatud isikutega selgitamiseks välja puudused, soovid ja võimalused.
Tehnilised põhiomadused	Nõuab, et tagatud oleks automaatne ja automatiseeritud testimine, varasemate projektiosade taastamisvõimalus ning pidev integreerimine tervikuga.

Selgitav tabel kristallmeetodi omadustest annab ülevaate ka sellest, kuidas neid põhimõtteid rakendada projekti läbiviimisel. Edukas tarkvaraprojekt on selline, mida pidevalt testitakse ja kontrollitakse. Meeskonnasisene suhtlemine peab olema konstruktiivne ning vastastikuliselt täiendav, mis tähendab, et informatsioon, mis jõuab ühe meeskonnaliikmeni saab vabalt edasi liikuda ka teistele. Oluline on, et projektiliikmed saaksid üksteist usaldada ja vabalt mõtteid vahetada. Samuti peab kõigil olema teada vajalikud tööülesanded koos vajalike vahenditega. Valmiva lahenduse testimine eeldab katsetamist päriskasutajate peal ja võimalust kontrollida koodi toimimist. Vajalikuks omaduseks on ka pidev võimalus integreerida valminud osa terviklahendusega ja probleemide tekkimisel taastada viimast versiooni.

Nende põhimõtete rakendamisel tagatakse avameelne, kuid samas turvaline lähenemine projekti valmimisele. Soovitud lahenduse valmimisele on kaasatud lisaks tellijale ka lõppkasutajad ja see annab meeskonnale kindlustunne, et liigutakse õiges suunas. Mida rohkem neist seitsmest omadusest esindatud on, seda suurema tõenäosusega on tagatud ka projekti edukus.

### 1.3.2. Dünaamiline tarkvaraarendusmeetod

Tarkvaraprojekti teostamisel on tavapärane, et projekti tahetakse teostada võimalikult kiiresti ja olla vastuvõtlik kliendipoolsetele ettepanekutele. Samas võib tekkida olukord,

kus projekt on loomult niivõrd mahukas ja riskantne, et projektijuht soovib suuremat kontrolli. Riskide maandamine tähendab meetodi kasutamist, mis soodustab põhjalikumalt planeerimist ning väiksemat järeleandlikkust muutustele, kuid samas annab juhised, mis võimaldavad keskenduda kõige tähtsamale.

Dünaamilise tarkvaraarendusmeetodi puhul ei ole tegemist niivõrd meetodi kui raamistikuga. Põhimõtete kirjeldamisel on lähtutud üldistest ideedest, mistõttu on see rakendatav erinevates keskkondades. Meetod pakub soovituslikud teeviidad nii struktuuri- kui ka eesmärgipõhise lähenemise jaoks. Suurim erinevus koskmeetodist tuleneb selle tsüklite kiirusest, kus soovitud funktsionaalsuseni tuleb jõuda minimaalse ajaga (mõni päev kuni mõni nädal) sõltuvalt projektist. Tõhusus tuleneb asjaoludest, et raamistik näeb ette pidevat kontrollimist ja probleemide lahendamist ning keskendutakse vähem briifingutele. (Cockburn 2003) Järgnev tabel annab ülevaate nimetatud meetodi üheksast juhtprintsibiist.

**Tabel 6.** Dünaamilise tarkvaraarenduse üheksa juhtprintsipi (Stapleton 2003: 13-22; autori koostatud).

Põhimõte	Selgitus
Aktiivne kasutajaseotus on hädavajalik.	Kasutaja hinnang olukorrast vähendab vigade tekkimist ning hoiab arendust järjepidevana.
Meeskonnad peavad olema volitatud otsuste tegemiseks.	Pidev vertikaalne suhtlemine iga väiksema ressursi taotlemiseks tõstab projekti aja- ja rahakulu oluliselt.
Keskendumine sagedasele väljastamisele.	Sage väljastamine tagab kiirema veatuvastuse, mis teeb nende parandamise lihtsamaks.
Äriline sobivus on aktsepteeritavuse kriteeriumiks.	Esmane eesmärk on tagada põhinõuded ning jätta täiendavad tegevused hilisemateks sammudeks.
Astmeline ja tsükliline arendus on kohustuslik.	Väikesammuline arendustegevus aitab vähendada projekti keerukust ja reageerimisaega muutustele.
Muutused peavad olema tagasipööratavad.	Avatus muutustele eeldab võimalust jätkata arendust varasematest tsüklitest.
Põhinõuded tuleb fikseerida enne projekti algust.	Kokkulepitud põhinõuded annavad ettekujutuse ulatusest ja väldivad ootamatuid suuri muutusi projektis.
Katsetamine peab toimuma läbi terve elutsükli.	Sage ja varajane testimine alates arendustegevuse algusest kindlustab õiget lähenemist.
Suhtlemine lähtub koostööst.	Tehniliste ja juhtivate töötajate vaheline koostöö võimaldab eesmärgipõhist lähenemiste ning objektiivset tagasisidet.

Antud meetodi puhul on tegemist tsüklilise ja pidevalt täieneva arendustegevusega. Oma põhimõtetest asub see paindlike ja traditsiooniliste meetodite vahepeal kuna

eeldab küllalt põhjalikku planeerimist ja fikseeritud lähenemist, olles suhtlemiselt ja arendustegevuselt siiski paindlik (vt ka tabel 3 lk 7).

Oluliseks väärtuseks hinnatakse kasutajate arvamust ja suhtumist. Pidev kontroll aitab ära hoida vigade tekkimist ning informeeritud valikute tegemist. Selle teostamiseks tuleb võimalikult tihti väljastada ja proovida uusi, täiendatud versioone. Protsesside optimeerimiseks peetakse vajalikuks otsustusõiguse andmist võimalikult madalale tasemele. Iga elementaarse ressursi või tehingu tegemiseks tehtav vertikaalne suhtlemine kulutab väärtuslikku aega ja vähendab muutuste paindlikkust. Erinevad modifikatsioonid projektis võivad osutada aga oluliseks ning nende vajalikkuse avastamiseks tuleb arendustsüklid teha võimalikult lühiajalised. Pidevad muudatused ja sagedane kontroll võivad välja selgitada olulised vead tehtud töös. Sellisel juhul on oluline, et lisandunud vigane tegevus oleks asendatav eelmise, toimiva versiooniga. (Stapleton 2003: 3-22)

Dünaamiline tarkvaraarendus näeb ette, et tähelepanu pööratakse suures osas ka projekti eelfaasis. Tähtsaim idee eeltegevuste läbiviimisel on ulatuse määratlemine, kus osapooled fikseerivad põhiohused ja funktsioonid, mis peavad jääma muutumatuks. See tagab projekti õige suuna ning välistab ettearvamatu, samas ollakse avatud kõrvaliste funktsioonide muutmiseks. Kogu arendustegevust peab saatma pidev kontroll ja testimine. Tagasiside optimeerimiseks ja kiireks informatsiooni vahetuseks peab suhtlemine olema korraldatud koostööd ja usaldust toetavalt. (Samas)

Põhitehnikateks on ajaraamistik (*timeboxing*), PVSE (*MoSCoV*) ja lihtsustatud töötoad (*facilitated workshops*). Ajaraamistiku käigus fikseeritakse ära kindlad kontrollpunktid, mis määravad ära mis ajaks mingi osa tööst peab olemas olema. PVSE tähendab nõuete fikseerimist, kus P- peab olema, V- võib olla, S- saab olla, E- ei ole vaja. See määrab prioriteedid ning tagab põhjalikuma planeerimise. Lihtsustatud töötoad on töövõte, mille abil kogutakse informatsiooni kõigilt osalistelt. Üheks töötoaks võib olla näiteks PVSE määratlemine, teiseks aga testimise planeerimine. Töötuba kaasab kõiki osapooli ning baseerub aktiivsele suhtlusele. (Probhaker 2006: 29-31)

Ulatuse järgi hinnates on dünaamiline tarkvaraarendus sobilik astmeliseks ja kiireks arenduseks väikeste kuni keskmiste projektide teostamiseks. Dünaamiline tarkvaraarendus toetab mitut meeskonda, kus ühe meeskonna suurus võib varieeruda kahest kuni kuue liikmeni. Meetod ei kehtesta piiranguid programmeerimise, tehniliste vahendite ega füüsilise keskkonna kohta. (Qumer, Henderson-Sellers 2007)

Antud meetod eeldab küllalt põhjalikku eelplaneerimist ning meetodilist lähenemist arendusprotsessi, mis viitab sarnasusi traditsioonilisele lähenemisele. Samas tagavad selle põhimõtted vabamat suhtlemist ning avatus muutustele kinnitab paindlikkust (vt ka tabel 4 lk 9). Kuna meetodi idee tuleneb traditsioonilisest lähenemisest, on see sobilik riskide maandamiseks. Paindlik suhtumine muudab selle sobivaks aga ka väikeste projektide jaoks, kus on suurem riskikartlikkus.

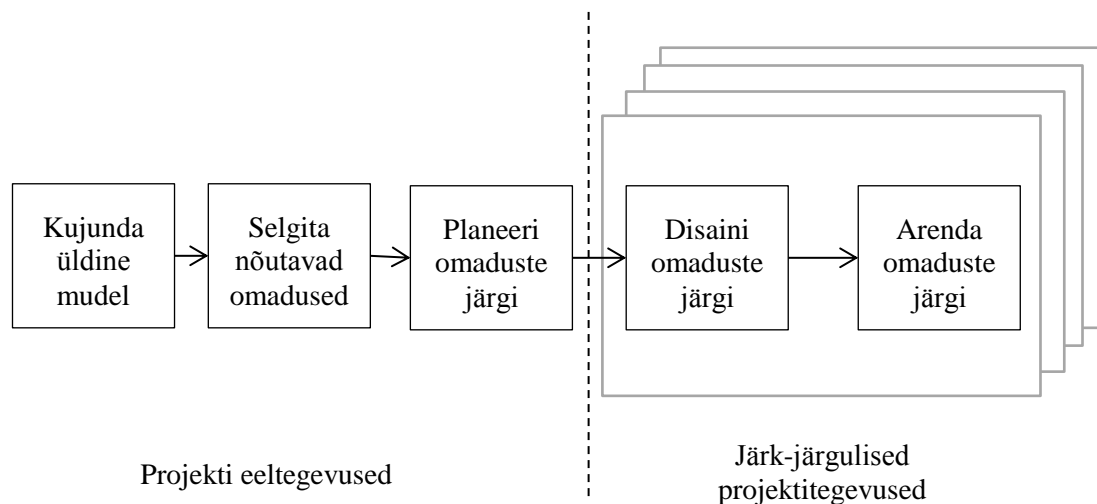
### **1.3.3. Omaduspõhine tarkvaraarendusmeetod**

Sarnaselt dünaamilisele tarkvaraarendusele ühendab traditsioonilise meetodi kaalutletud lähenemist ja avatust muudatustele ka omaduspõhine tarkvaraarendusmeetod. Keskseks teguriks on meetodi puhul kliendi ja arendajavahelised suhtlused, kus selgitatakse välja projekti üldine ülesehitus.

Omaduspõhine tarkvaraarendus (*Feature Driven Development*) on samm-sammuline ja juurdekasvuline meetod. Kuigi seda kirjeldatakse kui paindlikku meetodit, on see loodud lähtuvalt traditsioonilisest lähenemisest, hõlmates sealhulgas planeerimis-, disaini- ja dokumenteerimisfaase (Meyer jt 2007 :196). Põhivajadus meetodi järele tuleneb asjaolust, et projektijuhid tunnustavad muutuvat keskkonda ning adaptiivsuse olulisust, samas soovivad säilitada piisavat kontrolli. Juurdekasvuline ja kohanduv projekt on juhtimise seisukohast keerukam kui traditsioonilise koskmeetodi rakendamine. Siiski on see eelistatud kuna pakub kindlustunnet, et projekt üldise paindlikkuse tulemusena kaduma ei läheks. (Hunt 2006: 161-162)

Nimetatud meetod suurendab projektijuhi turvatunnet ning annab võimaluse olukorda paremini kontrollida. Järgnev joonis kujundab omaduspõhise tarkvaraarenduse viit põhiprotsessi, mille abil meetodit rakendada.





**Joonis 3.** Omaduspõhise tarkvaraarenduse viis protsessi ( Meyer jt. 2007: 197)

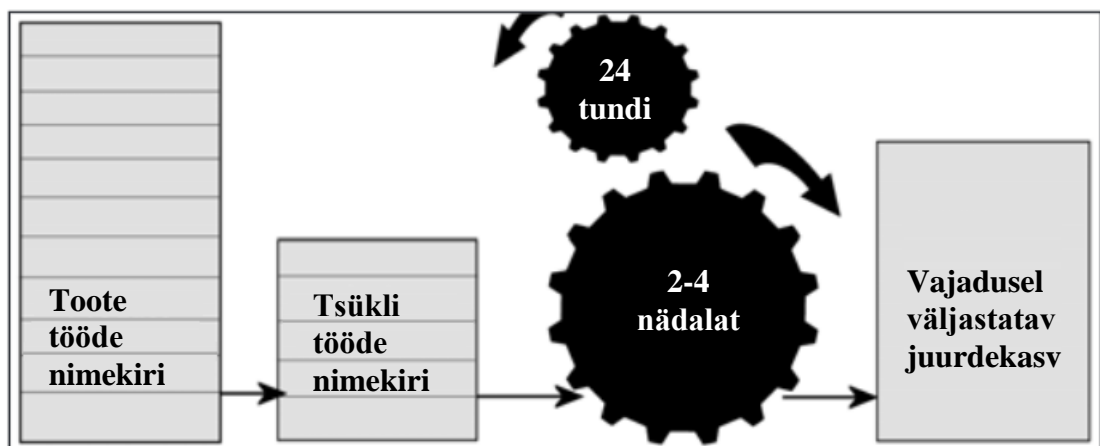
Omaduspõhisel arendusel algab projekt üldise mudeli väljatöötamisega, kus kirjeldatakse projekti kulgemist, ulatust ja iga probleemi sisu. Järgmisena tuleb välja selgitada nimekiri projektiga teostatavatest kliendi soovitud omadustest. Nimekiri peaks olema võimalikult täpne ning iga tegevus teostatav vähemalt kahe nädalaga. Viimane eeltegevus on omaduskogumike ja omaduste analüüsimine. See hõlmab eelkõige sorteerimist kliendipoolsete prioriteetide, ajalise osatähtsuse ja tehniliste sõltuvuste järgi. Projektitegevus koosneb omadustepõhisest disainimisest ja arendust, kus need paigutatakse töötappide kaupa järjestusse. Üks etapp on integreeritud teiste omadusetappide ja –meeskondadega. (Meyer jt. 2007: 198)

Sarnaselt Dünaamilisele tarkvaraarendusele on Omaduspõhine tarkvaraarendus tugevasti seotud traditsioonilise koskmeetodiga. Suur rõhk planeerimisele ning riski vähendavatele tegevustele muudab selle eelistatud valikuks keerukate projektide puhul, kus projekti edust sõltub ettevõtte olulisel määral. Ka see meetod on laiahaardeline lähenemine mida saab rakendada erinevate projektide jaoks. Selle läbivaks ideeks on planeerimine ning järk-järguline arendustegevus.

### 1.3.4. Scrum meetod

Üks paljukajastatud meetoditest on *Scrum*, mis on tarkvaramaailmas palju tunnustust kogunud. Selle eripäraks on adaptiivsus ning võime toime tulla ka keerukamate olukordadega.

Schwaberi (2007: 102-103) järgi ei saa probleemi alati täielikult mõista ega defineerida ning keskenduda tuleb hoopis võimele kohanduda. Selleks tuleb kasutada empiirilist protsessi juhtimist, mille põhitaladeks on läbipaistvus, inspekteerimine ja kohanemine. Scrum meetod põhineb nendel samadel põhimõtetel ning on üks populaarsemaid, kuna on hästi rakendatav keerukas ja raskesti prognoositavas tarkvaramaailmas. See on täiendav, järk-järguline ja võimekas protsess, mille abil saab toime tulla nii väga keerukate kui ka lihtsate projektide teostamisel. Järgnev joonis selgitab *Scrum* meetodi ülesehituse loogikat.



**Joonis 4.** *Scrum* ülesehitus (Rossberg 2008 :63)

Kõik vajalikud nõuded, mis on teada projekti alguses, kogutakse toote tööde nimekirja (*product backlog*), mis on üheks *Scrumi* põhivahendiks. Meeskond vaatab nimekirja üle ning otsustab, milliseid nõudeid hakatakse täitma esimese tsükli jooksul. Need lisatakse tsükli tööde nimekirja, kus need jaotatakse veel täpsemateks osadeks. Seejärel püütakse tsükli nimekirjast moodustada lõpptootest valmis osa, mida on võimalik väljastada ja esitleda kliendile. Meeskond on isemajandav ja delegerib ülesandeid iseseisvalt ning on pidevas koostöös kliendiga. Tsükli lõpus vaadatakse projekti tellijaga koos üle tehtud

osa ning ühiselt määratletakse muutused ning lisafunktsioonid, mis järgmiseks korraks esitleda tuleb. (Rossberg 2008: 64)

Kogu *Scrum* protsess on kohandumis põhine, mis tähendab, et koostöös tellijaga toote töönimetamine täieneb ja prioriteedid muutuvad. Põhivahend *Scrumi* elluviimiseks on koosolekud, mis viiakse läbi nii tihti kui vajalik. Pikem koosolek tehakse projekti alustamiseks, kus selgitatakse välja vajalikud nõuded ja prioriteedid tsükli alustamiseks. Järgnevad igapäevased lühiajalised koosolekud, kus selgitatakse välja tehtud ja planeeritavad tööd ning arutletakse võimalike takistuste üle. (Samas: 66-69)

*Scrumi* kasutamine on toonud positiivseid muutusi suhtluses, koostöös, sagedases prototüüpimises ning vähendanud tarkvaraarenduse riske ja probleeme (Holmström jt 2006: 7). Tegemist on tõhusa ning kergesti kohandatava meetodiga, mis sobib nii lihtsamate kui keeruliste projektide teostamiseks. Projektimeeskonna süsteemne sisesuhtlus ning arutelu tellijaga annab kindlustunde järgmiseks etapiks ning vähendab riske, mis võivad realiseeruda arusaamatuste või konfliktide koosmõjul. Kuna meetodi ülesehitus ja idee on lihtsad, saab seda rakendada ka meeskondades kellel puudub varasem koostöökogemus. Lähtudes fikseeritud töö ülesehitusest säilitatakse kontroll meeskonna tegevuse üle, kui ollakse pidevalt avatud ka kliendi vajaduste suhtes.

#### 1.3.4. Ekstreemprogrammeerimine

See väle meetod on suunatud arendustegevusele, mille põhimõtted aitavad kergendada projektimeeskonna koormust ja keskenduvad efektiivsuse hoidmisele. Põhiprintsiibid, mis nimetatud meetodit iseloomustavad, on toodud järgnevas tabelis.

**Tabel 7.** Ekstreemprogrammeerimise juhtprintsii bid (Beck 2000:36-37; autori koostatud)

Põhimõte	Selgitus
Kiire tagasiside	Mida kiirem reaktsioon juhtunule, seda rohkem õpitakse.
Eelda lihtsust	Igat probleemi saab lahendada väga lihtsal ka kiirel moel.
Juurdekasvuline muutus	Kõik probleemid tuleb lahendada läbi väikeste muutuste.
Muudatuste omaksvõtmine	Mida rohkem on võimalikke valikud, seda lihtsam on leida lahendust.
Kvaliteetne töö	Ainuke vastuvõetav töö on suurepärase kvaliteediga.

Esimeseks põhimõtteks on saada tagasisidet, see tõlgendada ning edastada õpitu süsteemi. Seeläbi õpitakse vigadest ise ning kogemust saavad ka teised. Lihtsuse eeldamine tähendab seda, et 98% probleemidest saab lahendada väga kergelt, nendelt kokkuhoitud aja võib rakendada ülejäänud 2 protsendile. Keerukate probleemide tekkimisel tuleb teostada see osa, mis on koheselt lahendatav ning lisada keerukamad sammud aja jooksul. Meetodi puhul tuleb suures osas lähtuda juurdekasvu printsiibist, mille järgi tuleb olukordade lahendamiseks teha järjestikulisi väikseid muudatusi. Selleks, et neid ellu viia tuleb olla avatud erinevatele võimalustele. Vältida tuleb ühele võimalikule lahendusele kindlaksjäämist. Läbi kõige selle töö on lõppeesmärgiks siiski kvaliteet, mis saab arengu tagamiseks olla ainult väga hea. (Samas)

Juhtprintsipiide jälgimine soosib avatust ja pidevat arengut. Olulisemad töövahendid millega neid põhimõtteid ellu viiakse on toodud järgnevas tabelis.

**Tabel 8.** Ekstreemprogrammeerimise töövõtted (Beck, Andres 2004: 64-75; autori koostatud)

Töövõtte	Selgitus
Lähestikku paiknemine	Töökeskkond peab olema avatud, vaheseinteta ning võimaldama mugavat suhtlemist. Tekitab suurema informatsiooni vahetumise.
Meeskonna tervikustamine	Probleemide lahendamiseks tuleb eelistada meeskonnatööd ja tekitada ühtekuuluvustunnet, mis pakub liikmetele vaheldust ja põnevust.
Informatiivse töökeskkonna loomine	Töökeskkonda sisenedes peaks projektist aru saama ka kõrvaline isik. Selleks tuleb kasutada juhtumikaarte, plakateid ja teadetahvleid.
Energia-rikas töö	Tööd tuleb teha nii kaua kui suudetakse olla produktiivne.
Paaris-programmeerimine	Kaks inimest peavad saama töötada ühe masinaga. Olemuselt väsitav kuid efektiivne lähenemine.
Kasutajakaardid	Projekti võimalikud lahendused kirjutatakse kaartidele, mille liikmed saavad neid näha ja mõelda nende teostamise peale.
Varu tekitamine	Igas plaanis peab olema mõni ülesanne, mille saab vajadusel kõrvale jätta.

Ühe liikme läbipõlemine lihtsalt selleks, et teha töötunde võib pikemas perspektiivis hävitada terve meeskonna vaimu. Paarisprogrammeerimine eeldab seda, et kaks koodi-kirjutajat saavad istuda mugavalt kõrvuti ühe masina ees. Tulemuseks on pidev täiendamine, dialoog ning parem veatuvastus. Parimal juhul soovitatakse vahetada koodipartnereid iga mõne tunni järel. Efektiivsuse tõstmiseks kasutatakse ka kasutajakaarte, kus selgitatakse mingisugune olukord, mis tegema peab või mida saavutada tahetakse. Need määratletakse ära võimalikult varakult ning meeskond saab

lahenduseni jõuda võimalikult väikese investeeringuga. Kindlustamaks, et lubatud funktsionaalsus ka esitletakse, tuleks algplaani panna rohkem nõudeid, kui tahetud. Seeläbi on planeeritud suurem funktsionaalsus, kui klient soovis, kuid probleemide korral on võimalik kuhugi taganeda.

Tegemist on nn kergekaalulise meetodiga väikese kuni keskmise suurusega meeskondade jaoks, kes arendavad tarkvara kiirestimuutuvas keskkonnas. Ekstreemprogrammeerimine on suures osas iseorganiseeruv lähenemine ning seda on lihtsam teostada väikeste meeskondade puhul. Eelistatud on projektid, kus püütakse teha tuntud rakendusi varem väljakujunenud ülesehituse abil. Sellisel juhul on projekti teostajad juba teadlikud kuidas ja milliste lahendustega probleemi lahendada. Meetod on paremini rakendatav ka sellistel puhkudel, kui mastaabi kasv ei ole probleemiks. (Hunt 2006 :159)

Nimetatud väledat lähenemist on võimalik rakendada ning siduda mitmete teiste lähenemistega. Meetod pakub kindlad põhimõtted, mille järgimiseks on välja töötatud mitmed töövahendid.

#### **1.4. Paindlike meetodite võrdlus**

Eelnevalt mainitud meetodid on põhilähenemised, mida erinevates projektijuhtimist käsitletavates allikates kajastatakse. Järgnevalt vaadeldakse nende erinevusi ja sarnasusi kokkuvõtvalt.

Kristallmeetodid on põhimõtete kogum, mille järgi saab projekti teostaja ise hinnata kui raske või keeruka projektiga tegemist on (vt joonis 2 lk 12). Mida väiksema projektiga on tegemist, seda paindlikum saab olla ka projekti kulg. Kristallmeetodid on avatud igasuguse arendusmeetodi jaoks, kuna lähtub põhimõttest, et projekti teostaja määrab vastavalt selle keerukusele ise meetodi. Seega võib kristallselge projekti jaoks kasutada *Scrumi* või ekstreemprogrammeerimist, samas kui oranžiks liigituv projekt tähendab põhjalikumat eelplaneerimist. Paindlikkuse säilitamiseks kõigis projektitüüpides tuleb järgida põhimõtteid(vt tabel 5 lk 13), mis annavad projektide teostamiseks ühtse aluspinna.

Dünaamiline tarkvaraarenduse meetod on sarnaselt kristallmeetodile pigem juhiste komplekt kui konkreetne juhend. See sobib olukordades, kus on oluline fikseerida teatud tingimused enne projekti algust, jäädes siiski avatuks kliendi nõudmiste muutustele. Olemuselt on meetodi põhimõtted (vt tabel 6 lk 14) sarnased kristallmeetodite omadele, kuid on mõeldud kindlale projektitüübile, kus on vajalik siduda riskide maandamine ja avarus muutustele.

Omaduspõhine tarkvaraarendusmeetod sisaldab eelnimetatutega võrreldes konkreetsemaid juhiseid projekti teostamiseks, keskendudes disainimisele ja planeerimisele. Sarnaselt dünaamilisele tarkvaraarendusele pannakse rõhku riskide maandamisele. Teistega võrreldes on planeerimisfaas põhjalikum ja suur roll on traditsioonilisest lähenemisest tuleneval järk-järgulisel arendustegevusel. Seega on meetod sarnaselt dünaamilisele arendusele avatud nii väikeste kui ka suurte projektide jaoks.

*Scrum* pakub sarnaselt omaduspõhisele lähenemisele vahendeid projekti kavandamiseks ja läbiviimiseks. Meetod ei keskendu põhimõtete järgimisele vaid tsükli põhisele arendustegevusele, kus on pidev kontakt projekti tellijaga ning tegevus on pidevalt avatud muudatustele (vt ka joonis 4 lk 17). Sarnaselt eelnimetatud meetoditele on ka *Scrum* hea meetod riskide maandamiseks, kuna võimaldab planeerimisfaasi kohandada vastavalt vajadusele. Kuna pidevalt küsitakse kliendi tagasisidet ja projekti eelfaasis ei fikseerita olulisi nõudeid, on see mõnevõrra avatum muutustele kui eelnimetatud meetodid.

Ekstreemprogrammeerimine pakub sarnaselt omaduspõhisele ja dünaamilisele meetodile kindlaid põhimõtteid, mida järgida (vt tabel 7 lk 19). Erinevalt teistest nimetatud meetoditest pakub ekstreemprogrammeerimine konkreetseid töövahendeid disainimise, programmeerimise ja testimise jaoks (Abrahamsson jt 2010: 39). Meetod ei fikseeri planeerimistegevust ning sarnaselt *Scrum* juhistega on see paindlik ja võrdlemisi kiire.

Mitmel korral kattuvad erinevate meetodite puhul olulised põhimõtted nagu avatus muutustele, suhtlemine inimestega, ausus ja pidev täiendamine. Lähenemisviisid nagu

Omaduspõhine ja Dünaamiline tarkvaraarendus on loomult rangemad ning sobilikud ka suurte ja riskantsete projektide puhul. Eelnevalt kirjeldatud meetodid on enimmainitud suhtumised paindliku projektijuhtimise teostamiseks ja vastavalt vajadusele on võimalik kombineerida erinevaid põhimõtteid ja ideid.

Paindlikud meetodid erinevad traditsioonilistest lähenemistest suuresti oma põhimõtete poolest. Kuigi mitmel korral on paindlik meetod loodud toetudes traditsioonilisele koskmeetodile, hoiab põhimõtete järgimine suuremal või vähemal määral avatust muutustele. Traditsioonilised meetodid on sobilikud pigem projektides, kus on suur riskitase, teadaolevad suhteliselt kindlad eesmärgid ja võrdlemisi muutumatu töökeskkond. Paindlikud meetodid on osutunud edukamateks väikestes projektides kuna need jätavad rohkem aega tegutsemiseks ja suhtlemiseks. Madalam riskitase võimaldab hoiduda pidevast dokumenteerimisest ja keskenduda sobiliku toote valmistamisele.

Projekti suuruse hindamisel tuleb lähtuda eelkõige konkreetse projekti läbiviija seisukohast. Väikesemahulise projekti määratlemine on pigem subjektiivne ja sõltub projekti teostaja hinnangust nii olemasolevate kui ka vajaminevate ressursside, oskuste ja lahenduste kohta. Väikesemahulised tarkvaraprojektid on kindlalt määratletud tegevused, mis on suunatud leidmaks lahendust probleemile kasutades selleks tarkvaralisi lahendusi. Väiksemate projektide puhul on eelistatud paindlike meetodite rakendamine, kuna need tähendavad suuremat vabadust ning kiiremat reageerimisega muutustele. Traditsioonilised lähenemised on sobilikud pigem projektides, kus on suur riskitase, teadaolevad suhteliselt kindlad eesmärgid ja võrdlemisi muutumatu töökeskkond.

## 2. PAINDLIKE MEETODITE VÕRDLUS NÄIDISPROJEKTI ABIL

### 2.1. Metoodiline alus

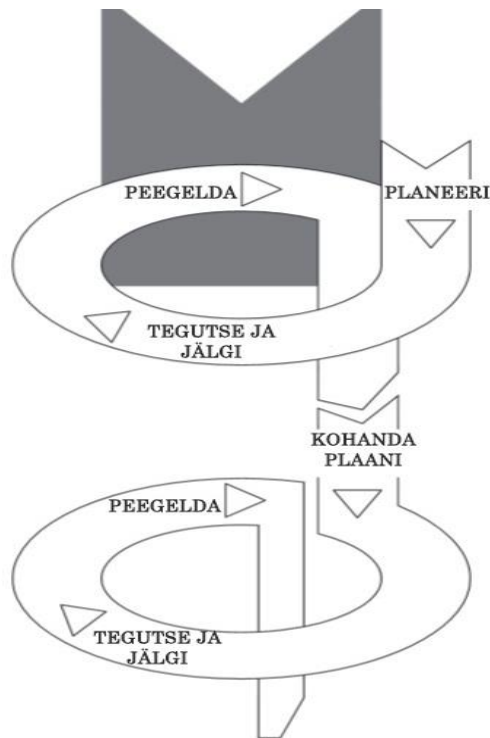
Lõputöö empiiriliste ülesannete täitmiseks kasutati käesoleva töö raames tegevusuuringut (*action research*). Nimetatud metoodiline lähenemine võimaldab uurijal olla osa uurimisobjektist ning aitab selgitada saavutatud tulemusi ja hinnata lisatud panust. Tegevusuuring on enesepeegelduslik uuring, mida viivad läbi praktikud, et parandada oma erialase tegevuse kvaliteeti nii enda kui ka teiste jaoks (Carr, Kemmis 1986 : 162). Winter ja Munn-Giddings (2001: 8) defineerivad seda kui sotsiaalse olukorra uuringut, mille viivad läbi isikud, kes on seotud konkreetse situatsiooniga ja mis on suunatud nende isikute kogemuste ja arusaamise kvaliteedi tõstmisele. Reason ja Bradbury kirjeldavad tegevusuuringut kui metoodikat, mis (2008: 3):

- reageerib inimeste soovile tegutseda loominguiliselt praktiliste valdkondadega;
- soodustab tegevust ja kaasamist koostööd nõudvates situatsioonides;
- keskendub väärtuse loomisele ning teemadele, mis puudutavad inimeste arengut, kogukonda ja nendevahelisi suhteid;
- on elav, arenev protsess, mida ei saa varasemalt kindlalt määratleda, kuna seotud inimeste mõistmine probleemidest ja olukordadest täieneb pidevalt.

Tegevusuuringu teostajad ei seo ennast uuringuga, et leida uusi fakte või kontrollida vastuvõetud teooriate õigsust. Lähenemisviisi kasutajad viivad läbi uuringut, et omandada informatsiooni konkreetsete probleemide kohta oma töövaldkonnas, kasutades selleks rakenduslikku lähenemist. (Stringer 2004: 3)



Kemmis ja McTaggart (2000: 567) leiavad, et tegevusuuring on loomult osalev ning viitavad sellele ka kui osalevale tegevusuuringule. Eelviidatud autorid leiavad, et tegevusuuring koosneb peegeldavatest tsüklitest, mis on välja toodud joonisel 5.



**Joonis 5.** Osaleva tegevusuuringu spiraal (Kemmis, McTaggart 2000: 595)

Tegevusuuring on spiraalne ja ennast täiendav. Iga tsükli alguses toimub planeerimine, seejärel tegutsemine ja jälgimine ning viimaks peegeldamine, kus vaadeldakse tehtud otsuseid. Vastavalt eelmises tsüklis toimunud planeeritakse järgmine samm, kus viiakse läbi järgmised tegevused. Autorid ei soovita seda mudelit kasutada kui fikseeritud struktuuri kuna reaalsuses on tsüklite erinevad protsessid omavahel põimunud või vahetuses. Algsed plaanid võivad muutuda tsükli kestel ning kogu protsess on muutuv, avatud ja reageeriva loomuga ning liigselt range lähenemine võib piirata uuringu tõhusust. (Samas: 595-605)

Sarnast mudelit on esitletud ka O’Leary (2004: 141), kelle tegevusuuringu tsükli elementideks on vaatlemine, peegeldamine, planeerimine, tegutsemine. Tegevusuuringu läbiviimiseks soovib tsüklilist mudelit ka Elliot (1991: 71), kelle tsükli elemendid on

üldidee määratlemine, faktide uurimine, planeerimine, tegutsemine, hindamine ning muutuste planeerimine.

Stuart jt (1997 :164) leiavad, et kuigi tsüklite ja nende siseste sammude määratlus võib olla vaieldav, on tsükliline mudel äärmiselt kasulik ja praktiline lähenemine. Uuringu läbiviijad näevad probleemi, mõtlevad lahenduste peale, proovivad erinevaid lahendusi ja vaatavad kui hästi need töötavad.

Tegevusuuring on olemuselt sarnane paindlikele projektijuhtimismeetoditele. Spiraalmudeli järgi on see samm-sammuline ja pidevalt täienev uurimistegevus, mis on avatud muudatustele ja võimaldab reageerida tekkinud situatsioonidele. Seetõttu võib tegevusuuringut pidada sobilikuks sarnase, loomuga muutliku projekti läbiviimise uurimiseks. Tegevusuuringu kajastamiseks on välja töötanud kuus kriteeriumit, mille abil tagada uuringu kasulikkus ja anda selles ülevaade. Need kuus kriteeriumit on toodud tabelis 9.

**Tabel 9.** Tegevusuuringu kriteeriumid (Nielsen 2007: 361-365; autori koostatud)

Kriteerium	Kuidas rakendada
Rollid	Selgita uurijate ja osalejate rolle.
Dokumentatsioon	Kirjelda, kuidas andmeid koguti.
Kontroll	Selgita, kuidas kontrollisid tulemusi.
Kasulikkus	Hinda teadasaadud tulemuste kasulikkust probleemi lahendamiseks
Raamistikud	Ühenda sooritatud tegevused teoreetiliste raamistikega.
Ülekantavus	Too välja olukorrad, kus tulemused on ülekantavad sarnastele olukordadele

Rollide selgitamine annab ülevaate ja selguse uuringuga seotud inimestest. Tegevusuuring tugineb kirjutistele, mis on tehtud uuringu jooksul toimunud ideede, tegevuste ja tulemuste kohta. Kontrollimiseks võib uurida seonduvate isikute huve ja hinnata kui hästi on tegevused nendeni suunanud. Kasulikkuse määratlemine aitab hinnata, kas uuringu läbiviija sai midagi uut teada ning raamistikule tuginemine näitab tegevuse kaalutletud sooritamist. Saadud teadmiste väärtuse tõstmiseks tuleb mainida sarnased situatsioonid, kus neid kasutada saab. (Nielsen 2007: 362).

Dokumenteerimiseks on sobilikud meetodid nagu süvaintervjuud, uuringuga seonduvad dokumendid, arhiivmaterjalid, olukorra kirjeldused (Pettigrew 1990: 274-277). Lisaks annavad head ülevaadet uuringust päeviku vormis kirjutised (Jepsen jt 1989: 207-211).

Lähenemisviisi rakendamiseks kasutatakse käesolevas töös tegevusuuringu dokumenteerimise meetodeid nagu uurimispäevik, tegevuste salvestused ning juhtumi analüüsi. Uurimispäeviku näol on tegemist autori märkmetega, kus on kirjeldatud projekti erinevates etappides tehtud otsused ja juhtumeid. Selleks, et näha progressi ning erinevate otsuste mõju lõpptulemusele salvestas autor projekti valminud osad erinevatel etappidel. Tegevuse salvestisteks on seega erinevad projektiga seotud dokumendid, meilivestlused, prototüübid ja koodiread, mis projekti eri etappidega seotud olid. Uurimispäeviku ning tegevuste salvestiste abil teostati juhtumiuuring, mille käigus testiti teooriaosas esitatud meetodite sobivust. Sobivuse kontrollimise hindamisel lähtuti nende eesmärgipärasusest.

Kirjeldatud metoodikat on projektide uurimiseks kasutanud näiteks Shames jt (2013). Nende sõnul on projektide uurimiseks kasulik kasutada tegevusuuringut, kuna see toetab selle nõrkade kohtade leidmist ning võimaldab teostada muutusi, mis võivad tuua kaasa oukorra olulise paranemise. O'Sheedy (2012: 85-88) sõnul on tegevusuuring hea vahend IT- projektide juhtimiseks, kuna võimaldab uurijal mõjutada projekti kulgu, ilma et oleks vajadust sõltuda välisest uurijast. Tänu sellele on võimalik saada parem mõistmine projektijuhtimisest pidevalt muutuvate kriteeriumitega projektides.

Juhtumiuuringu vajadus tuleneb probleemist, et tänapäeva arvutiajastu areng on kaasa toonud suurenenud vajaduse tarkvaraprojektide teostamiseks. Teoreetilised käsitlused viitavad sellele, et paindlikud meetodid on tarkvaraprojektide teostamiseks paremad, kui traditsioonilised. Paindlike meetodite hulk on üsna suur ning enamik neist sobivad nii väikeste kui suurte projektide teostamiseks. Iga meetodi pioneer on loonud omad põhimõtted ja suhtumishoiakud, mille alusel projekte teostada. Sellest hoolimata pole väikesemahulise tarkvaraprojekti puhul teada, milliseid meetodeid oleks otstarbekas rakendada teatud situatsioonides ning milliseid tuleks nende keerukuse ja ranguse tõttu üldse vältida. Võib-olla on väikesemahuliste tarkvaraprojektide teostamiseks kasulik hoopis kombineerida erinevaid põhimõtteid ja seonduvaid tööriistu. Kindla meetodi

valimine on algaja projektijuhi jaoks keerukas ning antud juhtumiuuring on suunatud andma ülevaadet erinevate lähenemiste rakendatavuse kohta väikesemahuliste tarkvara-projektide teostamiseks.

## **2.2. Tegevusuuringu läbiviimine ja uuritava objekti kirjeldus**

Lõputöö empiirilise osa täitmiseks kasutati juhtumiuuringut, milleks on projekt töövahendusportaali loomine Pärnumaal. Erinevate meetodite kasutamine ühe projekti raames oli oluline, kuna see pakub kindla võrreldava lähtepunkti, mille abil vaadelda, kas meetodi rakendamine oli kasulik või mitte. Sellega välditakse olukorda, kus uuritakse mingi kindla meetodi sobivust ainult teatud projekti kontekstis. Lõputöö eesmärgist lähtuvalt oli oluline katsetada erinevaid meetodeid ühe projekti raames, kuna võimaldab meetodeid omavahel võrrelda ning seega anda soovitusi ka teiste sarnaste projektide teostamiseks. Projektid (eriti paindlikud projektid) võivad olla loomult väga erinevad ning kui üks meetod osutub ühes projektis edukaks, ei pruugi see tähendada, et see toob kaasa edu ka teistes projektides.

Uuritav projekt valmis koostöös Eesti Töötukassa ja Pärnumaa Ettevõtlus- ja Arenduskeskusega (PEAK). Tegemist on teooria osas välja toodud kriteeriumitele vastava väikesemahulise projektiga, mille probleemistiku lahendamiseks tuli kasutada tarkvaralisi vahendeid. Projekti õnnestumiseks oli oluline reageerida muutustele ja ettepanekutele nii PEAKi kui ka töötukassa poolt ning samas arvestada ka lõppkasutaja vajadusi. Käesolev peatükk keskendub tegevusuuringu mudelist lähtuvalt uuringu planeerimisele. Ennem kui saab hakata planeerima konkreetsete meetodite rakendamist, on vaja aga selgitada projekti eesmärk ja vajalikkus.

Projekti teostamine tulenes vajadusest, et hetkel valitseb Eestis olukord, kus tööjõu väljavoolavus riigist on suureks probleemiks. Probleemi süvendab ka asjaolu, et lisaks töövõimelise elanikkonna väljarände tõuseb pidev riigisisene tasakaalutus elatus-taseme ning elukvaliteedi osas. Inimesed, kes soovivad keskmisest paremaid töötingimusi ning palka, otsivad töötamisvõimalusi kas välismaalt või siis suurematest Eesti linnadest. Rahva ja eluruumide loenduse (2011) andmetel on põhiliseks tööjõu väljarände sihtkohtadeks Soome (töötab 15 140 inimest), Norra (1872 inimest) ning

Rootsi (1532 inimest), kokku välisriikides ligi 25 000 elanikku. Loendus näitas, et riigisisene rändlus on pigem ühesuunaline ning Harjumaale orienteeritud. Merike Teder (2012) andmetel kaotas enim elanikke siserände tõttu Tartu ja Pärnu, Tallinna rändesaldo oli 1430 inimest.

Riina Martinson (2012) tõi välja, et Pärnumaal elavate inimeste arvu kukkumisel on suur roll lisaks negatiivsele iibele ka tööstusettevõtete järjepidev vähenemine. Töökohtade arvu vähenemisest tulenevalt ongi Pärnu elanikud sunnitud otsima töötamisvõimalusi mujalt, mis põhjustab pendelrännet ning lisaks püsivat elukohavahetust.

Projekti idee sügavamaks selgituseks on vajalik ka analüüsida põhjuseid, mis hetkelist probleemi põhjustavad. Probleemistikust ülevaate saamiseks saab kasutada probleemipuu meetodit, mille käigus analüüsitakse probleemide tekkepõhjused, uuritakse kuidas need edasi arenevad ning millised lõpptulemused. Lõpptulemusteks on probleemid, mida inimesed tajuvad ja näevad, probleemipuu aitab selgitada nende algpõhjuseid ja kujunemist.

Probleemipuu joonis on toodud lisas 1 lk 62, mis selgitab põhjuseid, miks ettevõtjad piirkonda siseneda ei julge. Probleemi põhjusteks on peamiselt madal palgatase, tööjõu võrdlemisi madal efektiivsus (lisaväärtus) ning inimeste madal rahulolu piirkonna, riigi ning iseendaga. Selle põhjusel otsivad inimesed paremat elupaika, mis toob kaasa väljarände ning piirkonna majandusliku mahajäämise. Kuna inimesed lahkuvad piirkonnast, väheneb oskustöölise arv ja levib inimeste üldine rahulolematuse, mistõttu ei soovi ettevõtteid piirkonda siseneda. Ohutum on siseneda piirkonda, kus tööjõud on mitmekesisem ning kättesaadavam. Sellega aga süvendatakse juba olemasolevat probleemi, kuna ettevõtete eemalehoidmine ning olemasolevate ettevõtete väljumine piirkonnast toob taaskord kaasa töökohtade vähenemise.

Hetkeolukorda kavatsebki PEAK lahendada ning on algatanud projekti, mis leevendab väljarännet kõige fundamentaalsemal tasandil – tööhõive ja madalate palkade tõstmise näol. Projekti eesmärgiks on Loode-Pärnu tööstuspiirkonna arendamine, kus ettevõtetele pakutakse linna poolt taristu väljaarendamist ning tööstusrajooni elavdamise soosimist (ehitus- ja tegevuslubade näol). Leitud on mitmed ettevõtteid, kes

oleksid huvitatud oma tegevuse paigutamisest Pärnusse ning kokku oleks nende poolt nõutav tööjõuvajadus ligikaudu 360 inimest. Probleem seisneb aga selles, et piirkonda sisenevatel ettevõtetel ei ole kindlust ega teadmist kohaliku tööjõu saadavuse kohta. Kuna järjest suurenev väljaränne nii Soome kui ka Harjumaale näitavad, et soovitud tööjõust võib väheks jääda, on ettevõtetel kindlam hoolimata eriliste soodustuste puudumiseta minna näiteks Harjumaale, kus on teada, et tööjõudu on lihtsam leida. Selle vältimiseks tuleb leida moodus, kuidas näidata ettevõtjatele, kui palju on Pärnumaa inimeste seas nende vajadustele sobivaid tööst huvitatud inimesi. (Sulev Alajõe suuline teade 05.11.2013)

Projekti viis läbi töö autor ning kasutas selleks erinevaid teoreetilises osas kajastatud projektijuhtimislikke võtteid ja meetodeid. Selleks, et näha progressi ning erinevate otsuste mõju lõpptulemusele salvestas autor projekti valminud osad erinevatel etappidel. See hõlmab prototüüpe, disainilahendusi ja ka mittekasutatud alternatiive. Samuti aitab tehtud otsuseid vaadelda projektipäevik, mis kajastab projekti erinevaid etappe.

Lähtuvalt uuritud taustandmetest ning projektipäeviku kirjutistest (vt ka lisa 2 lk 63) saab välja tuua projekti kokkuvõtavad ning seda kirjeldavad elemendid. Järgmine tabel 10 annab ülevaate projektiga seotud rollidest.

**Tabel 10.** Projektiga seotud rollid ja nende kirjeldus (autori koostatud)

Seonduv roll	Rolli kirjeldus
PEAKi esindaja (Sulev Alajõe)	projekti tellija
Töötukassa Pärnu esindus	projekti tulemi administraator
Lõputöö autor	projektijuht
Projektiliikmed	projektitegevuste teostajad
Pärnumaa tööhuvilised	projekti lõppkasutajad
Pärnumaa tööpakkujad	projekti lõppkasutajad

Projekti tellijaks on PEAK, mille juhatuse liige, Sulev Alajõe nägi selle vajadust ning esitas idee lõputöö autorile, kes probleemistikule lahenduse pidi leidma. Töötukassa Pärnu esinduse liikmed aitasid kaasa funktsionaalsuse määratlemisel ning võtavad üle andmete administreerimise pärast projekti lõppu. Projekti lõppkasutajateks on inimesed, kes on huvitatud töötamisest Pärnumaal, kelle andmeid hakatakse andmebaasi koguma ning administreerima. Lõppkasutajateks on ka Pärnumaa tööpakkujad, kes andmebaasist sobivaid töölisi otsivad.

Tegevusuuringu teostamiseks oli oluline teada ka projekti suurust. Selle määratlusest sõltus kasutatavate meetodite valik ning hinnang nende rakendatavuse kohta. Projekti määratlus on toodud tabelis 11.

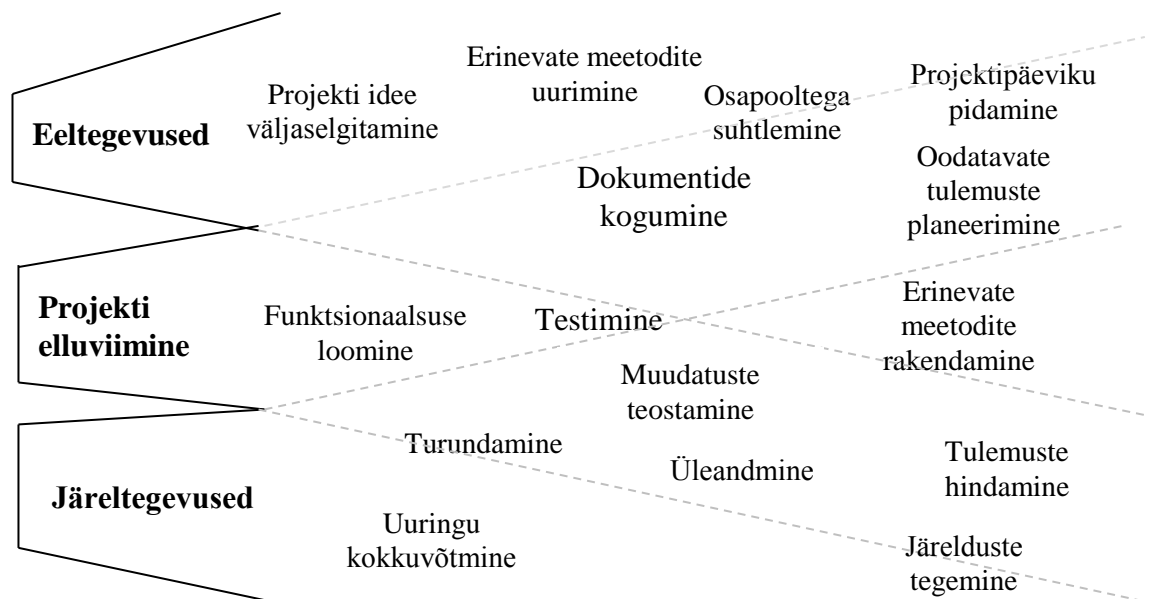
**Tabel 11.** Töövahendusportaali projekti määratlus (autori koostatud)

Tunnus	Väärtus
Kestvus	Kaks kuud
Tööaeg	Osaline tööaeg
Meeskonna suurus	Kuni 3 liiget
Nõutav oskuste hulk	Suhteliselt vähe erinevaid oskusi
Eesmärk	Üksikeesmärk: saada ülevaade töösoovijatest
Maksumus	Ainult projektiliikmete alternatiivkulu
Tulemused	Üheselt mõistetav: toimiv personaliandmebaas

Projekt kestis kaks kuud ning sellega tegeleti ainult osalise tööajaga. Projektiliikmeteks loeti projektijuhti, abitöölist programmeerimiseks ning selle tellijat. Projekti tellija, Sulev Alajõe, oli suhtluse koordinaator ettevõtjate ja töötukassa soovide vahendamisel. Abitöölise vajadus tulenes projektitegevuste eripärast tulenevast erinevate oskuste vajadusest, mis puudutas eelkõige programmeerimist Java abil. Projekti läbiviimisel püüti lähtuda paindlikust mõtteviisist, mille järgi tuleb püüda kasutada võimalikult lihtsat lahendust. Sellest tulenevalt oli erinevate nõutavate oskuste hulk üsna väike. Eesmärk ning tulemused olid üheselt mõistetav- luua personaliotsingu andmebaas, mille abil saab filtreerida töötajaid vastavalt tööpakkujate vajadustele. Projekt maksumuse all oleva projektiliikmete alternatiivkulu all peetakse silmas seda, et projekt ei läinud rahaliselt maksma mitte midagi ning kulus ainult projektiliikmete aeg (mida oleks saanud kasutada mingi muu töö tegemiseks). Seega sisulist rahalist kulu ei olnud.

Nimetatud kriteeriumid vastavad tabelis 3 lk 6 toodud kriteeriumitele, mille järgi hinnates on tegemist väikesemahuliste projektiga. Kuna erinevate organisatsioonide (projekti läbiviijate) jaoks võib hinnang olla erinev ning autoril puudus varasem projektijuhtimise kogemus ning programmeerimise oskus, võis projekti hinnanguline suurus muutuda. Sellisel juhul oleks tulnud eelistada meetodeid, mis aitavad maandada riske ja tegutseda plaanipõhiselt. Siiski arvestades, et projekti tegevus oli lühiajaline ning ei ohustanud läbiviijat rahaliselt, hindab autor nimetatud projekti väikesemahuliseks. Kuna projekti eesmärkide saavutamiseks on kesksel kohal tarkvaralised lahendused, on tegemist väikesemahulise tarkvaraprojektiga.

Tegevusuuringu abil kontrolliti teooriat ja selle rakendatavust erinevates situatsioonides, mis puudutavad projektiga seotud isikutega suhtlemist, muutuste tegemist ning katsetamist. Võrreldakse erinevaid etappe ja võimalikke kasutatavaid meetodeid. Projekti analüüsimisel lähtuti projekti erinevate etappide salvestistest ning projektipäevikust. Hinnati tehtud otsuste sobivust, alternatiivseid valikuid ja nendest tekkinud tulemusi. Projekti analüüsimisel lähtuti tsüklilisest lähenemisest vt joonis 5 lk 25), mida rakendati projekti erinevate etappide hindamiseks. Uuringu teostamise erinevad etapid on toodud järgnevas joonises.



**Joonis 6.** Uuringuga teostatavad tegevused (autori koostatud)

Projekti eeltegevused algavad projektiidee väljaselgitamisega, projekti teostamise faas algab funktsionaalsuse loomisega ning järeltegevused lõppevad järelduste tegemisega. Uuring algab projekti idee väljaselgitamisega ning lõppeb järelduste tegemisega. Töö eesmärgiks on testida erinevaid paindlikke meetodeid ühe projekti raames. Seetõttu ei peatuta ühe meetodi rakendamisele läbi mitme tsükli, vaid proovitakse erinevaid meetodeid erinevates projekti tsüklites. Projekti tsüklil lõppeb kas funktsionaalsuse loomise või selle arvestatava täiendamisega.

Kuna uuritav objekt on paindlik projekt, millel ei ole kindlat muutumatut tegevusplaani, on sobilik seda uurida metoodikaga, mis lubab ka uurimisprotsessi jooksvalt kohandada



ja täiustada vastavalt uurija arengule. Seetõttu kasutas autor juhtumiuuringut, mis võimaldab paindlikku uurimisprotsessi ja uurija osalemist tegevuses. Kuna tegevusuuringu tsükliline mudel näeb ette, et planeerimisfaas toimub enne reaalsete tegevuste sooritamist, on need kajastatud olevikuvormis. Peegeldamisfaasis antakse hinnangud tehtud plaanidele ning nende põhjal sooritatud tegevustele. Autor soovib juhtida ka tähelepanu asjaolule, et tegevusuuringu läbiviimise tsüklid ei pruugi samastuda projekti elutsüklitega.

### **2.3. Paindlike meetodite rakendamise planeerimine**

Projekti tegevusuuringu teostamiseks lähtutakse joonises 5 lk 25 toodud tegevusuuringu spiraalsest mudelist ning planeeritakse projekti läbiviimist. Samas peetakse silmas ka mudeli autorite hoiatust, et struktuuri range kinnipidamine võib erinevate meetodite sobivuse üle otsustamiseks tuleb aga hinnata nende sobivust projektiga. Planeerimise alustamiseks on vaja seega teada üldist projektiideed, mille abil hinnatakse erinevate lähenemiste sobivust ja toimimispotentsiaali. Projektitegevus on esitatud ka lisa 2 lk 63

Selleks, et määratleda projekti idee, uuriti esmalt SA Ida-Virumaa Tööstusalade Arenduse poolt tehtud küsimustikku, mille väljavõtted ja kokkuvõte on lisa 3 lk 67. Selgus, et nende ankeeti tuleb tööhuvilistel täita MS Excel failis ning saata see kontaktisiku meilile. Küsimustik oli ülesehituselt üsna keerukas, kuid selle täitmise lihtsustamiseks oli juurde tehtud ka põhjalik juhend. Põhiliselt uuriti inimese üldandmeid, erialast väljaõpet, soovitud töökohta ning isikuomadusi. Kuna vestlusest tellijaga selgus, et midagi sarnast on plaanis luua ka Pärnusse, lähtuti esimeste funktsioonide loomisel just nendest sisenditest.

Projekti eesmärgiks on luua andmebaas, mille peamiseks funktsiooniks on viia kokku ühe piirkonna tööandjad ning töötajad. Idee on väga sarnane ettevõtte personaliportfelli koostamisega, kuid kui tavaliselt tehakse portfelli ühele ettevõttele, siis antud andmebaasis on portfelli loodud mitmele ettevõttele. Seesuguse süsteemiga loodetakse teada saada kui palju on ühes piirkonnas töövõimelisi isikuid, kes sooviksid oma elukohas tööd ning kas see tööjõud kataks vajamineva töökohtade arvu uutes piirkonda sisenevates ettevõtetes. Andmebaasi loomiseks tuleb eelkõige koostada küsimustikud

ettevõtetele nende vajaduste väljaselgitamiseks ning seejärel teha küsimustik töö- võimelistele inimestele, et kaardistada tööjõu spetsialiseeritus ja töö nõudluse suurus. Järgnevalt kajastatakse planeerimisetapp, mis hindab erinevate meetodite kasutatavust nimetatud projekti raames.

Kristallmeetodid pakuvad lahendust organisatsioonile, mis viib läbi mitmeid projekte ning soovib vastavalt nende määratlusele rakendada erinevaid meetodeid. Lähtudes meetodi ülesehitusest ning tegevusuuringu aluseks oleva projekti määratlusest (vt ka tabel 11 lk 31), hindas autor, et tegemist on M6 tüüpi projektiga. See tähendab, et projekti edukus mõjutab inimeste mugavust ning meeskonna suurus ei ületa kuut inimest. Seega tuli kristallmeetodi suuniste järgi otsustades kasutada võimalikult paindlikku tarkvaraarendusmeetodit.

Nimetatud meetodi kasutamise hindamiseks oli oluline vaadelda ka selle põhimõtteid ning võrrelda, kui rakendatavad need olid nimetatud projekti kontekstis. Järgnev tabel 12 annab ülevaate kristallmeetodi põhimõtete sobivusest töövahendusportaali loomise projektiga.

**Tabel 12.** Kristallmeetodite põhimõtete sobivushinnang uuritava projektiga (autori koostatud)

Põhimõte	Hinnang sobivusele
Sagedane väljastamine	Sobib, kuna võimaldab kontrollida tehtud töö sobivust tellijaga.
Peegeldav täiendamine	Sobib, informatsiooni vaba liikumine meeskonnas annab hea ülevaate hetkeseisust.
Osmootne suhtlus	Enamasti ei sobi, kuna ei kasutata kontorikeskkonda ning ühes ruumis koosviibimise aeg on piiratud.
Isiklik turvalisus	Sobib, kuna anab võimaluse konstruktiivsele kriitikale.
Fookus	Sobib, kuna võimaldab analüüsida teostatavaid tegevusi.
Oskuskasutajate kaasamine	Sobib, asjast huvitatud inimesed võivad anda häid ideid võimalike täienduste kohta.
Tehnilised põhiomadused	Sobib teatud määral, kuna puudutab eelkõige koodi kirjutamist, mida nimetatud projektis on vähesel määral.

Kristallmeetodite põhimõtted on seega suures osas rakendatavad nimetatud projektiga, mitesobivad põhimõtted tulenevad projekti eripärast. Osmootne suhtlus ei sobi, kuna tegevus ei toimu ühtses kontoriruumis, seda saab aga teatud määral parandada tihedama

informatsiooni vahetamisega. Samuti ei sobi automaatne ja automatiseeritud testimine, kuna koodi kirjutamist on vähe. Seetõttu oleks automaatse testimissüsteemi loomine ebaotstarbekas, samas on koodi pidev testimine vajalik. Samuti on vajalik ka varasemate versioonide taastamisvõimalus.

Kristallmeetod viitab, et tuleb kasutada võimalikult paindlikku lähenemist. Järgnevalt jätkatakse planeerimisfaasi, uurides kuidas sobilik on teostatava projekti raames dünaamiline tarkvaraarenduse meetod. Järgnev tabel 13 annab ülevaate dünaamilise tarkvara arendusmeetodi põhimõtete sobivusest.

**Tabel 13.** Dünaamilise tarkvaraarenduse meetodi sobivushinnang teostatava projektiga (autori koostatud).

Põhimõte	Sobivus
Aktiivne kasutajaseotus on hädavajalik.	Sobib, põhimõtte järgimine vähendab arusaamatusi lõppkasutajate seas.
Meeskonnad peavad olema volitatud otsuste tegemiseks.	Sobib, printsiipi saab arvestada ka meeskonnasiseselt.
Keskendumine sagedasele väljastamisele.	Sobib, kuna vähendab vigade tekkimise ohtu.
Äriline sobivus on aktsepteeritavuse kriteeriumiks.	Sobib, kui on võimalik fikseerida põhinõuded, sest siis on ka teada, kuna on saavutatud minimaalne valmisolek.
Astmeline ja tsükliline arendus on kohustuslik.	Sobib, kuna võimaldab testida erinevate muutuste mõju ja proovida erinevaid võimalusi.
Muutused peavad olema tagasipööratavad.	Sobib, kuna võimaldab ülevaadet projektist ja vajadusel võtta toiminguid tagasi.
Põhinõuded tuleb fikseerida enne projekti algust.	Sobib, kui neid on võimalik fikseerida, sest need kindlustavad õige tegevussuuna.
Katsetamine peab toimuma läbi terve elutsükli.	Sobib, kuna võimaldab pidevat täiendamist.
Suhtlemine lähtub koostööst.	Sobib, sest koostööl põhinev suhtlemine suurendab sünergiaid.

Üsna traditsioonilise lähenemisega meetodina on dünaamilisel tarkvaraarenduse meetodil palju põhimõttelisi ideid, mis sobivad uuritava väikesemahulise projektiga. Probleeme võivad tekitada meetodi tuumikkomponendid, mis puudutavad projekti eeltegevusi. Võib tekkida olukord, kus põhinõudeid ei ole võimalik rangelt fikseerida või pole see lihtsalt otstarbekas. Sellisel juhul ei ole kasutatavad ka mitmed meetodile omased tööriistad nagu ajaraamistiku või PVSE kasutamine. Raskusi võib tekkida ka lihtsustatud töötubade rakendamisel, sest projekt ei ole niivõrd mahukas, et läheks vaja

pikka vestlusprotsessi kõigi projektiga seotud isikute vahel. Samas on võimalik seda teha miniatuursetel tasandil.

Teine võrdlemisi range, kuid siiski paindliku loomuga lähenemine - omaduspõhine tarkvaraarendus - pakub projektijuhile kindlustunnet, et projekti tellija oma nõudeid ootamatult ei muudaks. Nõuete fikseerimine võib ka selle meetodi rakendamise puhul muutuda keerukaks ning tuleb arvestada vajadusel minna üle teistele meetoditele. Kuna omaduspõhine meetod pakub välja põhiprotsessid, mille abil projekti ellu viia (joonis 3 lk 17), tuleb meetodi katsetamisel lähtuda just nendest.

Meetodi järgi tuleb kujundada üldine mudel, selgitada nõutavad omadused ja planeerida nende järgi kogu projekti tegevus (tsükliid). Nimetatud eeltegevuste rakendamiseks uuritava projekti raames võib tekkida vajadus planeerida alternatiivseid põhinõudeid ning kavandada mitu erinevat tegevusplaani. Samas võib tekkida vajadus ka eeltegevuste fikseerimine järk-järguliseks muuta. See aga vähendab turvatunnet projekti teostaja jaoks, mille pakkumine on antud meetodi põhisisuks.

Kristallmeetodi järgi otsustades tuleks M6 tüüpi projektide teostamiseks eelistada võimalikult paindlikku lähenemist. Omaduspõhine ja dünaamiline tarkvaraarenduse meetod on aga mõlemad suhteliselt traditsioonilise suunaga, mis võib nende testimise nimetatud projekti raames keerukaks muuta.

Paljukasutatud ja üsna paindliku loomuga meetod on *Scrum*, mis oma fikseeritud ülesehituse (joonis 4 lk 18), kuid avatud töövõtete abil on näidisprojektis suhteliselt kergesti katsetatav. *Scrum* eeldab sarnaselt omaduspõhisele ja dünaamilisele tarkvaraarenduse meetodile mõningast planeerimist. Samas on selle planeerimine palju rohkem avatud ja seega rakendatav väikeste projektide puhul, mille algusfaasis ei ole fikseeritud kindlaid lahendusviise.

Lisaks tööde nimekirja planeerimisele võib probleeme tekitada ka tsükliite fikseerimisel. Kaks kuni neli nädalat kestev tsükkel peab jaotuma 24 tunnisteks etappideks. Seetõttu peavad olema teada vähemalt tsükli tööde nimekirja. Arvestades meetodi paindlikku loomust, võib kohandada ka tsükliite pikkust ning vajadusel täiendada tsükli tööde nimekirja. Sobivust tõstab ka asjaolu, et *Scrumi* meeskonnad on iseseisvad ning

progressi üle arutletakse iganädalaste koosolekute abil. Koosolekuid, mis kajastavad tekkinud probleeme või tehtud töid, saab asendada seega kas meilivestluse või kiirsõnumitega. See säilitab meetodi operatiivse ülevaate tegevusest ning aitab meeskonnal hoida õiget suunda. Pidev tagasiside projekti tellijalt ning järgmiste tegevuste planeerimine võib väikesemahulise projekti puhul väga tõhusaks osutuda.

Kuna *Scrum* annab juhised projekti faaside korraldamiseks ning tehnikad ülesannete, meeskonna ja ülesannete planeerimiseks, on see tegevuste teostamiseks avatud teistele meetoditele. Konkreetseid töövahendeid funktsionaalsuse loomiseks annab ekstreemprogrammeerimine. Järgnev tabel annab ülevaate ekstreemprogrammeerimise põhimõtete sobivusest uuritava projekti kontekstis.

**Tabel 14.** Ekstreemprogrammeerimise

Põhimõte	Sobivus
Kiire tagasiside	Sobib, kuid sõltub projekti tellija kättesaadavusest.
Eelda lihtsust	Sobib, valides lihtsaima lahenduse on võimalik kiirendada tegevuse protsessi.
Juurdekasvuline muutus	Sobib, lisafunktsionaalsuse loomine väikeste sammudena väldib vigade tekkimise ohtu.
Muudatuste omaksvõtmine	Sobib, avatus erinevatele alternatiividele võimaldab teha parimaid saadavalolevaid valikuid.
Kvaliteetne töö	Sobib, kvaliteedi hoidmine tähendab suuremat rahulolu projekti kõigi osapoolte jaoks.

Põhimõtete poolest on ekstreemprogrammeerimine väga sobilik uuritava projekti läbiviimiseks. Ainus põhimõte, mille järgimisega raskusi võib tekkida on kiire tagasiside. Kui projekti läbiviija saab valmis mingi osa või prototüübi, sõltub selle tagasisidestamine projekti tellija kättesaadavusest. Muudatuste juurdekasvulisus sobib ja ühtib kõigi meetoditega. See puudutab andmebaasi ülesehitust, kuid suuremal määral on see oluline andmete analüüsitavaks muutmise juures, kus tuleb vältida liigset keerukust. Ekstreemprogrammeerimine pakub ka mõningaid töövõtteid, millega tarkvara loomise tegevust lihtsustada. Järgnev tabel annab ülevaate nende sobivusest uuritava projekti raames.

**Tabel 15.** Ekstreemprogrammeerimise töövõtte ja nende sobivus juhtumiuuringu raames (autori koostatud)

Töövõtte	Selgitus
Lähestikku paiknemine	Ei sobi, kuna valdav enamus tööst ei toimu ühises ruumis viibides.
Meeskonna tervikustamine	Sobib, ühtekuuluvustunde loomine tagab eesmärgipärase lähenemise.
Informatiivse töökeskkonna loomine	Ei sobi, kuna füüsilist ühest töökeskkonda enamasti ei ole.
Energiarikas töö	Sobib, kuid arvestada tuleb ka tähtaegadega.
Paaris-programmeerimine	Sobib, kui toimub koodi kirjutamine.
Kasutajakaardid	Sobib, kuid on efektiivne eelkõige ühises ruumis viibides.
Varu tekitamine	Sobib, kuna võimaldab vajadusel vähendada ajakulu.

Meetodi poolt soovitatud töövõtted, mis juhtumiuuringu projektiga ei sobi, puudutavad ühes ruumis töötamist. Uuritava projekti käigus kasutatakse aga virtuaaltöö vahendeid ning ühes ruumis tehtavat arendustegevust ei pruugi üldse esineda. Sellest hoolimata on sobivad töövõtted nagu mõne lisafunktsiooni varuks jätmine, meeskonna ühistunde loomine. Projekti teostamisel saab testida sobilikke töövõtteid ja järgida meetodile iseloomulikke põhimõtteid.

Esmane planeerimine andis ülevaate erinevate meetodite kasutatavuse kohta uuritava projekti raames. Kristallmeetod viitab, et tuleb kasutada võimalikult paindlikku lähenemist. Töös kajastatud paindlikumad meetodid on *Scrum* ja ekstreemprogrammeerimine. *Scrum* kasutamine annab projekti läbiviimisel juhised selle üldise kujunemise kohta ja on kasulik ka erinevate tehnikate määratlemisel. Ekstreemprogrammeerimine annab vajalikud tööriistad, millest mitmed sobivad ka antud projekti raames kasutamiseks. Väikesemahulise tarkvaraprojekti teostamiseks on sobilikud ka mõneti rangemad meetodid nagu dünaamiline tarkvaraarendusmeetod ja omaduspõhine lähenemine, mis sobivad suures osas oma põhimõtete poolest. Probleeme võivad tekitada nõuded, mis puudutavad põhjalikku ja fikseeritud planeerimist. Siiski saab meetodeid katsetada, lähtudes nende põhimõtetest ning tehes vajadusel järeleandmisi ranguse osas.

## 2.3. Projekti elluviimise tegevusuuring

Projekti teostamise analüüsimisel lähtutakse tegevusuuringu spiraalse analüüsi mudelist. Eelplaneerimisest selgus, et kõik teooriaosas kajastatud meetodid sobivad suuremal või vähemal määral näidisprojekti raames testimiseks. *Scrum* meetod on sobilik uuringu aluseks oleva väikesemahulise projekti jaoks ning on ülesehituselt lihtne. Kuna ka kristallmeetodi sobivusanalüüs näitas, et kasutada tuleks võimalikult paindlikku meetodit, alustatakse projekti tegevust *Scrum* meetodi abil.

*Scrumi* järgi tuleb esmalt koostada toote tööde nimekiri, mille alusel planeeritakse kõiki edasisi tegevusi. Selleks, et katsetada ühe projekti raames võimalikult erinevate meetodite sobivust, otsustas autor kasutada *Scrum* lähenemist vaid ühe tsükli vältel. Samas hinnatakse selle kasutatavust ka teiste meetodite rakendamise juures. Tsükli alustamiseks soovitab meetod teha alguses pikem koosolek projekti tellijaga, et selgitada välja vajalik funktsionaalsus. Koosoleku kokkuvõtte on projektipäevikuna esitatud lisas 2 lk 63. Järgnev tabel 16 annab ülevaate nimetatud meetodi testimise aluseks oleva toote tööde nimekirjast.

**Tabel 16.** Uuritava projekti tööde nimekiri *Scrum* meetodi rakendamiseks (autori koostatud)

Töö nimetus	Töö selgitus
SA Ida- Virumaa Tööstusala Arenduse poolt tehtud küsimustiku lahtimõtestamine	Ida-Virumaal teostatud sarnase projekti idee ja olemuse väljaselgitamine.
Teiste töövahendusportaalide ideede väljaselgitamine.	Uurimine, milliste kriteeriumite alusel teised portaalid tööhuviliste andmeid koguvad.
Alternatiivsete lahenduste kaalumise ja ühe väljavalimine.	Selgitada välja optimaalseim viis töövahendusportaaali loomiseks.
Andmebaasi ülesehituse kujundamine.	Selgitada välja andmebaasi küsimustiku struktuur.
Andmete sisestamise funktsiooni loomine	Lua funktsionaalsus, mis võimaldab töösoovijatel oma andmeid sisestada.
Andmete analüüsitavuse funktsiooni loomine	Lua funktsionaalsus, mis võimaldab sisestatud andmeid filtreerida.
Testimine	Testida tehtud tööd.
Andmete sisestamise ja analüüsimise funktsiooni täiendamine.	Vastavalt saadud tagasisidele täiendatakse loodud funktsionaalsust.
Ettevõtjatega konfidentsiaalsuslepe sõlmimine, nendega suhtlemine.	Selgitamiseks välja ettevõtjate töövajadused, on vaja sõlmida nendega konfidentsiaalsuslepped.

*Scrum* meetodi tööde nimekiri ei ole fikseeritud järjestus teostatavatest tegevustest. Tegemist on esialgselt määratletud tegevustega, mille alusel teostatakse esimene

tsükkel, millest peab valmima ka esimene prototüüp. Esimesse tsüklisse valis projekti läbiviija kõik loetletud tegevused välja arvatud täiendustegevused ja ettevõtjatega suhtlemine. Esimese tsükli eesmärgiks oli välja selgitada põhisisedid, mida tööhuviliselt teada soovitakse saada.

Esmase funktsionaalsuse loomise juures andis projekti tellija vabad käed tehniliste lahenduste valimise juures. See tähendab seda, et alternatiivide seast tuleb valida välja parim võimalik lahendus ning tsükli lõpuks teha sellele tuginev prototüüp. Seejärel saab projekti tellijaga arutleda lahenduse sobivust projekti eesmärkide täitmisega ja planeerida edasisi tegevusi. Seega on esimene hinnang *Scrumi* kasutamiseks positiivne, kuna see ei nõua kriteeriumite fikseerimist ega kindla lahenduse kasutamist. Funktsionaalsuse loomise tehnilised lahendused on toodud järgnevas tabelis 17.

**Tabel 17.** Tehnilised lahendused andmete kogumiseks selgitustega (autori koostatud)

Tehniline lahendus	Selgitus
Eraldiseisev veebi-keskkond	Teha eraldi foorumilaadne keskkond, kus töösoovijad saavad tööpakkujaga suhelda, samas oleks andmebaas kättesaadav töötukassa administraatorile väljavõtete tegemiseks.
Excel vormis küsimustik	Teha lähtuvalt Ida-Virumaal tehtud küsimustikule sarnane ankeet ja korraldada andmete analüüs tabelprogrammis.
Veebipõhine küsimustik	Teha küsimustiku ankeet veebipõhiselt ning analüüsida andmeid tabelprogrammis.

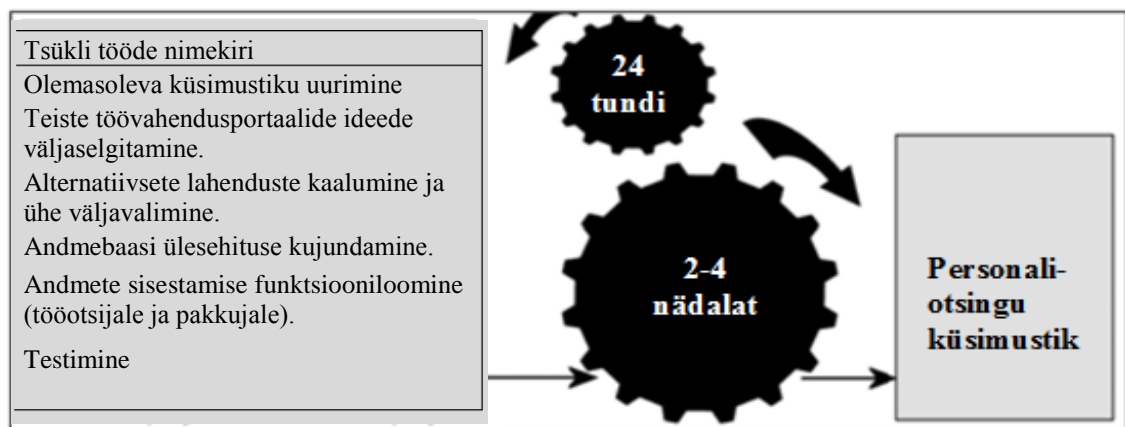
Tehniliselt kõige keerukam, kuid kõige rohkem funktsionaalsust pakkuvam lahendus oleks luua eraldiseisev veebikeskkond. See oleks sarnane teistele töövahendusportaalidele, kuid oleks Pärnumaa põhine ning andmed oleksid kättesaadavad töötukassa haldajatele. Samas oleks lahendus tülikas tavakasutajale ja nõuab rohkem lisapanust, kui näiteks lihtsalt küsimustiku täitmine. Foorumites suhtlemine võib muutuda kohmakaks ja passiivseks, mis tõrjuks eemale veelgi rohkem tööhuvilisi.

Exceli vormis küsimustiku peaks töösoovija laadima alla, selle täitma ja seejärel saatma selle kontaktisikule, kes informatsiooni ühisesse andmestikku kannab. Nimetatud lahendus võib tavakasutajale isegi põhjaliku juhendi abiga keerukaks või ebamugavaks osutada. Samas tähendab piisav keerukus ka seda, et tööst vähem motiveeritud isikud ei soovi asjaga tegeleda ning andmebaasi tekivad isikud, kes on valmis töökoha nimel



pingutama. Antud projekti raames see siiski ei sobi, kuna eesmärgiks on saada võimalikult palju huvilisi, kelle andmeid analüüsida ja anda ülevaadet sisenevatele ettevõtetele.

Parimaks lahenduseks hindas projekti teostaja teha veebipõhine uuring, kuna võimaldab andmete kiiret sisestamist. Saab teha eraldi küsimustikud töötajatele ja tööpakkujatele ning lihtsate tabeliprogrammi funktsioonide abil neid analüüsida ja kokku viia. Veebipõhise uuringu keskkonna valimisel lähtus autor ekstreemprogrammeerimise põhimõttest, mille järgi tuleb eelistada lihtsaimat lahendust. Projekti läbiviija jaoks osutus selleks keskkonnaks vabavarana kasutuses olev Google Docs, mis võimaldab küsimustikku jooksvalt muuta ja täiendada. Järgnev joonis annab ülevaate *Scrumi* esimesest tsüklist.



**Joonis 7.** *Scrum* meetodi testimine esmase funktsionaalsuse saamiseks (autori koostatud).

Esimese tsükli tulemusena valmis personaliotsingu küsimustik Google keskkonnas. Andmete analüüsile esialgu suurt rõhku ei pandud, kuna esmalt tuleb veenduda, kas tehtud töö on projekti tellija jaoks sobiv. Kuna kasutatud küsimustikukeskkond on lihtne, sai seda tegevuse käigus testida. Tehti valmis esialgne küsimustiku struktuur ning katsetati seda kolme inimese abiga. Kontrolliti andmete salvestatavust ning testiti nende filtreeritavust. Vastavalt testi käigus saadud ettepanekutele kohandati küsimustiku sõnastust. Vajadusel oleks saanud teha muutusi ka struktuuris.

Lähtuvalt *Scrumi* ideest peab tsükli lõpuks valmima toimiv toode, mis esitatakse projekti osapooltele. Valminud küsimustik esitati projekti tellijale ning tehtud töö vaadati üle projekti läbiviija, tellija ning administreerija osalusel. Koosoleku kokkuvõtte on esitatud lisa 2 lk 63 projektipäeviku osana.

Tsükli käigus valminud küsimustik on esitatud lisa 4 lk 70. Lähtuvalt tellija juhiseist püüti küsimustiku loomisel lähtuda Ida-Virumaal käivitatud andmebaasist. Arvestada tuli ka teiste veebipõhiste personalivahenduse keskkondade põhimõtteid, mis on välja toodud lisa 5 lk 75.

Ühise koosoleku käigus selgusid mitmed uuenduslikud ideed, mis muutsid struktuuri olulisel määral. Hetkeversioon nägi ette, et andmebaasi küsimustikud on koostatud lähtuvalt tööd otsivate ettevõtete soovidest (vt ka lisa 4 lk 70). See eeldas seda, et Pärnumaale sisenev ettevõtte annab andmebaasi administreerijale informatsiooni oma töövajaduse kohta, misjärel kohandatakse küsimustikku, täiendades seda vastava tööandja valdkonnaga. Seejärel kuulutatakse välja konkurss uute töökohtade täitmiseks. Koosoleku käigus selgus aga, et tegelikult ei ole vaja piirata tööpakkumisi sisenevate ettevõtjatega vaid töövõtjad võiksid saada sarnaselt tööotsijatele sisestada oma tööpakkumised. Kuna töötukassal on selline võimalus juba olemas ning nende esindaja hakkab ka seda andmestikku haldama, otsustati kasutada töötukassasse registreerinud ettevõteteid.

Seega tuli andmebaas ümber kujundada ning koguda inimeste andmeid mitte ettevõttepõhiselt vaid küsida sisendeid vastavalt nende tööootustele. Meetodi järgimine näitas, et *Scrum* on võimeline tulevasi muutusi arvestama ning reageerima vastavalt erinevate projektirollide soovidele. Eelkõige oli meetodi süsteemne lähenemine kasulik suhtlemisele ja tegevuste planeerimisel, kuna andis teadmise, et valminud prototüüpi saab ühiselt analüüsida. Põhimõtteliste vigade esinemisel oleks olnud võimalik planeerida uus tsükkel teistsuguste lahendustega. Traditsiooniliste või rangemate paindlike meetodite puhul oleks nende muudatuste tegemine olnud keerukam, kuna need eeldavad rohkem fikseerimist.

Autori hinnangul sobib *Scrum* kasutamiseks hästi koos ekstreemprogrammeerimisega. Kuna esimene tsükkel oli pigem uuriv ja analüüsiv ning tehnilisi ülesandeid oli vähe, ei olnud vajadust konkreetsete töövahendite järgi, mida teised meetodid pakuvad. Siiski oli abi ekstreemprogrammeerimise põhimõtetest, mille järgi tuleb eelistada alati lihtsaimat teed ning olla avatud alternatiividele. Nende ideede abil oli projektijuhil turvalisem valida lihtsaim vahend (Google Docs) ning samas olla arvestav ka teiste variantide võimalikkuses. Alternatiivide valimisel oli *Scrum* tervikuna väga tõhus, kuna võimaldas tegutseda ilma fikseeritud põhja olemasoluta. Kuna projekti tellija poolt ei olnud rangeid piiranguid tehniliste lahenduste osas ega olnud teada kindlad kriteeriumid, oli selle paindliku meetodi kasutamine eesmärgipärane.

*Scrum* meetodiga jätkamisel oleks tulnud sarnaselt esimesele tsüklile planeerida järgmise sammu käigus teostatavad tegevused. Need oleks hõlmanud eelkõige uute küsimuste planeerimist ning juba ka andmete analüüsitavusele süvenemist. Kuna juhtumiuuringu käigus soovitakse vaadelda erinevate meetodite sobivust, siis järgmise uuringutsükli ja prototüübi valmistamiseks lähtutakse teistest meetoditest.

Järgmise versiooni valmimiseks planeeriti vaadata, kui võrd hästi kasutatavad on võrdlemisi fikseeritud lähenemisega paindlikud meetodid nagu dünaamiline ja omaduspõhine tarkvaraarendusmeetod. Meetodi eripäraks on nõuete fikseerimine ennem tsükli alustamist. Käesoleva projekti raames võib meetodite järgimine tekitada probleeme just nõuete kinnitamise osas. Samas võib see anda meelegi ja vähendada ajakulu erinevate võimalike lahenduste analüüsimiseks.

Dünaamiline tarkvaraarenduse meetod pakub tööriistu ja põhimõtteid ning soovitab tsüklilist ja pidevalt täiendavat tegutsemist. Sarnast lahendust soovitab ka omaduspõhine tarkvaraarenduse meetod, mis tugineb oma tegevuses viiele põhiprotsessile. Järgmise funktsionaalsusastme loomiseks lähtutakse eelkõige nende kahe meetodi põhimõtetest ja töövahenditest.

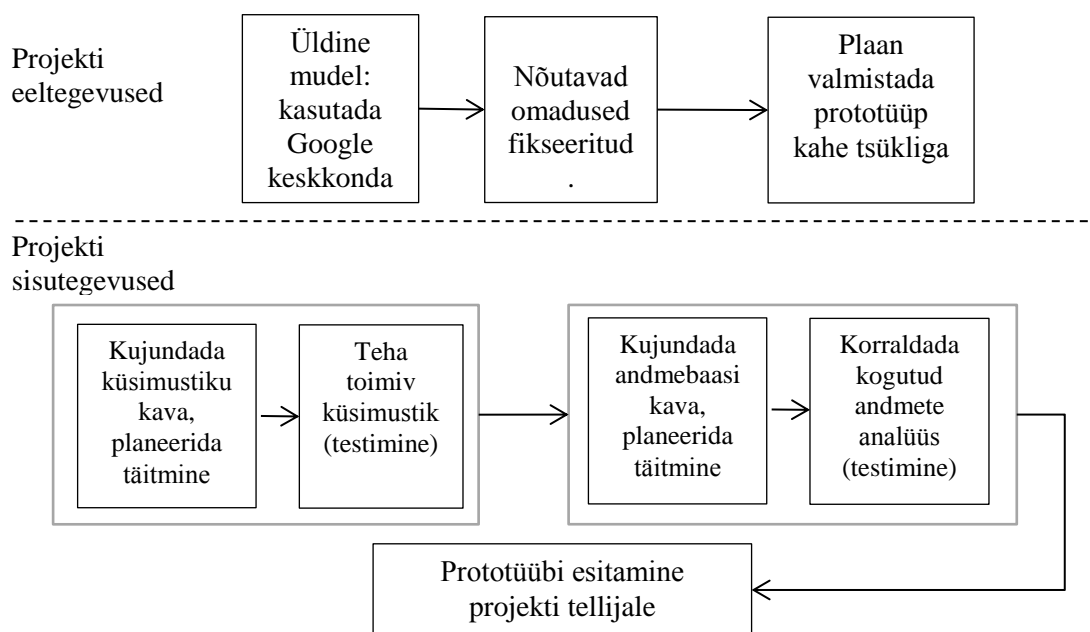
Mõlemad meetodid näevad ette, et riskide maandamiseks tuleb fikseerida põhinõuded, mis peavad jääma muutumatuks kogu projekti vältel. Projektitegevustes tuleb seega keskenduda põhinõuetele, vastuvõtlikkust muutustele tuleb rakendada aga

lisafunktsionaalsuse osas. Nende vajalikkus võib esile tulla pärast tsükli lõppu või ka selle vältel. Kuna dünaamiline tarkvaraarendusmeetod näeb ette aktiivset kasutaja-seotust, kiireid prototüüpide väljastamisi ja katsetamist, võib esineda mitmeid muudatusettepanekuid erinevate nõuete osas.

Järgmise prototüübi loomiseks määratleti järgmised muutumatud põhinõuded.

- Tööst huvitatud isik peab saama oma andmeid sisestada andmebaasi.
- Tööd pakkuv ettevõtte peab saama vajadusel ülevaadet tööst huvitatud inimestest.
- Tööd pakkuv ettevõtte peab saama ülevaadet inimestest, kes vastavad tema poolt seatud kriteeriumitele.
- Andmed peavad olema filtreeritavad.
- Andmed ei tohi olla avalikkusele kättesaadavad.

Põhinõuded selginesid esimese uuringutsükli lõpus toimunud koosoleku käigus, kus oma soove selgitas nii projekti tellija kui ka töötukassa esindajad. Uued nõuded erinesid algselt kavandatud nõuetest, mis *Scrum* meetodi kasutamisel vajalikuks peeti. Suurimaks erinevuseks oli tööpakkujapoolsete andmete mitteküsimisega. Joonis 9 toob ülevaate dünaamilise tarkvaraarenduse meetodi viie põhiprotsessi kasutamise kohta



**Joonis 8.** Omaduspõhise tarkvaraarenduse viis protsessi (autori koostatud).

Üldiseks mudeliks jäi jätkuvalt *Google* keskkonna kasutamine, kuna see võimaldas muutmisevõimalust pärast küsimustiku avalikustamist, jooksvat testimist ja tegemist oli vabavaraalse teenusega. Fikseeritud omadusteks olid eelnevalt nimetatud muutumatud põhiolemused. Juhtumiuuringu teine tegevustik, milleks on järgneva prototüübi valmistamine, teostati kahes etapis. Esmalt tegeleti küsimustiku disainimise ning selle koostamisega, seejärel andmebaasi ülesehituse planeerimise ja selle valmistamisega.

Küsimustiku kava erines esimesest prototüübist olulisel määral. Kaotati ära ettevõttepõhine lähenemine ja keskenduti ainult töösoovija andmete kogumisele. Sellest tulenevalt ei saanud enam töösoovijalt küsida, millisesse valdkonda ta soovib siseneda, sest tööpakkujateks on oluliselt suurem hulk ettevõtteid. Ettevõtjate arvukuse suurenemine tuleneb asjaolust, et enam ei otsita tööhuvilisi konkursipõhiselt vastavalt tehtud tööpakkumisele, vaid pakkujateks on kõik olemasolevad ja tulevased töötukassas registreerinud ettevõtted. Kuna inimeste huvid on väga erinevad ning ametite klassifikaator on loetelu jaoks liialt mahukas, otsustati küsida inimeste varasemat töökogemust ja hetkelist töösoovi ametipõhiselt. Kuna inimesed võivad sisestada ametinimetusi väga erinevalt, kasutades erinevaid kirjeldusviise ja lühendeid, ei oleks nende filtreerimine lihtsa andmestiku abil enam võimalik.

Eelnevalt esitatud prototüübi järgses koosolekus selgus, et töötukassa kasutab oma andmebaasides ISCO (*International Standard Classification of Occupations*) koodi. Töötukassa kasutab süsteemi, kus tööotsija andmed sisestab andmebaasi kliendi-teenindaja, kes ametite määratlemiseks kasutab programmisest otsingusüsteemi, kust valitakse sobiv tegevusvaldkond. Kuna kasutatav küsimustiku keskkond salvestab andmed *Google Form* vormingus, on andmed analüüsitavad eelkõige tabelprogrammide (nt *Excel*) abil.

Siin aga esines konflikt planeeritud projektietappide vahel. Algne kavand nägi ette, et planeeriti teostada küsimustik ning seejärel planeerida andmete analüüsitavus. Samas selgus küsimustiku planeerimise käigus, et oluliseks faktoriks on ka järgmine etapp, mis keskendus andmete analüüsitavusele. Konflikti oleks saanud ennetada põhjalikuma planeerimisega või siduda teostatavad tegevused omavahel nõu spiraaliga, kus neid üksteise järel pidevalt täiendatakse. Siiski võib tekkida olukord, kus liigne tegevuse

struktureerimine võib pärssida üldpildi jälgimist ja toob kaasa ootamatuid olukordi projekti mingis etapis, kus selgub, et eelmised tegevused tuleb ümber kujundada uue etapi tõttu.

Lähtuvalt tihedast projektietappide vastastikusest mõjust otsustati kaks etappi omavahel seondada. Tulemuseks saadi küsimustik, mis uurib küsitletava varasemat töökogemust ja töösoovi ISCO koodi põhjal. Kui töösoovija sisestab ise koodi, kasutades selleks statistikaameti klassifikaatorit (Ametite... 2014), mille abil saab otsingust leida oma ametile vastav kood. Probleemiks antud juhul on keerukus tavakasutaja jaoks, samas toetab see Ida-Virumaa näite hoiakut, kus piisava keerukusega hoitakse eemale asjast vähesel määral huvitatud isikud. ISCO koodi sisestamisel täitjate poolt oleks väiksem töökoormus ka töötukassa andmete haldajal, kes saab inimeste töösoove vastavalt koodile filtreerida. Küsimustiku testimine kõrvaliste isikute abil selgus aga, et koodi kasutamine on üsna keerukas ja vajab aega, et süsteemist aru saada.

Valminud prototüüp esitati tellijale, kes samuti nägi probleemi ISCO koodi määraltlemises. Tulevaste täienduste planeerimiseks tehti koosolek, kus projekti tellija ning elektroonikaettevõtte Oshino juht, Ingvar Kuusk, said esitada ettepanekuid muudatuste elluviimiseks. Selginesid mitmed asjaolud küsitavate objektide puhul ning ettevõtte juhilt saadi ka mitmeid tehnilisi ja sisulisi ettepanekuid. Otsustati kasutada küll ISCO koodi, kuid selle määratlemisega peab tegelema andmebaasi haldaja, töösoovijale on selle otsimine valikuline. Tehnilise poole pealt soovitati optimeerida kujundust ning integreerida Statistikaameti klassifikaatori valik küsimustikuga.

Toetudes tsükli käigus tehtud otsustele ja tegevustele saab järeldada, et suhtelised ranged meetodid nagu omaduspõhine ja dünaamiline lähenemine sobisid uuritava projekti teostamiseks. Nagu uuringu planeerimises kardetud, tekitas probleeme nõuete fikseerimine ja sellele toetudes tegevuse sammude planeerimine. Järgmise versiooni valmimiseks planeeriti kasutada erinevaid meetodeid üheskoos. Uurimistsükli vältel püüti hoiduda kindlate meetodite struktuurist kinnihoidmist ning vaadelda, milliseid meetodeid kasutatakse loomuliku valikuna.

Järgmise versiooni valmistamiseks püüti esmalt arvestada OÜ Oshino juhi soovitusel integreerida Statistikaameti ISCO koodi otsing küsimustiku sisenditega. Kuna *Google* keskkond sellist süsteemi ei toeta, prooviti kasutada *Limesurvey* keskkonda ning lisada vastavasisuline kood (*javascript*), mis võimaldab valida sobiv ametinimetuse alustades üldkategorias, liikudes edasi väiksemani.

Kuna projektijuhil ei ole oskused *Java* programmeerimiskeele kasutamises piisavalt head, otsustati kasutada lisaks abitööjõudu. Vastavasisuline kood kirjutati valmis kasutades ekstreemprogrammeerimise töövõtet paarisprogrammeerimine. Mõlemad projektiliikmed tegutsesid ühiselt koodi arendamisega ning tulemuseks saadi toimiv küsimus. Põhjalikuma analüüsi käigus selgus aga, et *Limesurvey* suurem funktsionaalsus ei kaalu üles selle tasuta konkurendi eripärasid. Projekti tellijaga koostöös otsustati jääda *Google* teenuste juurde, kuna see on tasuta ning projekti lõpphaldajale andmete opereerimiseks märgatavalt lihtsam.

Järgnevalt määratleti ära järgmise versiooni esitamistähtaeg, milleks oli päev enne Pärnumaa töömessi. Tegeleti küsimustiku sõnastuse parandamisega ja optimeerimisega. Põhiline sisutegevus oli andmete analüüsitavuse korraldamine. Otsustati kasutada *Exceli* makrode kirjutamise võimalust. Selle järgi pidi andmete haldaja kopeerima küsimustiku andmed ümber *Exceli* töölehele, mis oli valmis seatud andmete sorteerimiseks. Lihtsate andmete nagu töökohariigi ja palga sorteerimiseks sai kasutada lihtsaid filtreid. Valminud küsimustikus (vt ka lisa 6 lk 76) küsitakse ametinimetusi kolme tehtud töökoha ja kahe soovitud töövaldkonna kohta eraldi. Seetõttu salvestuvad need ka erinevatesse veergudesse, mis muutis filtripõhise lähenemise keerukaks ja tuli lihtsustada nende sorteerimist. Selleks otsustati kirjutada makro, mis võimaldab teostada otsingut erinevate isikute põhised ja mille abil saab sorteerida isikuid varasema töövaldkonna või soovitud töövaldkonna järgi vaid nupuvajutuse abil.

Kuna makrode loomisel oli vaja programmeerimisvahendit VBA (*Visual Basic for Applications*) ning projektijuhti kogemused nimetatud keeles ei olnud suured, oli ka siinkohal abi mitmest ekstreemprogrammeerimise töövõttest. Parimateks kasutatud vahenditeks oli varasemate versioonide alleshoidmine ja taastamisvõimalus, töö

väikesesammuline edendamine ning pidev testimine. Nende vahendite kasutamine võimaldas pidevat koodi täienemist ja programmeerija oskuste suurenemist.

Küsimustiku täitmist ja andmete analüüsitavust testiti prooviküsitluste abil ning selgus, et arusaadavuse osas probleeme ei olnud. Täiendatud küsimustik esitati projekti tellijale ja vältimaks tundlike isikuandmete ohtuseadmist, otsustati vältida sünniaja ja elukoha-aadressi küsimist. Osalemine töömessil võimaldas testida portaali toimimist lõppkasutajate peal ning ERR-i ja Pärnu Postimehe poolne meediakajastus aitasid kaasa selle avalikkuseni jõudmisele. Täiendavaks turundustegevuseks kasutati sotsiaalmeediat ning tehti reklaamribad, mida erinevatele lehekülgedele paigutada.

Töömessil esitatud andmebaasi struktuuri loetakse projekti lõppversiooniks. Kõik kasutatud meetodid aitasid kaasa lõpplahenduse valmistamisele. Kristallmeetodi põhimõtetest lähtuvalt tuli kasutada võimalikult paindlikku lähenemist. *Scrumi* ja ekstreemprogrammeerimise juhiste ja põhimõtete kasutamine sobis erinevate tegevuste teostamise jaoks hästi. Nagu kahtlustatud, osutusid piiravateks rangemad meetodid nagu dünaamiline ja omaduspõhine tarkvaraarendus. Samas pakkusid need kindlustunnet, et nõuded jäävad edaspidi muutumatuks ja võimaldasid keskenduda ainult konkreetsete tulemuste saavutamisele. Järgnev peatükk analüüsib tehtud tegevusi, otsuseid ja erinevate meetodite sobivust kokkuvõtvalt terve projekti raames.

## **2.4. Uuringu järeldused ja soovitused**

Uuringus läbiviimiseks kasutati tegevusuuringut, mille käigus uuriti teooriaosas uuritud meetodite sobivust uuritava projekti raames ning kirjeldati tehtud tegevusi ja otsuseid. Selle põhjal anti hinnang meetodi sobivusele ja kaalutleti alternatiivide kasutamise üle konkreetse tsükli raames. Järgnevalt tehakse järeldused erinevate teoorias kajastatud meetodite kohta uuritava projekti kontekstis, hinnatakse meetodi kasutatavust ja antakse soovitusi erinevate olukordadega toimetulemiseks.

Kristallmeetodid pakkusid välja, et uuritava projekti raames tuleks kasutada võimalikult paindlikku meetodit. Kogu projekti tegevusele tagasisivaatavalt võib öelda, et soovitus oli hea, kuna avatus muutustele oli oluline mitmel korral. Samas oli kasu ka teatud



omaduste fikseerimisest, mis andis projekti läbiviimiseks kindluse tegeleda mingi funktsiooni loomisega. Selle tõttu ei pidanud kartma, et tehtud tegevused osutavad tühiseks mingisuguse suurema muutuse tõttu. Kristallmeetod ei andnud konkreetset tegevuskava konkreetse projektitüübi teostamiseks. Selle asemel aitas see määratleda selle keerukust ja raskust ning valida selle põhjal kas paindlikumate või traditsioonilisemate meetodite vahel. Lisaks andis meetod oma põhimõtetega tuge erinevate projektietappide teostamisel.

Meetod on lihtsasti rakendatav ka teistele sarnastele projektidele. Väikesemahulise projektijuht saab selle abil määratleda projekti raskust/keerukust ning selle põhjal otsustada, milliseid meetodeid kasutama peaks. Konkreetse meetodi rakendamisel peaks seega lähtuma nii kristallmeetodi perekonnale omastest kui ka teostatava projekti tüübi jaoks valitud meetodi põhimõtetest.

Dünaamilise tarkvaraarenduse meetodi juhiseid otsustati kasutada integreerituna omaduspõhise tarkvaraarenduse meetodiga. Otsus tulenes asjaolust, et teooriaosas selgus mõlema meetodi sarnasus ning seetõttu ühendati dünaamilise meetodi põhimõtted omaduspõhise meetodi struktuuriga. Selline meetodite segunemine oli uuritava projekti raames rakendatav, kuna tegemist oli suhteliselt väikesemahulise arendustegevusega. Seetõttu oli parem ülevaade erinevatest situatsioonidest ning sai hinnata mõlema meetodi sobivust.

Nii dünaamiline kui ka omaduspõhine meetod paneb suurt rõhku eeltegevustele, milleks on põhikriteeriumite määratlemine. Projekti teostamise teise tsükli raames oli meetod sobilik ning sai fikseerida teatud nõuded ja tingimused. Tingimuste fikseerimine oli positiivse mõjuga projekti tegevuste suhtes, kuna võimaldas tegutseda täielikult funktsionaalsuse loomisega. Kui kindlad oodatavad tulemused ei oleks teada olnud, oleks olnud suurem sõltuvus projekti tellijast, kuna oleks tulnud pärida pidevat tagasisidet tehtud otsuste õigsuses. See pidev hinnangu ootamine ja kohandumine aga võib projekti ajalises mõttes venitada.

Projekti tegevuses tekkis pärast omaduspõhise ja dünaamilise meetodi kasutamist mitmed muutused, millega kohanemine ei oleks rangelt fikseeritud põhinõuete

kasutamisel võimalik olnud. Fikseeritud funktsioonidega saavutatud maandatud riskitase ei oleks terve projekti vältel väärt allumatust muutustele. Ärilise projekti puhul võib muutustest keeldumisele projekti teostajale kahjulikuks osutada. Seda eriti juhul, kui tegemist on väiksemahulise tarkvaraprojektiga, sest sellisel juhul võib muutustest keeldumine jääda arusaamatuks ka projekti tellijale. Tulemuseks võib olla rahulolematuse projekti tulemustega ning projekti teostaja maine langus. Kui aga tegemist oleks keerukama või suurema projektiga, on põhinõuete fikseerimine kasulik, kuna sellisel juhul ei pea kartma, et võivad toimuda muutused olulistes juba tehtud tegevustes.

Projekti algfaasis oleks fikseeritud nõuete kasutamine osutunud aga ebaotstarbekaks. Selle asemel, et esitada mingisugune prototüüp, mis sõltub projektijuhi nägemusest ja projekti tellija juhistest, oleks pikalt ja põhjalikult tulnud planeerida juba lõpplahendust. Projekti käigus toimusid aga mitmed põhimõtteid muutvad otsused, mis näitab seda, et *Scrumi* suhtluspõhine lähenemine projekti algfaasis oli sobilikum, kui võrdlemisi traditsioonilisemad tegevuse planeerimine vahendid.

Dünaamilise tarkvaraarenduse meetodi poolt väljapakutud töövahendid nagu fikseeritud ajaraamistiku koostamine, lihtsustatud töötoad või PVSE, ei olnud uuritava projekti raames otstarbekad. See tulenes asjaolust, et projekt oli olemuselt liiga ettearvamatu, et määratleda kindlalt tähtsate või kestvusi, fikseerimaks nõutud tulemuste soovitud aeg. Lihtsustatud töötoad on sobilikud suurema arvu huvigruppidega tegelemisel ning väikese projekti raames piisab koosolekute korraldamisest. PVSE töövõtte poolt pakutud idee määratlemaks erinevaid funktsioone vastavalt sellele, kas neid on vaja, need võivad olla või ei peaks olema, ei olnud otstarbekas. Funktsionaalsuse loomisel lähtuti vaid vajaminevatest omadustest ning mittevajalike või võib-olla teostatavate omaduste loetlemisel ei olnud mõtet.

Dünaamilise või omaduspõhise tarkvaraarenduse meetodit peaks projektijuht kaaluma olukorras, kus tegemist ei ole väikeprojektiga või kui selle teostamisega seonduv risk on suur. Sellisel juhul on kasu kindlastest kokkulepetest, millele terve projekti vältel tugineda saab. Väikesemahulise, madala riskiga projekti jaoks ei ole otstarbekas näidata üles paindumatust, kuna tulevikus tehtavad muutused põhiomaduste osas võivad olla

hädavajalikud selle õnnestumiseks. Vajadusel on aga võimalik kombineerida dünaamilise ja omaduspõhise meetodi fikseeritud lähenemist teiste meetoditega. Sellisel juhul tuleks esimeste prototüüpide valmistamisel olla avatud igasugustele ettepanekutele ja muudatustele. Hilisemas faasis võib aga fikseerida kindlad tehtud ja tulevikus teostatavad nõuded. Sellisel juhul ollakse võimalikult paindlik muudatuste osas tarkvaralise lahenduse kujundamise osas, kuid maandatakse riskid hilisemate ootamatute muudatuste osas.

Kuna projekti alguses ei olnud teada, milliseid vahendeid andmebaasi loomiseks kasutada, otsustas tegevusuuringu läbiviija kasutada *Scrumi*. Meetod pakub paindlikkust ning samas annab konkreetseid juhiseid prototüüpide väljastamiseks. Eriti kasulikuks osutus meetodi poolt välja pakutud pikem planeerimine projekti tellija ja teostaja vahel, kus selgitati välja üldine idee ja oodatavad tulemused. Sellele tuginedes sai alustada tegevusega ning luua esmane funktsionaalsus. Prototüübi valmimisel soovitas *Scrum* teha järgnev kokkusaamine projekti tellijaga ning määratleda vajalikud muutused ja järgmise etapi tegevused.

*Scrum* meetod oleks uuritava projekti raames olnud kasutatav terve projekti elutsükli vältel. Pärast esmase funktsionaalsuse loomist selgitati välja järgmise sammu vajaminevad tegevused. Selliselt tegutsedes oleks tagatud pidev avatus muudatustele ning kiire reageerimine projekti tellija soovidele. Programmeerimistegevuse teostamiseks tulnuks kasutada sellisel juhul ka teisi meetodeid, mis pakuvad tööriistu ja juhiseid selle tegemiseks.

Meetodi põhimõtted ja struktuur oleks kasutatav ka teiste väikesemahuliste projektide puhul. Kuna nimetatud meetodi tsüklil peab lõppema toimiva prototüübina, on see sobilik ka olukordades, kus ei ole teada lõplik väljastamise kuupäev. Sellisel juhul saab toodet täiendada ning arendada vastavalt selle tellija nõudmisele ning vajadusel saab tellija esitleda juba teatud funktsionaalsuse saavutanud toodet. Kirjeldatud meetodit kasutatakse näiteks videomängutööstuses, kus klientidele müüakse poolikuid kuid samas toimivaid varajase ligipääsu (*early access*) tooteid. Konkreetsete tegevuste teostamisvahendite valimisel jätab *Scrum* otsustusvõimaluse projektijuhile. Vastavalt meeskonnaliikmete vajadustele ja nägemusele saab programmeerimis- või kujundamis-

tegevuse teostamiseks kasutada erinevaid juhendeid. Lisaks on võimalus kombineerida meetodi üldstruktuuriga mingi teise meetodi põhimõtted, mis pakuvad rohkem suuniseid konkreetsete tegevuste teostamiseks.

Programmeerimistegevust oli uuritava projekti raames võrdlemisi vähe. Kirjutati *javascript* kood, mis lõplikku kasutust ei näinud ning kirjutati mõned *Exceli* makrod, mille abil filtreerimistegevust korraldati. Programmeerimistegevuse teostamisel oli enam abi ekstreemprogrammeerimise põhimõtetest ja töövõtetest. Kiire tagasisidestamise kasutamine hoidis ära ebavajalikud tegevused ning informeeris osapooli situatsioonidest, mis paremini oleks võinud olla. Lihtsuse eeldamine oli samuti väga tõhus, kuna hoidis ära liigse ajakulu erinevate lahenduste kaalumisele. Samas oldi avatud alternatiivsete lähenemiste suhtes, mis võimaldas teha informeeritud otsuseid.

Ekstreemprogrammeerimise töövõtted on sobivad eelkõige tarkvaraarendusprojekti, kus toimub mahukas koodi kirjutamine ning programmeerijad paiknevad ühes hoones. Tänapäeval toimub aga üha rohkem tegevusi kaugtööna ning väikesemahuliste projektide puhul ei pruugi alati olla otstarbekas luua eraldi töökeskkonda. Kuigi seeläbi on informeerituse tase kõrgem ning toimub pidev oskuste täiendamine, on väikeprojektide raames võimalik suhtlust korraldada ka teiste kanalite, kui otsesuhtlemise kaudu.

Erinevate meetodite katsetamine andis ülevaate nende sobivusest uuritava projekti raames. Hinnati meetodite juhistest tehtud tegevusi ja nende sobivust ja selgust, et teatud tingimustel on näiliselt väga erinevad meetodid siiski väikesemahuliste projektide loomiseks sobilikud. Konkreetse meetodi valik sõltub projektijuhi riskitundlikkusest ning avatusest muutustele.

Uuritava väikesemahulise projekti juures olid sobivaimateks meetoditeks *Scrum* ja ekstreemprogrammeerimine, mis võimaldasid avatud suhtlemist ning andsid piisavalt juhiseid. Nende universaalsust ja kohandamisvõimet arvestades on meetodid sobilikud teistegi väikesemahuliste projektide teostamisel. Lisaks on meetodite konkreetsed ja paindlikkusele suunatud juhised abiks projektijuhtidele, kellel ei ole suurt varasemat töökogemust. Erinevaid meetodeid on võimalik omavahel kombineerida ja siduda teiste

meetoditega, saades tulemuseks vastavalt kas paindlikuma või traditsioonilisema projektijuhtimismeetodi.

Eeltoodud ülevaade on juhiseks ja üldpõhimõteteks, mida projektijuht väikesemahuliste tarkvaraprojektide puhul jälgima peaks. Soovituste järgimine lihtsustab meetodi valimist ning annab ülevaate nende kombineeritavuse kohta. Järgnev loetelu annab koondülevaate nõuannetest, mida projektijuht jälgima peaks:

- hinda projekti ligikaudne maht;
- kaalu riski ja määratle, kuivõrd muudatustele avatud saad olla;
- arutle projekti tellijaga, kuivõrd fikseeritud on väljastamistähtajad prototüüpidele ja lõpptootele;
- planeeri peamine töökeskkond (kas virtuaaltöö või kontorikeskkond);
- selgita välja erinevate meetodite põhimõtted.

Projekti mahu hindamisel selgub, kas teostatav projekt on väikesemahuline või suurem. Muudatustele avatuse ja riskikartlikkuse hindamine annavad üldise arusaama sellest, kuivõrd plaanipõhine peab projektitegevus olema. Paremat planeerimisvõimalust pakub ka kindlate tähtaegade teadmine. Projektiliikmete töökeskkonnast lähtuvalt saab kaaluda erinevate töövõtete nagu näiteks ülesandekaartide kasutamist. Tulenevalt väljaselgitatud kriteeriumitest, aitab sobilikke meetodeid valida järgnev tabel 18.

**Tabel 18.** Erinevate meetodite eelistus vastavalt kriteeriumitele (autori koostatud)

Kriteerium	Kristall-meetodid	Dünaamiline tarkvaraarendus	Omaduspõhine tarkvaraarendus	Scrum meetod	Ekstreem-programmeerimine
Ebakindel ajakava vahe-tähtaegade, koosolekute jm osas	+			+	+
On teada kindel, muutumatu lõppfunktsionaalsus	+	+	+		
Projektiliikmed töötavad ühes ruumis		+	+		++
Suur rõhk koodi kirjutamisel					++
Vaja maksimaalset paindlikkust	+			++	+
Vaja vältida riski/muutust		++	++		
Väike maht	+			+	+
Lõpptähtaeg ei ole teada	+			++	+

Tabelis on vastavalt kriteeriumile märgistatud sobilikumad meetodid „+“ märgiga. Meetod, mis on eriliselt sobilik vastale kriteeriumile, on märgistatud „++“ märgiga. Projekti läbiviimisel tuleb lähtuda eelkõige konkreetsest situatsioonist, hinnata selle suurust ja riskikartlikkust. Teoreetiline teemakäsitus ja tegevusuuring näitasid, et väikesemahulise tarkvaraprojekti loomisel tuleks eelistada võimalikult paindlikku lähenemist. Uuringust selgus, et erinevate meetodite kombineerimine on võimalik ja eriti kasulikuks võib osutuda rangete meetodite sissetoomine hilisemates faasides. Kui projekt ja üldine riskikartlikkus ei luba tegevuste liigset voolavust, tuleb eelistada rangemaid lähenemisi. Tabeli abil on võimalik välja selgitada sobilikud meetodid ning algatada tegevust kas lähtuvalt ühe lähenemise põhimõtetest või neid kombineerides.

Uuringu tulemused ei ole konkreetselt seotud alalise organisatsiooniga vaid keskenduvad projektidele ja nende juhtimisele. Eelkõige on uuringu tulemused kasulikud väikeprojektide juhtidele, keda ootavad ees muutliku loomuga projektid. Lisaks on tulemused kasulikud ka väikeste ettevõtete projektijuhtidele, kes võivad kokku puutuda tarkvara arendusprojektidega. See, kuidas alalise organisatsiooni olemasolu mõjutab erinevate meetodite sobivust ja efektiivsust, jääb käesoleva lõputöö ulatusest välja ning on ettepanekuks järgnevate uuringute teostamiseks.

## KOKKUVÕTE

Käesoleva diplomitöö eesmärgiks oli katsetada erinevaid meetodeid väikesemahuliste tarkvaraprojektide juhtimiseks ning anda juhised sarnaste projektide läbiviimiseks. Eesmärgi täitmiseks viidi läbi tegevusuuring, mis viidi läbi näidisprojekti abil, milleks oli töövahendusportaali loomine Pärnumaal. Tegevusuuringu käigus anti ülevaade erinevate meetodite sobilikkusest uuritava tarkvaraprojekti raames ning anti hinnang nende sobilikkusest teiste sarnaste projektide teostamiseks.

Väikesemahulised tarkvaraprojekti määratlus sõltub suures osas kontekstist. Projekti läbiviija peab arvestama oma organisatsiooni suurust ja töötajate arvu ning võrdlema seda teostatava projekti ressursinõudlikkusega. Väikesemahuline tarkvaraprojekt on selle läbiviija jaoks teostatav suhteliselt väheste vahenditega. Lisaks on selle tegevused eesmärgistatud, ajaliselt piiratud ja konkreetse probleemi lahendamiseks kasutatakse tarkvaralisi lahendusi.

Traditsioonilised projektijuhtimismeetodid on ametlikud, fikseeritud tegevusplaaniga ja suhteliselt bürookraatlikud. Eelkõige on need sobilikud mahukate projektide elluviimiseks, kus eeltingimused on suures osas planeeritavad ja tegevuste teostamisega seondub arvestatav risk. Nüüdisajale omase arvutitööstuse pideva arenguga toimub aga pidevaid muutusi ja täiendusi, mis projekti kulgu mõjutada võivad. Eriliselt suur on muutuste oht tarkvaratööstuses, kus täienevad pidevalt tehnilised võimalused ja suureneb kasutajate nõudlus. Muutustega kaasas püsimiseks on töötatud välja paindlikud projektijuhtimismeetodid, mis pakuvad põhimõtteid ja töövahendeid, mille abil teostada tegevusi ning samas säilitada avatus muudatustele. On loodud väga palju paindlikke meetodeid, kuid lõputöö ulatust silmas pidades uuriti kirjanduses enim viidatuid.

Uuritud paindlikud meetodid lähtusid Paindliku projektijuhtimise manifestist ja pakkusid välja järgitavad põhimõtted või tööriistad. Kristallmeetod soovib määratleda projekti suuruse ning vastavalt sellele kasutada paindlikumat või traditsioonilisemat lähenemist. Dünaamiline ja omaduspõhine tarkvaraarenduse meetod suunavad projektijuhti vähendama paindlikkusest tulenevat riski. Nende meetodite järgi tuleb panna suurt rõhku tegevuse algfaasis ja fikseerida teatud põhiohused, mille alusel planeeritakse samm-sammulist ja pidevalt täiendavat arendustegevust.

Terviklikult paindlikku lähenemist soovitas *Scrum* ja ekstreemprogrammeerimise meetod. *Scrum* pakkus välja tegevusstruktuuri, mille järgi prototüüpide valmistamist planeerida. Ekstreemprogrammeerimine on mõeldud rohkem konkreetsete arendustegevuste nagu koodikirjutamise või disainimise jaoks, pakkudes välja töötehnikad, mida järgides tagada avatus muutustele ja arenev töökeskkond.

Tegevusuuringu raames uuriti näidisprojekti ning selle läbiviimiseks planeeriti erinevate meetodite kasutamist ja anti hinnang nende sobivusele. Uuringu tegevus lähtus tsüklilisest mudelist, mille osadeks oli planeerimine, tegutsemine ja vaatlemine ning peegeldamine. Uuringu raames selgus, et vähemal või suuremal määral sobisid kasutamiseks kõik uuritud meetodid. Kristallmeetodid soovitasid valida võimalikult paindlikum meetod ja seda kinnitas ka uuring. Paindlikumad meetodid nagu *Scrum* ja ekstreemprogrammeerimine pakkusid vabadust otsustamiseks ning samas andsid kindlaid juhiseid otsuste tegemiseks ja tegevuste elluviimiseks.

Rangemad meetodid nagu dünaamiline ja omaduspõhine tarkvaraarendus olid sobilikud uuritava projekti raames, kuid ainult hilisemates etappides. Kindlate kriteeriumite määratlemine ei oleks tegevuse alustamisel osutunud otstarbekaks ning oleks põhjustanud ebavajalikke toiminguid. Lisaks tekitas probleeme kindla tsüklite struktuuri määratlemine, kuna põhjalik etteplaneerimine eeldab piisavat informatsiooni hulka. Uuritava projekti käigus aga selgus, et kui põhiohuseid on võimalik fikseerida, siis tegevuste samm-sammuline planeerimine on keerukas.

Teiste sarnaste projektide puhul soovib autor uurida erialast kirjandust projektijuhtimise kohta ja tutvuda erinevate meetodite võimaluste kohta. Konkreetse



meetodi valimisel tuleb lähtuda projekti läbiviija hinnangust selle suurusele ning riskantsusele. Tuleb välja selgitada kuivõrd rangelt on fikseeritavad lõpptulemused ning tähtjad ja vastavalt sellele valida sobivaim meetod. Kui viiakse ellu mitmeid erineva suurusega projekte, on hea kasutada kristallmeetodite jaotust, mis aitab selgitada, kuivõrd paindlikku lähenemist tasub kasutada. Projektijuht peab kaaluma kuivõrd suure riskiga on konkreetne projekt seotud ning vastavalt sellele valida meetodi.

Madalama riski puhul sobib *Scrum*, mida saab kergesti kombineerida ekstreemprogrammeerimisega. Suurema riski puhul aga soovitatakse kasutada dünaamilist tarkvaraarenduse meetodit ja selle tööriistu või omaduspõhise tarkvaraarenduse meetodit. Kui väikesemahulise tarkvaraprojektiga seondub suur risk ning kõiki põhinõudeid ei saa projekti eeltegevuste käigus fikseerida, on võimalik kombineerida *Scrum* ja ekstreemprogrammeerimine teiste meetoditega. Sellisel juhul tuleks projekti käimasaamisel eelistada võimalikult paindlikku (näiteks *Scrum*) lähenemist. Põhisuuna selgumisel võib aga ka üle minna dünaamilisele või omaduspõhisele lähenemisele ja fikseerida edasised teostatavad tegevused.

## VIIDATUD ALLIKAD

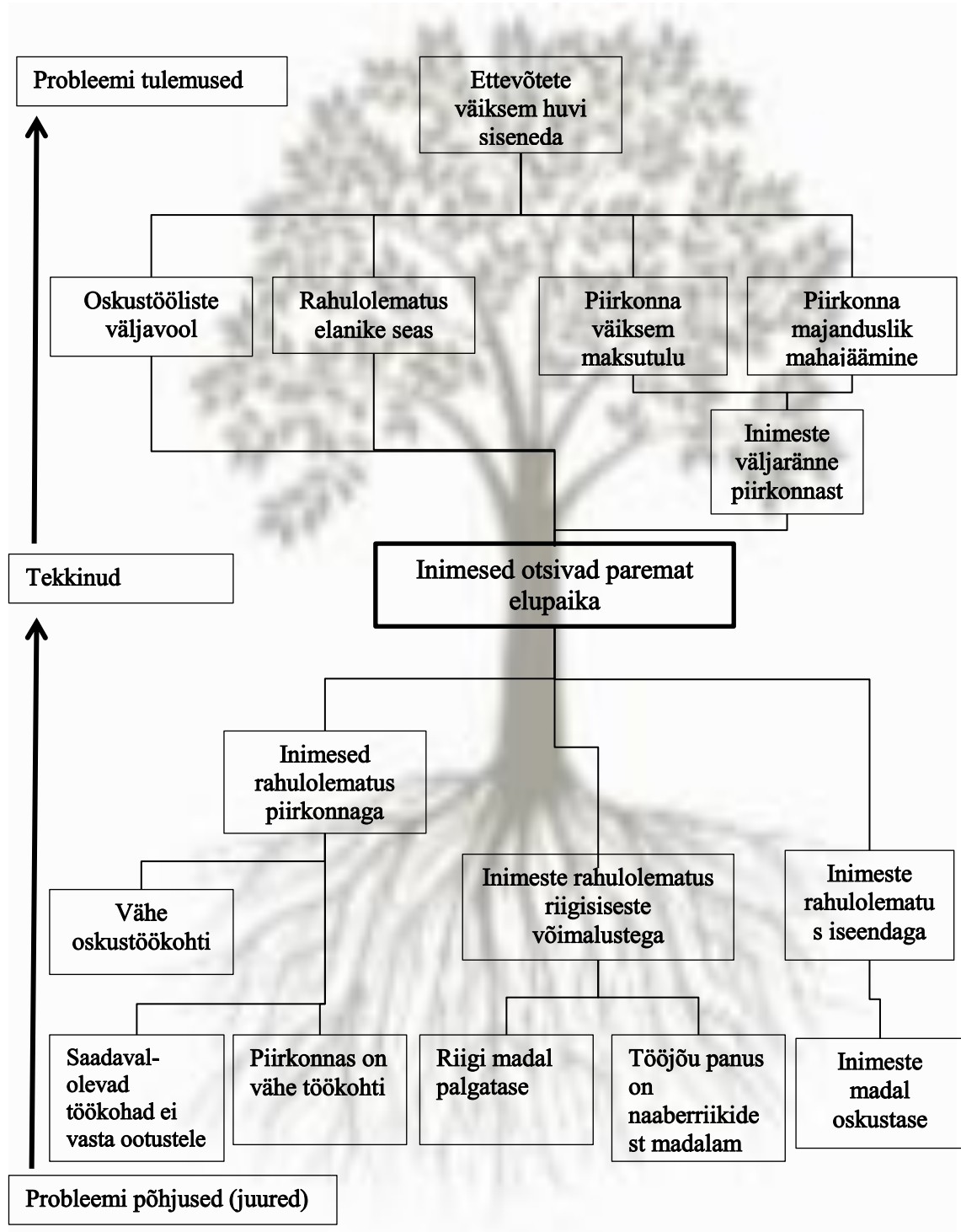
1. **Abrahamsson, P., Oza, N., Siponen, M. T.** 2010. Agile Software Development Methods: A Comparative Review. - Agile Software Development pp 31-59.
2. Ametite klassifikaator 2008v1.5b. Eesti Statistikaamet  
[[http://metaweb.stat.ee/view\\_xml.htm?id=3005499&selectedRow=0&siteLanguage=ee](http://metaweb.stat.ee/view_xml.htm?id=3005499&selectedRow=0&siteLanguage=ee)] 13.03.2014
3. **Artto K., Martinsuo M., Kujala J.,** 2011. Project business. Helsinki.
4. **Beck, K.** 2000. Extreme Programming Explained: Embrace Change. Addison-Wesley.
5. **Carr, W., Kemmis, S.** 1986. Becoming Critical - Education, Knowledge and Action Research. New York: Taylor & Francis Group.
6. CDO = Cambridge Dictionaries Online.  
[[http://dictionary.cambridge.org/dictionary/american-english/project\\_1?q=project](http://dictionary.cambridge.org/dictionary/american-english/project_1?q=project)] 15.02.2014.
7. **Cockburn, A.** 2003. DSDM: Business Focused Development, second ed. Vastutav toim Stapleton, J. Harlow: Pearson Education.
8. **Cockburn, A.** 2004. Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams. Boston: Addison-Wesley Professional.
9. **Cockburn, A.** 2006. Agile Software Development: The Cooperative Game. Boston: Addison-Wesley Professional.
10. Commission Recommendation of 6 May 2003. concerning the definition of micro, small and medium-sized enterprises. - Official Journal of the European Union, 06.05.2003, dok nr C(2003) 1422.
11. **Dybå, T., Dingsøy, T.** 2008. Empirical studies of agile software development: A systematic review. Information and Software Technology Vol. 50 pp 833-859.
12. EKSS = Eesti keele seletav sõnaraamat.  
[<http://www.eki.ee/dict/ekss/index.cgi?Q=projekt&F=M>] 15.02.2014.
13. **Elliot, J.** 1991. Action Research for Educational Change. Philadelphia: Open University Press.

14. **Engels, G., Sauer, S.**, 2010. A Meta-Method for Defining Software Engineering Methods. - Lecture Notes in Computer Science, Vol 5765, pp 411-440.
15. **Finkelstein, A., Kramer, J.** 2000. Software engineering: a roadmap. Proceedings of the Conference on The Future of Software Engineering. New York: ACM Press.
16. **Ford, G.** 1990 SEI Report on Undergraduate Software Engineering Education - Software Engineering Curriculum Project. Pennsylvania: Software Engineering Institute.
17. **Holmström, H., Fitzgerald, B., Ågerfalk, P. J., Conchúir, E. O.** 2006. Agile Practices Reduce Distance in Global Software Development. – Information Systems Management, Vol 23 (3) pp 7-18.
18. **Hunt, J.** 2006. Wiltshire: Agile Software Construction. Wiltshire: Experis Ltd.
19. Ida- Virumaa tööstusalad. SA Ida- Virumaa Tööstusalade Arendus. Ankeet [<http://ivia.ee/rabota/ru>] 15.02.2013.
20. IEE Standard 610.12.1990, IEEE Standard Glossary of Software Engineering Terminology.
21. **Jepsen, L. O., Mathiassen, L., Nielsen, P. A.** 1989. Back to Thinking Mode: Diaries for the Management of Information Systems Development Projects. - Behaviour and Information Technology, Vol 8 (3), pp 207-217.
22. **Kemmis, S. and McTaggart, R.** 2000. Participatory action research. - Handbook of Qualitative Research (2nd Ed.) Vastutav toim Denzin, N., Lincoln, Y. California: Sage Publications, pp 567-605.
23. **Kuura, A., Blackburn, R., A., Lundin, R., A.** 2013. Entrepreneurship and projects- Linking segregated communities. – Scandinavian Journal of Management, Vol 30 (2), pp 214-230.
24. **Lin, Y., Liu, S.**, 2011. Small and large projects: their dynamics and which ones to take. – Kybernetes, Vol 40 (9) pp 1354-1372.
25. **Lundin, R. A., Söderholm, A.** 1995. A theory of the temporary organization. – Scandinavian Journal of Management, Vol 11 (4), pp 437-455.
26. Manifesto for Agile Software Development. [<http://agilemanifesto.org/>]. 01.02.2014.

27. **Meyer, B., Nawrocki, R., J., Walter, B.** 2007. Balancing Agility and Formalism in Software Engineering. Central and East European Conference on Software Engineering Techniques - konverentsi kokkuvõte. Poznań: Springer.
28. **Münch, J., Kowalczyk, M., Ambrust, O., Soto, M.** 2012. Software Process Definition and Management. Heidelberg: Springer-Verlag.
29. **Nielsen, P., A.** 2007. IS Action Research and its criteria. – Information Systems Action Research – An Applied View of Emerging Concepts and Methods. Vastutav toim Kock, N. Texas: Springer Science+Business Media, pp 354-375.
30. **O’Leary, Z.** 2004. The Essential Guide to Doing Research. London: Sage Publications.
31. **O’Sheedy, D., G.** 2012. A study of agile project management methods used for IT implementation projects in small and medium-sized enterprises. DBA thesis, Southern Cross University, Lismore, NSW.
32. **Pettigrew, A., M.** 1990. Longitudinal Field Research on Change: Theory and Practice.- Organization Science, Vol 1 (3),pp 267-292.
33. **Probhaker, A.** 2006. Dynamic Solutions Delivery Model (DSDM). - Journal of the Quality Assurance Institute, Vol 20 (3), pp 29-32.
34. Project. Oxford English Dictionary.  
[<http://www.oed.com/view/Entry/152265?rskey=aDrD7K&result=1&isAdvanced=false#eid>] 15.02.2014.
35. **Reason, P., & Bradbury, H.** 2008. Handbook of Action Research: Participative inquiry and practice 2nd edition. London: Sage Publications.
36. **Rossberg, J.** 2008. Pro Visual Studio Team System Application Lifecycle Management. New York: Springer-Verlag New York.
37. **Rowe, S. F.** 2007. Project Management for Small Projects. Vienna: Management Concepts Inc.
38. **Schwaber, K.** 2007. The Enterprise and Scrum. Redmond: Microsoft Press.
39. **Shames, S., Bernier, Q., Masiga, M.** 2013. Development of a participatory action research approach for four agricultural carbon projects in east Africa. - CAPRI Working Paper nro113. Washington, D.C.: International Food Policy Research Institute. [<http://dx.doi.org/10.2499/CAPRIWP113>]

40. Software Engineering. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. A Volume of the Computing Curricula Series. 2004. The Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery.
41. Software engineering. Oxford English Dictionary.  
[<http://www.oed.com/view/Entry/183938?redirectedFrom=software+engineering#eid21971735>] 15.02.2014.
42. **Sommerville, I.** 2011. Software Engineering Ninth Edition. Massachusetts: Pearson Education, Inc.
43. **Stapleton, J.** 2007. DSDM Business Focused Development. Harlow: Pearson Education Limited.
44. **Stoica, M., Mircea, M., Ghilic-Micu, B.** 2013. Software Development: Agile vs. Traditional. – Informatica Economică, Vol. 17 (4), pp 65-77.
45. **Stringer, E. T.** 2004. Action Research for Educators. Lanham: Rowman and Littlefield.
46. **Stuart, J., Morojele, M., Lefoka, P.** 1997. Collaborative classroom action research in Lesotho. Vastutav toim Crossley, M., Vulliamy, G. New York: Routledge, pp 161-198.
47. **Turner, J. R., Müller, R.** 2003. On the nature of the project as a temporary organization. - International Journal of Project Management, Vol 21 (1) pp 1-8.
48. **Winter, R., Munn-Giddings, C.** 2001. A Handbook for Action Research in Health and Social Care. New York: Routledge.

Lisa 1. Pärnu töökeskkonda iseloomustav probleemipuu



## Lisa 2. Projektipäevik

Järgnevalt kajastatakse projekti päevik. Iga suurema tegevuse või sündmuse järel on autor teinud kokkuvõtte. Kuupäev tähistab kokkuvõtte tegemise kuupäeva, mille ajaks oli tegevus tehtud.

21.10.2013

Projektipraktikat teostades uurisin, kas on mingisugune probleem või olukord, mida saaksin oma lõputöö raames uurida ja lahata. Sain vastuseks, et on võimalik aidata kaasa lahenduse leidmisele tööjõu väljarände peatamiseks Pärnumaal. Selleks tuleb luua töövahendusportaal, mis võimaldab tööpakkujatel näha, palju on tööst huvitatud inimesi. Töösoovijatel on võimalik ennast registreerida andmebaasi. Portaali loomise eelduseks on tööstusküla arendamine, mille raames soovivad Pärnu töömaastikku siseneda juba seitse ettevõtet. Samas ei ole neil teada, kas ja millises mahus on olemas sobiv tööjõud. Sarnane projekt viidi läbi Ida-Virumaal, kus konkursi raames paluti töösoovijatel täita *Excel* failis olev küsimustik.

5.02.2014

Sain kinnituse, et tööstusküla rajamine sai kinnituse. Seega läheb vaja ka personaliotsingu andmebaasi loomist. Alustasin uurima erinevaid lahendusi. Võimalikeks variantideks on eraldiseisva veebikeskkonna loomine, veebipõhise küsimustiku loomine ning sarnaselt Ida-Virumaa näitele, *Excel* küsimustiku loomine. Koos PEAKi esindajaga arutledes otsustati kasutada veebipõhist küsimustikukeskkonda *Google Form*.

15.02.2014

Uurisin põhjalikumalt Ida-Virumaa küsimustikku. Lisaks uurisin ka erinevaid veebikeskkondi. Jõudsin järeldusele, et võib lähtuda Ida-Virumaa näitest, kuna on põhjalik ja annab hea ülevaate erinevatest töösoovijatest.

## Lisa 2 järg

28.02.2014 põhikriteeriumid, vajadused ev, jne

Täpsustasin põhikriteeriumid, vajadused. Uurisin erinevaid küsimustiku keskkondi nagu *Limesurvey* ja *Surveymonkey*. Valituks osutus siiski Google, kun on tasuta ning lihsamini ligipääsetav.

12.03.2014

Valmis esimene prototüüp, mille esitasin ka projekti tellijale. Lähtusin Ida-Virumaa küsimustikust.

19.03.2014

Toimus koosolek Töötukassa esindajate, Sulev Alajõe ja projektijuhi vahel. Vaatasime üle kõik küsimused ning arutlesime nende vajalikkuse või täiendusvõimaluste üle. Selgus, et tegelikult pole tööpakkujale suunatud küsimustikku vaja, kuna see on olemas Töötukassa lehel. Seega pole mõtet tööpakkujaid eraldi koguda, kui andmebaasi hakkab haldama nagunii töötukassa. Samal põhjusel otsustati muuta ka küsimustiku struktuuri ning arutleti, et parem on küsida kõigi inimeste töösoove, mitte ainult konkreetsest pakkumisest huvitatute omi. Seega tuleb teha veidi ümber struktuur ja kujundada ümber andmete kogumine. Arutelu käigus otsustati kaotada mitmed küsimused ning vältida isikuomaduste uurimist. Andmebaasi idee seisneb lihtsuses ja oma kodukoha töösoovist peab saama kiirelt märku anda ka Soomes töötav inimene, kes satub arvutisse võib-olla harva ja aja viitmiseks.

28.03.2014

Sain valmis teise prototüübi, mis arvestas koosoleku käigus tehtud ettepanekuid.

3.04.2014

Toimus kokkusaamine Oshino esindajaga. Kokkusaamist ajendas soov teada saada ettevõtja nägemus küsimustikust ning eristada olulisemad kriteeriumid. Selgus, et mõned küsimused olid ebavajalikud. Nendeks olid küsimused arvuti kasutamise kohta



## Lisa 2 järg

või ISCO koodi konkreetne küsimine. Arutelu käigus otsustati koodi küsimine asendada nimetuse või kirjelduse küsimisega, mille alusel andmebaasi haldaja ise koodi määrab. Ettevõtte esindaja tegi ettepanekuid ka kujunduse osas ning soovitas proovida ISCO koodi funktsioon lisada küsimustikku.

02.04.2014

Kuna mõlemad Oshino esindaja soovitusel ei ole *Google Form* abil teostatavad, otsustasin proovida *Limesurvey* keskkonda. Nimetatud keskkond oli palju kasutajavaenulik, mis tähendaks halvemat vastuvõtlikkust töötukassa esindaja poolt. *Javascripti* abil lisati küsimustikku ISCO klassifikaator. Kuna *Limesurvey* keskkond lubab tasuta aga vaid 10 sisestamist, ei ole see väga otstarbekas. Eriti veel kui võrrelda kasutajasõbralikkust *Google Formiga*. *Limesurvey-l* on olemas ka Tartu Ülikooli leping, kuid sellel on piiratud funktsionaalsus koodide lisamise osas ning lisaks on see sellisel juhul kasutatav vaid kuni projektijuht oleks seotud kolledžiga. Sulev Alajõe tegi ettepaneku osaleda töövahendusportaali esindamisega ka töömessil.

7.04.2014 III uus täisversioon

Valmis sai uus täisversioon, mille esitasin projekti tellijale.

8.04.2014

Esinesin personaliandmebaasiga Pärnumaa töömessil. Huvi oli küllalt suur ning kõige rohkem tundsid huvi vanemaealised inimesed, kes rääkisid, et laps või lapselaps käib Tallinnas või Soomes tööl, sest Pärnus lihtsalt ei leia tööd.

Pärnu Postimehe hinnangul oli töövahendusportaal üks huvitavamaid esitlusi töömessil ning tegi videointervjuu minu ja projekti tellijaga. Lisaks tundis huvi ka ERR-i toimetuse, kes tegi minuga lühiintervjuu, mida näidati ka Aktuaalse Kaamera raames. Selline meediakajastus oli hea turunduseks ning suurendas inimeste teadlikkust veelgi.

## Lisa 2 järg

Messil tekkis plaan kasutada turundustegevuse jaoks veebireklaami ning osaleda sotsiaalmeedia töövahendusfoorumites.

14.04.2014

Koostasid mõned reklaamriba versioonid. Kujutised valmistasin *Flash* programmi abil ja esitasin tehtud eksemplariid projekti tellijale. Reklaamribad hakkavad paiknema PEAKi kodulehel ja hiljem ka teistes veebikeskkondades. Teostasid esmase portaali jagamise suhtlusportaalide vastavates töövahenduse kommuunides.

Projekti käigus õppisid omavahel siduma erinevaid kolledžis omandatud teadmisi. Kõige enam sai kinnistatud teadmisi projektijuhtimise, personalijuhtimise, turunduse, majandusarvestuse, finantsjuhtimise ja majandusinformaatika kohta. Juhtimisainetes õpitu andis kindluse erinevate otsuste vastuvõtmisel ja muudatuste tegemisel. Teadmised majandusarvestuse ja finantsjuhtimise aspektidest võimaldas mõista projekti vajalikkust ja ootusi tööpakkujatele, majandusinformaatika oli toeks suhtluse korraldamisel ja erinevate tegevuste planeerimisel.

### Lisa 3. SA Ida- Virumaa Tööstusalade Arenduse ankeet ja selle kokkuvõte

10	<b>I. ОБРАЗОВАНИЕ</b>			
11	<b>I.1. Уровень образования:</b>			
12	Высшее (ВО)	<input type="checkbox"/>	→	
13	в т.ч. высшее техническое	<input type="checkbox"/>	→	
14	Среднее профессиональное (СП)	<input type="checkbox"/>	→	
15	Среднее (С)	<input type="checkbox"/>		
16	Неполное среднее (НС)	<input type="checkbox"/>		
17	Начальное (Н)	<input type="checkbox"/>		
18	<b>I.2. Дополнительное проф. образование:</b>			
19	Обучение на квалификацию, с получением сертификата (К)	<input type="checkbox"/>	→	
20	Прочее обучение, без получения сертификата (П)	<input type="checkbox"/>	→	
21				
22				
23				
24				
25				
26	<b>I.2. Название специальности в соответствии с дипломом или квалификационным свидетельством (КС):</b>			
27	Название специальности по диплому или КС	Название специальности на русском языке (если в дипломе или КС написано не на русском)	Разряд/категория	
28				
29	1			
30	2			
31	3			
32				
33	<b>II. ОПЫТ</b>			
34	<b>II.1. На предприятиях и организациях какого типа у Вас есть опыт работы?</b>			
35			лет:	
36	Логистика	<input type="checkbox"/>		
37	Торговля	<input type="checkbox"/>		
38	Бизнес-услуги	<input type="checkbox"/>		
39	Сфера обслуживания населения	<input type="checkbox"/>		
40	Сфера образования	<input type="checkbox"/>		
41	Государственное/муниципальное управление служба	<input type="checkbox"/>		
42	Некоммерческие организации	<input type="checkbox"/>		
43	Производство	<input type="checkbox"/>		
44	Крупное (более 250 работников)	<input type="checkbox"/>		
45	Среднее (20-250 работников)	<input type="checkbox"/>		
46	Малое (до 20 работников)	<input type="checkbox"/>		
47				
48	<b>II.1.a. Если работали на производственных предприятиях, то в какой сфере?</b>			
49			лет:	
50	Машиностроение и металлообработка	<input type="checkbox"/>		
51	Деревообработка и мебель	<input type="checkbox"/>		
52	Химическая промышленность	<input type="checkbox"/>		
53				
54				
	<b>II.2. На каких позициях Вы имели опыт работы? Сколько лет?</b>			
			лет:	
	Управляющий персонал			
	Руководитель высшего звена	<input type="checkbox"/>		
	Руководитель среднего звена	<input type="checkbox"/>		
	Руководитель первичного звена	<input type="checkbox"/>		
	ИТР, специалисты и вспомогательный персонал			
	Ведущий специалист	<input type="checkbox"/>		
	Специалист среднего звена	<input type="checkbox"/>		
	Лаборант	<input type="checkbox"/>		
	Бухгалтер	<input type="checkbox"/>		
	Секретарь	<input type="checkbox"/>		
	Контролер качества	<input type="checkbox"/>		
	Рабочие производства и логистики			
	Наладчик оборудования	<input type="checkbox"/>		
	Оператор станка	<input type="checkbox"/>		
	Операторы линии	<input type="checkbox"/>		
	Сборщик ("отверточная" сборка)	<input type="checkbox"/>		
	Водитель погрузчика	<input type="checkbox"/>		
	Слесарь	<input type="checkbox"/>		

**II.2.a. По какому направлению Вы действовали, находясь на указанных позициях? Сколько лет?**

	Складское хозяйство и логистика	<input type="checkbox"/>		
	Электроника	<input type="checkbox"/>		
	Электротехника	<input type="checkbox"/>		
	Автоматика и управление	<input type="checkbox"/>		
	Пластмассы, стекловолокно	<input type="checkbox"/>		
	Машиностроение и металлообработка	<input type="checkbox"/>		
	Технология материалов	<input type="checkbox"/>		
	Технология товаров широкого потребления	<input type="checkbox"/>		
	Химические технологии	<input type="checkbox"/>		
	Другие технические направления	<input type="checkbox"/>		
	Естественные науки и математика	<input type="checkbox"/>		
	Экономика и управление	<input type="checkbox"/>		

# Lisa 3 järg

51	Машиностроение и металлообработка	<input type="checkbox"/>				Сборщик ("отверточная" сборка)	<input type="checkbox"/>				
52	Деревообработка и мебель	<input type="checkbox"/>				Водитель погрузчика	<input type="checkbox"/>				
53	Химическая промышленность	<input type="checkbox"/>				Слесарь	<input type="checkbox"/>				
54	Электроника и приборостроение	<input type="checkbox"/>				Фрезеровщик	<input type="checkbox"/>				
55	Изготовление пластиковых изделий	<input type="checkbox"/>				Токарь	<input type="checkbox"/>				
56	Электроэнергетика	<input type="checkbox"/>				Электрик	<input type="checkbox"/>				
57	Текстильное производство	<input type="checkbox"/>				Шлифовальщик	<input type="checkbox"/>				
58	Швейное производство	<input type="checkbox"/>				Инструментальщик	<input type="checkbox"/>				
59	Производство кожи и обуви	<input type="checkbox"/>				Другие квалифицированные рабочие	<input type="checkbox"/>				
60	Другое производство (уточнить):	<input type="checkbox"/>				Рабочий склада	<input type="checkbox"/>				
61						Разнорабочий	<input type="checkbox"/>				
62											
63	Обслуживание населения	<input type="checkbox"/>				Производство и технология	<input type="checkbox"/>			Маркетинг	<input type="checkbox"/>
64	Туризм	<input type="checkbox"/>				Обслуживание производства	<input type="checkbox"/>			Финансы	<input type="checkbox"/>
65	Задачи публичного сектора	<input type="checkbox"/>				Работа с персоналом	<input type="checkbox"/>			Логистика	<input type="checkbox"/>
66	Розничная продажа	<input type="checkbox"/>				Разработка продукции	<input type="checkbox"/>			Снабжение	<input type="checkbox"/>
67	Охрана, спасение	<input type="checkbox"/>				Обучение/консультирование	<input type="checkbox"/>				
68											
69	<b>III. КАКУЮ РАБОТУ ИЩЕТЕ (НА ПРОИЗВОДСТВЕННОМ ИЛИ ЛОГИСТИЧЕСКОМ ПРЕДПРИЯТИИ)?</b>										
70											
71	<b>III.1. Желаемая позиция:</b>			Желаемая зарплата (нетто), евро:							
72											
73	Управляющий персонал										
74	Руководитель высшего звена										
75	Руководитель среднего звена										
76	Руководитель первичного звена										
77	ИТР, специалисты и вспомогательный персонал										
78	Ведущий специалист										
79	Специалист среднего звена										
80	Лаборант										
81	Бухгалтер										
82	Секретарь										
83	Контролер качества										
84	Рабочие производства и логистики										
85	Наладчик оборудования										
86	Оператор станка										
87	Операторы линии										
88	Сборщик ("отверточная" сборка)										
89	Водитель погрузчика										
90	Слесарь										
91	Фрезеровщик										
92	Токарь										
93											
94	Требуется ли Вам дополнительное обучение, чтобы работать на выбранных позициях: <input type="checkbox"/> да <input type="checkbox"/> нет										
95											
96	<b>IV. ЛИЧНОСТНЫЕ ХАРАКТЕРИСТИКИ И МОТИВАЦИЯ</b>										
98	<b>IV.1. Оцените по 5-ти бальной системе наличие у Вас следующих качеств (0-отсутствуют вообще, 5- присутствуют в максимальной степени):</b>										
99	1 Коммуникабельность					6 Способность к обучению					
100	2 Самообладание					7 Аккуратность					
101	3 Лидерство					8 Способность к выполнению рутинной работы					
102	4 Внимательность					9 Активность					
103	5 Конструктивность					10 Способность генерировать новые идеи					
104											
105	<b>IV.2. Расставьте по степени значимости для Вас, следующие показатели (0 - наименее важное, 10 самое важное). Свои дополнительные пожелания по льготам можно вписать в позиции 5-7.</b>										
106											
107	1 Деньги					8 Стабильность, надежность предприятия					
108	2 Дополнительные льготы:					9 Престиж компании					
109	3 Бесплатное питание					10 Дружелюбный коллектив					
110	4 Бесплатный транспорт до работы					11 Приобретение нового опыта работы					
111	5					12 Самостоятельность и ответственность позиции					
112	6					13 Возможность карьерного роста					
113	7					14 Работа в одной сфере					

## Lisa 3 järg

106	льготам можно вписать в позиции 5-7.				
107	1	Деньги		8	Стабильность, надежность предприятия
108	2	Дополнительные льготы:		9	Престиж компании
109	3	Бесплатное питание		10	Дружелюбный коллектив
110	4	Бесплатный транспорт до работы		11	Приобретение нового опыта работы
111	5			12	Самостоятельность и ответственность позиции
112	6			13	Возможность карьерного роста
113	7			14	Работа в одну смену
114					
115	IV.3. Оцените свое знание языков (5-отлично, 0-плохо. Если Вы знаете другие языки, то допишите их в строки 4 и 5)				
116			Говорю	Понимаю	Пишу
117	1	Русский			
118	2	Эстонский			
119	3	Английский			
120	4				
121	5				
122					
123					
124					
125					
126					
127	Предоставляя заполненную анкету Нарвскому Парку производства и логистики (НППЛ), Вы даете согласие на обработку Ваших личных				
128	данных, а также на передачу данных потенциальным работодателям, а также организациям партнерам НППЛ по трудоустройству.				
129					

Tegemist üsna mahuka küsimustikuga, kuid peamised sisendid on järgmised. Uuritakse üldandmeid (nimi, sünniaasta, sugu). Seejärel selgitatakse välja isiku haridustase ja täiendavad erialad, tunnistused. Varasem töökogemus ja erialane oskus palutakse kirjeldada üsna detailselt, küsides ettevõtte valdkonda, suurust, töötamise positsiooni ja tööaega. Tuntakse huvi juhtimiskogemuste kohta. Erilise asjaoluna võib märkida soovitud ametipositsiooni küsimise, mis on piisavalt põhjalik, et esitada inimesele sobiv tööpakkumine. Lisaks uuritakse isikuomadusi ja motivatsiooni, mis aitavad välja selgitada töötaja iseloomu.

## Lisa 4. Personaliotsingu andmebaasi esimene prototüüp

Page 1 of 6


### Küsimustik




Form Description

**Eesnimi**  
Sisestage oma eesnimi

**Perekonnanimi**  
Sisestage oma perenimi

**Sünniaeg**  
Valige oma sünniaeg

Month ▾ Day ▾ 2014 ▾ 

**Kontakttelefon**     
Sisestage oma telefoninumber

**Meili address**  
Sisestage oma meili address.

**Sugu**

**Milline on Teie amet?**

Add item ▾

After page 1 [Continue to next page](#) ▾

## Lisa 4 järg

Page 2 of 6

### Kui pikalt olete olnud töötu (küsimus aktiveerub vaid töötutele)

Mitu täisaastat?

Mitu kuud?

Add item ▾

After page 2 Continue to next page ↕

Page 3 of 6

### Haridustase ja väljaõpe

Milline on Teie haridustase?

Täiendavad erialakoolitused

- Olete läbinud kvalifikatsiooniväljaõppe (tunnistusega).
- Olete omandanud väljaõppe oma erialas erialatunnistusega.

**Valige loendist eriala, mis ühtib Teie erihariduse või väljaõppega.**

- Puidutööstus
- Elektroonika
- Metallitööstus
- Erialad tulevad vastavalt ettevõtete tööjõuvajadustele

Muu (täpsustage)

**Kui Teil on omandatud erialane diplom, sisestage selle nimi siia.**  
Sisestage diplomi või väljaõppepaberi nimetus nagu see on kirjas vastaval dokumendil.

Add item ▾

After page 3 Continue to next page ↕

## Lisa 4 järg

Page 4 of 6

After page 3 [Continue to next page](#) ↕

### Töökogemus

**Milliste erialadega neist on teil töökogemusi?**  
Mis tüüpi organisatsioonides olete töötanud?

Puidutööstus

Elektroonika

Metallitööstus

Kaubandus

Sisendid tulenevad siia ettevõtjate vajadustest lähtuvalt.

▾

Page 5 of 6

After page 4 [Continue to next page](#) ↕

### Soovitud ametipositsioon

**Mis tüüpi ettevõttes sooviksite töökohta omandada esimese eelistusena.**

Puidutööstus

Elektroonika

Metallitööstus

Kaubandus

Energeetika

Konkreetsed sisendid vastavalt ettevõtjate vajadustele.

**Oodatav ametipositsioon**

tippastme juht

keskastmejuht

spetsialist

raamatupidaja

tootmisüksuse juht

liinijuht

liinitööline

laotööline

abitööline

**Oodatav brutotootasu kuus**  
Märkige oodatav brutokuutasu eurodes.



## Lisa 4 järg

### Mis tüüpi ettevõttes sooviksite töökohta omandada teise eelistusena.

Teise eelistuse märkimine ei ole kohustuslik.

- Puidutööstus
- Elektroonika
- Metallitööstus
- Kaubandus
- Energeetika
- sisendid vastavalt ettevõtete vajadusele--

### Oodatav ametipositsioon

- tippastme juht
- keskastmejuht
- spetsialist
- raamatupidaja
- tootmisüksuse juht
- liinijuht
- liinitöoline
- laotöoline
- abitöoline

### Oodatav brutotöötasu kuus

Märkige oodatav brutokuutasu eurodes.

### Mis tüüpi ettevõttes sooviksite töökohta omandada kolmanda eelistusena.

Kolmanda eelistuse märkimine ei ole kohustuslik.

- Puidutööstus
- Elektroonika
- Metallitööstus
- Kaubandus
- Energeetika
- sisendid vastavalt ettevõtete vajadusele--

### Oodatav ametipositsioon

- tippastme juht
- keskastmejuht
- spetsialist
- raamatupidaja
- tootmisüksuse juht
- liinijuht
- liinitöoline
- laotöoline
- abitöoline

### Oodatav brutotöötasu kuus

Märkige oodatav brutokuutasu eurodes.

Add item ▼

## Lisa 4 järg

Page 6 of 6 Alter page 5 Continue to next page ▾

**Lisaküsimused**  
küsimused isiksuseomaduste või muu kohta.

1 2 3 4 5

▾

## Lisa 5. Erinevate töövahendusportaalide ülevaade ja võrdlus

Keskkonna nimi	Kirjeldus	Erinevus uuritavast projektist
CV-keskus (Firmast...2014)	Toob töötajatele tööpakkumised lähemale ning aitab ka CV-sid koostada. Tööpakkujatele pakub ta aga kiiret ja lihtsat viisi leida vajalikku töötajad. Oma kodulehel pakub CV-keskus välja ka et võib ise korraldada ka personali värbamise.	Ei ole otseselt suunatud piirkonda sisenevatele ettevõtetele ning samuti ei too ta välja ühe piirkonna potentsiaalset tööturget kuna ei hõlma endas inimesi, kes käivad tööl teises piirkonnas.
Leia.ee (Keskkonnast..2014)	Pakub nii töötajatele kui ka tööandjatele tasuta veebikeskkonda, kuhu igaüks võib oma töökuulutuse üles panna. Nimelt saab ettevõtja tasuta otsida töötajad kui ka töötaja panna üles oma kuulutuse, millist töötajad ta omalt poolt pakub.	Hõlmab endas üle-eestilist tööturget. Samuti ei anna leia.ee mingisugust statistikat, millest saaks aru kui palju töötajaid ühes piirkonnas kokku on.
Soov.ee (Töö- pakkumine... 2014)	Rohkem kuulutustele suunatud veebikeskkond, mis on kuulsust saanud oma loosungiga „ostan-müün-vahetan“. Kuigi ta hõlmab endas nii töö pakkumise kui ka töö otsimise kuulutusi, ei ole nende kuulutuste arv võrreldav eeltoodud veebikeskkondadega.	Ei keskendu töötajatele ja ei võimalda teha ülevaadet töötajate arvu kohta.
Töötukassa (Arengukava.. 2014)	Töötukassa tähtsaim ülesanne on aga abistada nii töötajaid kui ka tööandjaid nõu, otsese abiga ning pakudes välja ka erinevaid koolitusi töötajate spetsialiseerumiseks.	Kuigi Eesti Töötukassa annab ka välja erinevat statistikat nii töötajate hõive kohta kui ka muu tööturget puudutava informatsiooni kohta, ei paku ta teenust kaardistamiseks potentsiaalset töötajad ühele piirkonda sisenevatele ettevõtetele.

Võrdluse koostamiseks kasutatud viited on järgnevad.

1. Firmast. CV Keskus. [<http://www.cvkeskus.ee/company.php>] 16.01.2014.
2. Keskkonnast. Leia.ee. [<http://www.leia.ee/et/keskkonnast>] 16.01.2014.
3. Tööpakkumine. Kuulutused. Soov.ee  
[<http://www.sov.ee/eesti?q=&cg=7120&w=3&ca=1&l=0&md=th>] 16.01.2014.
4. Arengukava. Töötukassast. Eesti töötukassa.  
[<http://www.tootukassa.ee/content/tootukassast/arengukava>] 19.01.2014.

## Lisa 6. Personaliotsingu andmebaasi lõppversioon

Page 1 of 4

### Töötajate andmebaas

Töötajate ja -pakkujate paremaks kokkuvõimiseks oleme loonud küsimustiku, mis keskendub töötajate andmete kogumisele. Küsimustikku on oodatud täitma kõik, kes otsivad töökohta Pärnus või Pärnumaal. Eesmärkideks on tööjõu ja ettevõtjate tõhusam kokkuvõimine, pendelrände vähendamine ning Pärnumaa tööturu parendamine.

Kõik kogutud andmed salvestatakse turvaliselt ega kuulu üldsusele avalikustamiseks. Sisestatud andmeid võrreldakse tööpakkujate soovidega ning sobivuse korral antakse Teile sellest teada.

Sobivimate pakkumiste saamiseks palume täita väljad võimalikult täpselt.

### Üldandmed

Eesnimi\*

Perekonnanimi\*

Praegune riik, kus töötate\*

Kontakttelefon\*

Meiliaadress\*

### Haridustase ja oskused

Mis on Teie kõrgeim omandatud haridustase\*

Valige loetelust

Eriala (valikuline)

Kui kõrgeima omandatud haridustasemega omandasite ka eriala, märkige see siia.

Õppeasutuse nimi\*

Nt: Tartu Ülikooli Pärnu kolledž.

Lõpetamise aasta\*

Sisestage lõpetamise aasta.

## Lisa 6 järg

### Sõidukategooriad

13		9	10	11	12
	A1				
	A				
	B1				
	B				
	C1				
	C				
	D1				
	D				
	BE				
	C1E				
	CE				
	D1E				
	DE				
	T				
	R				

#### Milliste kategooriate juhtimisõigust omate?\*

Märkige vastavad linnukestega

- Ei oma juhiluba
- A1
- A
- B1
- B
- C1
- C
- D1
- D
- BE
- C1E
- CE
- D1E
- DE
- T
- R

#### Milline on Teie keeleoskus järgmistes keeltes.

	Ei oska	Madal	Kesktaase	Kõrgtaase
inglise	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
soome	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
vene	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
saksa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
muu (täpsustada allpool)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kui märkisite, et oskate lisaks mõnda muud keelt, sisestage see siia.

#### Kas soovite lisada informatsiooni läbitud täiendkoolituste kohta?\*

Kui olete läbinud täiendkoolitusi, mida soovite andmebaasi sisestada, valige loetelust "Jah".

Jätkamiseks vajutage nupule "Continue".

Add item ▼

After page 1

Continue to next page ⇅

## Lisa 6 järg

Page 2 of 4

### Täiendkoolitused

Sisestage andmed kuni kolme läbitud erialakoolituse kohta (A,B ja C).

(A) Eriala nimetus

(A) Koolitusasutuse nimi

(A) Lõpetamise aasta

---

(B) Eriala nimetus

(B) Koolitusasutuse nimi

(B) Lõpetamise aasta

---

(C) Eriala nimetus

(C) Koolitusasutuse nimi

(C) Lõpetamise aasta

---

Jätkamiseks vajutage nupule "Continue", tagasi liikumiseks nupule "Back".

Add item ▼

After page 2 [Continue to next page](#) ↕

## Lisa 6 järg

Page 3 of 4

### Varasem töökogemus

Märkige üles viimased kolm töökohta (1,2 ja 3) alustades hilisemast. Kui töökohti on olnud vähem, siis sisestage niipalju, kui neid on olnud.

Ametinimetuste määramisel soovitame kasutada ISCO koodi, mis on saadaval järgneval lingil. <http://goo.gl/p6zYDs>

Näiteks maalritöö puhul 7131.

#### Tööandja 1 nimetus

Kirjutage siia oma tööandja nimetus. Nt: OÜ Saarepealne.

#### Vastava töökooha ametinimetus

#### Vastava tööandja juures töötatud aeg aastates

#### Tööandja 2 nimetus

Kirjutage siia oma tööandja nimetus. Nt: OÜ Saarepealne.

#### Vastava töökooha ametinimetus

#### Vastava tööandja juures töötatud aeg aastates

#### Tööandja 3 nimetus

Kirjutage siia oma tööandja nimetus. Nt: OÜ Saarepealne.

#### Vastava töökooha ametinimetus

#### Vastava tööandja juures töötatud aeg aastates

## Lisa 6 järg

Page 4 of 4

### Töösoovid

Märkige üles informatsioon kuni kahe oodatava töökoha kohta.

Ametinimetuste määramisel soovitame kasutada kuni kolmekohalist ISCO koodi, mis on saadaval järgneval lingil. <http://goo.gl/p6zYDs>

Võimalik on sisestada kuni kaks (A ja B) eelistatud ametikoha/valdkonna soovi.

**(A) Kirjeldage, millist tööd sooviksite teha.\***

**(A) Kui oskate, siis sisestage ka soovitud töökoha ametinimetus.**

**(A) Milline on eelistatud tööaeg?**

**(A) Minimaalne netopalk kuus (eurodes)?**

Milline oleks Teie minimaalne netosissetulek antud ametikohal. Näiteks: 450

**(B) Kirjeldage, millist tööd sooviksite teha.**

Kui soovite, saate lisada ka teise valdkonna või töökoha.

**(B) Kui oskate, siis sisestage ka soovitud töökoha ametinimetus.**

**(B) Milline on soovitud tööaeg?**

**(B) Minimaalne netopalk kuus (eurodes)?**

Milline oleks minimaalne netosissetulek antud ametikohal, alla mille Te võimalikke tööpakkumisi ei soovi.

Vastuste kinnitamiseks vajutage nupule "Submit", tagasi liikumiseks nupule "Back".



# SUMMARY

## APPLYING AGILE METHODS ON SMALL-SCALE SOFTWARE PROJECTS

Marco Peedo

Agile methods are believed to be the most suitable choice for managing software development projects. As the technology keeps developing in the modern world at a rapid rate, the importance of software projects is rising and continuously changing the related requirements. The common approach is characterized by profound design and is best applied in a relatively immutable environment. Therefore an increasing number of developers find that using agile methods is more productive than classical project development techniques.

The main goal of this thesis is to experiment with different methods, which are used in managing small-scale software projects and to work out directions on how to conduct similar projects. According to the objective, there are a few certain investigative tasks :

- Find the characteristics of small-scale software projects.
- Analyse the differences between the agile and traditional methods.
- Analyse the principles and implementation of different agile strategies based on literature.
- Test the utility of various methods on a demonstration project.
- Analyse the devised demonstration project and evaluate the suitability of used methods.
- Create directions on conducting similar small-scale software projects.

The theoretical background is created by using knowledge in project management and mainly books, articles or internet sources reflecting the current methods. Different theoretical treatments give an overview of the differences, similarities, principles and implementation of evolved approaches.

The researched agile methods originated from the Agile manifesto of project management and they suggested the tools and principles to follow. The crystal method

recommends specifying the size of the project and according to that deploy either flexible or more traditional approach. Dynamic software development model (DSDM) and feature-driven development (FDD) guides the project manager to reduce the possible risk that comes with the flexibility. These methods insist on putting more focus on the beginning of the activity and defining certain requirements, before starting to plan a step-by-step constantly progressing development.

An approach based thoroughly on flexible methods was suggested by Scrum and the extreme programming method. *Scrum* offered an operational structure, which helped to plan the creation of prototypes. Extreme programming (XP) is meant for rather specific development, for example writing a piece of code or designing while presenting techniques, which allow for a freely changeable and progressive environment.

Using action research, a model project was investigated within the framework of the study and in order to do the research, the use of different methods and their evaluation was planned beforehand. The activities followed a cyclic model consisting of planning, taking action, observation and reflection. During the research it appeared that all of the studied methods were at least somewhat suitable for deployment. The Crystal methods suggested choosing the most flexible methods, which was later confirmed by the analysis as well. Agile methods such as Scrum and extreme programming offered freedom of choice while giving firm directions on making decisions and carrying out activities.

More traditional approaches like DSDM model and FDD were found to be suitable within context of small-scale software projects. The main benefits of these methods appeared when they were combined with other methods. This meant starting the project as agile as possible, using Scrum or XP and when it was possible, agreeing with key features and continuing with traditional methods. This meant that the project is flexible and open to rather big changes during the first few stages of the project and limiting the risk in the later stages. Defining key features in the beginning of the project is not always practical and in the test project, it would have meant extra work for project team, that the owner really doesn't need. Action research showed that even if it is possible to

define unchangeable features, it could be difficult to plan exact cycles for developing them.

The Author recommends evaluating the scope and risk to choose a certain agile method. It is necessary to understand how strictly definable are the features and deadlines and choose and combine methods accordingly. If project manager has many projects in different sizes and risk levels, they should use Crystal methods, which helps to define suitable methods and tools.

Scrum is most applicable when dealing with lower risk levels, which can easily be combined with XP. With greater risk, it is recommended to use DSDM or FDD and its tools. When a project manager faces a small scale project which has medium or big risk level, they should combine Scrum and XP with other, more traditional methods. In that case the manager should prefer more agile methods (Such as Scrum) in the first cycles of the project. As soon as the project manager and owner both know the course and main features of the project, it is recommended to start using DSDM or FDD.

## **Lihlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, Marco Peedo,

1. annan Tartu Ülikoolile tasuta loa (lihlitsentsi) enda loodud teose „Paindlike meetodite rakendamine väikesemahuliste tarkvaraprojektide puhul“, mille juhendaja on dotsent Arvi Kuura,

1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

1.3 olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

1.4.kinnitan, et lihlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Pärnus, **21.05.2014**