

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Viktoria Lemberg 172780IABM

**AGIILSE LÄHENEMISE
RAKENDUSVÕIMALUSTE UURIMINE
KESKKONNAMINISTEERIUMI
HALDUSALA TARKVARAPROTSESSIS**

Magistritöö

Juhendaja: Kuldar Taveter
Professor

Tallinn 2019

Autorideklaratsioon

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

Autor: Viktoria Lemberg

03.01.2019

Annotatsioon

Käesoleva töö eesmärgiks on uurida agiilse lähenemise rakendusvõimalusi Keskkonnaministeeriumi (KeM) haldusala tarkvaraprotsessi jaoks.

Tulemuste saavutamiseks on autor kasutanud teiseid ja esmaseid andmeid. Magistritöö raames uurib autor erinevaid agiilseid arendusmetoodikaid ning tuvastab maailma ja Eesti parimad praktikad. Uuringutest selgub millised arendusmetoodikad on kasutatavad maailmas ja Eestis ning millised on enim kasutatavad avalikus sektoris. Magistritöö autor viis läbi kvalitatiivse uuringu saades vastuseid läbi viidud intervjuudest, kasutades struktureerimata küsimustikku. Tulemusi analüüses leiab autor, et Keskkonnaministeeriumi haldusalas on võimalik rakendada agiilset lähenemist, arvestades olemasolevate probleemidega ning võimalustega.

Autor on käesoleva magistritöö raames leidnud vastuseid kõikidele uurimusküsimustele ning saavutas oma eesmärgi. Autori ettepanek on Scrum metoodika juurutamiseks KeM haldusala tarkvaraprotsessi.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 41 leheküljel, 5 peatükki, 5 joonist, 2 tabeli.

Abstract

Studying the application possibilities of agile approach in the software process of the Ministry of Environment's administrative domain

The thesis aim is to study application possibilities of agile approach in the software process of the Ministry of Environment's administrative domain.

To achieve the aim of the thesis, secondary and primary data were used. Different agile development methodologies and studies were found and explored by the author. Studies showed which agile methodologies are used worldwide and in Estonia and also in public organisations. Author conducts a qualitative research via unstructured survey. Results showed, that there's a possibility to apply agile approach in the Ministry of Environment administrative domain, considering all existing problems and possibilities.

Author founded all answers to the research questions and the Master's thesis achieved its aim. Author's main proposal of the thesis is to root Scrum methodology into software process of the Ministry of Environment's administrative domain.

The thesis is in Estonian and contains 41 pages of text, 5 chapters, 5 figures, 2 tables.

Lühendite ja mõistete sõnastik

| | |
|----------------------------------|--|
| Arendusprojekt | Ka tarkvara arendusprojekt. Tegevus, mis algab ideest/visioonist kuni arendatava lahenduseni välja. |
| DSDM | Inglise keelest Dynamic Systems Development Method ehk dünaamiline süsteemiarendusmeetod. Tegemist on lähenemisega või metoodilise raamistikuga |
| Fond | Rahastusmudel, mille kaudu on võimalik taotleda raha infosüsteemide arendamiseks või analüüside läbiviimiseks. Töös kasutatakse ka nimetusena rahastusfond. See on |
| IT | Infotehnoloogia |
| Jira | Infosüsteem, milles on kirjas kõik tegevused, mida on vaja teha, mida tehakse ja mis on juba tehtud. Süsteembtalletab ka tegevuste ajaloo. |
| Kanban | Praktika, Lean metodoloogia meetod, mida kasutatakse tavaliselt Scrum-s tööde visualiseerimisel. |
| KeM | Keskkonnaministeerium |
| KEMIT | Keskkonnaministeeriumi Infotehnoloogiakeskus |
| KEMIT-i tarkvara arendusprotsess | Arendusprotsess reguleerib tarkvara arendust Keskkonnaministeeriumi haldusalas. Töös kasutatakse samuti nimega kord või KeM haldusala tarkvara protsessi kord või KeM haldusala tarkvaraprotsess. |
| Kose mudel | Inglise keelest Waterfall – traditsiooniline tarkvara arendusmetoodika, mille puhul sooritatakse tarkvara ehitamisel jadamisi selleks vajalikud tegevused nagu nõuete analüüs, projekteerimine, kodeerimine, testimine, kasutamine ja hooldus. |
| KPMG | Firma, kes pakub auditi-, õigus- ja ärinõustamisteenuseid. |
| Lean metodoloogia | Kulusäästlik lähenemine või metoodiline raamistik, mille eesmärk on kliendile väärtuse pakkumine, üleliigse kõrvaldamine. |
| LSD | Lean Software Development |
| MoSCoW | Tööde hindamise meetod |
| Partner | Töös kasutatakse ka nimetustena hankepartner, väline partner. Tegemist on ettevõttega, kes hankelepingu kohaselt teostab arendustöid või viib läbi analüüsi. |

| | |
|--------------------------|---|
| PWC | PricewaterhouseCoopers, raamatupidamis- ja konsultatsioonifirma |
| Reliis | Inglise keeles release ehk tarkvara versiooni väljalase. |
| Scrum | Iteratiivne ja inkrementaalne agiilse tarkvaraarendusmetoodika toote arendamiseks. |
| Slack | Meeskonnatöö haldusrakendus |
| Skype | Suhtlusprogramm |
| SMIT | Siseministeriumi infotehnoloogia- ja arenduskeskus |
| Tarkvara arendusprojekt | Vt arendusprojekt |
| Tarkvara arendusprotsess | Tarkvaraprotsess on tegevuste jada, mis algab vajaduse määratlemisest ning lõpeb vajadust rahuldava toote käigust maha võtmisega. |
| Tarkvaraarendusprotsess | Tarkvaraprotsess |
| Tarkvaraprotsess | Vt tarkvara arendusprotsess |
| XP | Inglise keelest extreme programming. Tegemist on agiilse arendusmeetodiga, mis põhineb iteratiivsel mudelil. |

Sisukord

| | |
|---|----|
| 1 Sissejuhatus | 10 |
| 2 Teoreetilised alused | 13 |
| 2.1 Traditsiooniline ja agiilne tarkvaraarendus. Agiilne filosoofia. | 13 |
| 2.2 Agiilsuse komponendid | 14 |
| 2.2.1 Printsüübid | 15 |
| 2.2.2 Tehnikad | 16 |
| 2.2.3 Metoodikad | 20 |
| 3 Metoodika | 33 |
| 3.1 Uurimismeetod | 33 |
| 3.2 Andmekogumismeetodid | 33 |
| 3.3 Valim | 36 |
| 3.4 Andmete töötlemine | 36 |
| 4 Tulemused ja analüüs | 37 |
| 4.1 Intervjuude tulemused | 37 |
| 4.1.1 Tulemuste tõlgendamine | 37 |
| 4.2 Agiilse lähenemise rakendusvõimaluste ettepanekud KEMIT-i tarkvaraprotsessi | 45 |
| 4.2.1 Järeldused ning ettepanekud | 45 |
| 5 Kokkuvõte | 49 |
| Kasutatud kirjandus | 51 |
| Lisa 1 – Intervjuu küsimused | 54 |

Jooniste loetelu

| | |
|---|----|
| Joonis 1. Tarkvaraarenduse metoodikate ajajoon..... | 15 |
| Joonis 2. Vajalikud organisatsioonilised muudatused agiilse lähenemise rakendamiseks | 39 |
| Joonis 3. Võimalikud organisatsioonilised barjäärid agiilse lähenemise juurutamisel .. | 41 |
| Joonis 4. Arendusprojektide läbiviimist takistavad tegurid..... | 44 |
| Joonis 5. KEMIT-i võimalik tarkvaraprotsess jätkuarenduste puhul | 48 |

Tabelite loetelu

| | |
|--|----|
| Tabel 1. Populaarsemad agiilse tarkvaraarenduse tehnikad | 16 |
| Tabel 2. Agiilsete meetodikate erinevad rõhuasetused ja ühised omadused..... | 27 |

1 Sissejuhatus

„Eluslooduse ajalugu näitab, et olelusvõitluses jäid ellu need liigid, kes suutsid keskkonnatingimuste muutumisele kõige kiiremini enda muutmisega vastata.“

(Warner Bennis, 1966)

“Muutumine ei ole ilmtingimata vajalik. Ellujäämine ei ole kohustuslik.”

(Edwards Deming, 1955)

Innovatsioon - see on kui lahendus probleemidele, väljakutsetele või võimalustele, mis kerkivad üles sotsiaal- ning majandusühiskonnas [1]. Ümbritsev keskkond muutub pidevalt ning selleks, et organisatsioon püsiks elavana ning konkurentsivõimeline, on paratamatult tarvis viia sisse muudatusi ja nendega kohaneda. Paraku ainult uudest ideedest üksi ei piisa, vaid need tuleb suuta ka ellu viia. Alas ning Salu Pramann Salu kirjutavad enda raamatus [2], et mitmed uuringud on näidanud, et paljudel juhtudel muudatused ebaõnnestuvad ja peamiseks põhjuseks on töötajate ebapiisav osalemine muudatustes. Siinkohal on oluline, et just juht oleks ettevõtlik ning suudaks töötajatele arusaadavaks teha vajaduse muudatuste järele ja luua muudatusi toetava ning õppimist soodustava organisatsioonikultuuri [2]. Margretta ja Stone väidavad oma raamatus [3], et vähe on ettevõtjaid ja mittetulundusühingute juhte, kes tunnistavad, et ilma investeeringuteta organisatsiooni ülesehitusse ning innovatsiooni ei suudeta tulevikus paremaid tulemusi saavutada. Tõepoolest, mistahes organisatsioonile on tarvis distsipliini, mis suunaks tulevikku ja töötaks vastu oleviku survele. Edler ja Fagerberg [1] toovad välja, et innovatsioon kui olemasoleva teadmise, võimekuse ja ressursside uus kombinatsioon, on peamiseks muudatuste allikaks igas majandustegevuses, olgu selleks siis vaene või rikas riik, madal- või kõrgtehnoloogia, teenused või tootmine, era- või avalik sektor. Head juhid teavad, kuidas ära kasutada strateegia, ärimudeli ja eelarve piiranguid – need tuleb muuta väljakutseteks, mis innustavad loovust ja leidlikkust. See ongi tõhusa

innovatsiooni olulisim meetod [3]. Meetod on konkreetse sihi saavutamise süstemaatiline viis [16].

Saavutamaks 21. sajandil konkurentsieelist, peavad organisatsioonid loobuma reageerivatest hoiakutest ning püüdlema ennetava tegevuse poole. See eeldab organisatsiooni liikmetel uute oskuste omandamist, hoiakute ning käitumisviiside muutmist ja pidevat õppimist. Organisatsioonikultuuri muutmist peetakse organisatsiooni muudatuste protsessi kõige raskemaks osaks. See on ka mõistetav, sest organisatsioonikultuur põhineb sügavalt juurdunud väärtustel. Kui eelnevalt on pidevalt rõhutatud nende väärtuste olulisust, on väga raske kõigile töötajatele selgeks teha, et see, millesse kõik uskusid, enam ei kehti [2].

Viimasel ajal on äri-ja ettevõtlusvaldkonnas populaarseks saanud mõiste agiilsus. Eesti keele instituudi Ametniku soovitusõnastiku kohaselt [4] tähendab agiilne eesti keeles *väle, kiire, paindlik, kohanev, kergesti kohandatav, muutlik, kiiresti muutuv*. Agiilne juhtimismall ja selle arendusmetoodikad on üha enam võitmas tähelepanu [5]. Selle alustalaks on läbipaistvus, riski varajane nähtavaletoomine, kontroll ja kohanemine. Just erasektoris on agiilsed meetodikad ennast tõestanud, aidates kaasa projekti edukusele, kliendi rahulolule ning võimaldades paindlikku muudatuste juhtimist [6]. Alles hiljuti on agiilsed arendushanked hakanud juurduma ka avalikku sektorisse. Eesti avaliku sektori organisatsioonid kasutavad agiilseid või hübriidmetoodikaid vähesel määral ning teadaolevalt ainult kaks organisatsiooni on oma tarkvara arendamiseks võtnud kasutusele agiilse tarkvara arendusprotsessi ehk tarkvaraprotsessi mudeli [7], [8]. Selleks, et tänapäeval olla mistahes majandustegevusharus konkurentsivõimeline, peavad organisatsioonid pakkuma kliendile väga head kliendikogemust. Kiirus, kvaliteet, prioritseerimine, distsipliin ja kohandumine on need võtmeomadused, mis ühendavad kliente ja teisi huvigruppe toote või teenuse kasutamisel ja pikaajalise kliendisuhete loomisel [9].

Antud töös käsitletavaks organisatsiooniks on Keskkonnaministeeriumi Infotehnoloogiakeskus (edaspidi KEMIT). KEMIT on Keskkonnaministeeriumi poolt hallatav riigiasutus, mille tegevusvaldkond on ministeeriumi ja selle valitsemisala asutustele infosüsteemide ja andmeside arendamine ning haldamine. Oma ülesannete täitmisel esindab KEMIT riiki.

KEMIT-i kõrgemalseisvaks valitsusasutuseks on Keskkonnaministeerium. Organisatsiooni struktuur, ülesanded ja juhtimise korraldus on kindlaks määratud KEMIT-i põhimääruses [10]. KEMIT loodi 1. jaanuaril 2013, mil ühendati Keskkonnaministeeriumi haldusala asutuste IT töötajad ja funktsioonid. Asutused, mida KEMIT teenindab, on Keskkonnaministeerium, Keskkonnainspektsioon, Keskkonnaamet, Keskkonnaagentuur, Loodusmuuseum, Maa-amet ja KEMIT ise.

KeM haldusala tarkvara arendust reguleerib KEMIT-i tarkvara arendusprotsess [11] (edaspidi tarkvaraprotsess). Protsess koosneb viiest (5) etapist: arendusvajaduste esitamine, ärianalüüs, arendamine, testimine ning rakendamine. Arendusprotsessi etapid on kirjeldatud tegevussammudena ning nende juures on välja toodud rollid, kes antud tegevustega seotud.

Käesoleva magistr töö eesmärgiks on uurida agiilse lähenemise rakendusvõimalusi KEMIT-i tarkvaraprotsessi jaoks. Uurimisküsimused, millele töö autor vastust otsib, on järgmised:

1. Milliseid agiilse arendusprotsessi praktikaid kasutatakse avalikus sektoris?
2. Milliseid agiilse arendusprotsessi praktikaid kasutatakse Eesti avalikus sektoris?
3. Kuidas agiilne lähenemine võiks kaasa aidata KeM haldusala tarkvaraarendusprojektidele?
4. Kuidas rakendada agiilset lähenemist KEMIT-i tarkvaraprotsessi?

Käesolev magistr töö koosneb viiest peatükist. Esimene peatükk on sissejuhatus. Teises peatükis tuuakse välja läbitöötatud kirjandust – erinevaid uurimusi, raamatuid ja artikleid, kaasa arvatud teadusartikleid. Kolmas peatükk annab ülevaate magistr töö uurimismeetodist, andmekogumismeetoditest, intervjueritavate valimist ja intervjuu käigus saadud andmete töötlemisest. Neljas peatükk on analüüs, autori poolt läbi viidud uuringu tulemustest saadud andmete põhjal, kus muuhulgas tuuakse välja järeldusi ning tehase ettepanekud rakendusvõimaluste kohta. Viimane peatükk on kokkuvõte.

2 Teoreetilised alused

Antud peatüki eesmärk on anda lühiülevaade tuntumatest agiilsetest tarkvaraarendusmetoodikatest. Seda kasutatakse neljandas (4) peatükis läbiviidud uuringu tulemuste analüüsis ning järelduste ja rakendustepanekute tegemisel KEMIT-i tarkvaraarendusprotsessi.

2.1 Traditsiooniline ja agiilne tarkvaraarendus. Agiilne filosoofia.

Tarkvaraprotsess on tegevuste jada, mis algab vajaduse määratlemisest ning lõpeb vajadust rahuldava toote käigust maha võtmisega [12]. Agiilne tarkvaraprotsess on välja arenenud traditsioonilisest lineaarsest lähenemisest, nagu seda on tarkvaraprotsessi Kose mudel (*Waterfall*) ja V-mudel. Traditsiooniline tarkvaraprotsess järgib Kose mudelit, mille puhul sooritatakse tarkvara ehitamisel jadamisi selleks vajalikud tegevused nagu nõuete analüüs, projekteerimine, kodeerimine, testimine, kasutamine ja hooldus. Ühe etapi töö tehakse korraga ära ja liigutakse uude etappi. Kose mudelit kasutatakse siis, kui probleem ja nõudmised on selged või olemasolevasse süsteemi tehakse arusaadavaid muudatusi. Kose mudelit järgides on keerulisemaid muudatusi paraku raske sisse viia, sest metoodika näeb ette, et kliendi soovid võetakse arvesse ja arendusprotsessi puudutavad otsused langetatakse tarkvaraprotsessi varajases faasis. Mudeli puuduseks on ka asjaolu, et projekti faasid on paindumatud ning valmistulemust nähakse alles lõppfaasis [13], [14].

Agiilne lähenemine tarkvaraarenduses sai alguse 90-ndatel aastatel vastukaaluks traditsioonilisele tarkvaraarendusele. Agiilne tarkvaraarendus muutus populaarseks 21. sajandi algul, mil üllitati „Agiilse tarkvaraarenduse manifest“ [15]. Manifesti avaldamise põhjuseks oli eelkõige asjaolu, et paljud tarkvaraprojektid ebaõnnestusid, järgides ettemääratud protsessi ja kindlat plaani. „Agiilse tarkvaraarenduse manifestiga“ [15] toodi välja agiilse tarkvaraprotsessi üldised põhimõtted ja väärtused. Agiilne tarkvaraprotsess on oma loomult adaptiivne ning kaasab rohkem tarkvaraarendusmeeskonna liikmeid ja klienti. Metoodika, see on standardite,

protseduuride ja tugimeetodite kogum, mis määratleb toote või süsteemiväljatöötamise käsitlemise [16]. Metoodikad sisaldavad endas protsessi kirjeldust, standardeid, rolle ja tarneid ning samuti tööriistu ja tehnikaid. Agiilsed metoodikad eeldavad, et tarkvaraarendusprojekti kõige olulisemad edutegurid on projektis osalejate omadused ja nende koostöö kvaliteet, protsessi ja instrumentide kasutamine on teisejärguline [14], [15].

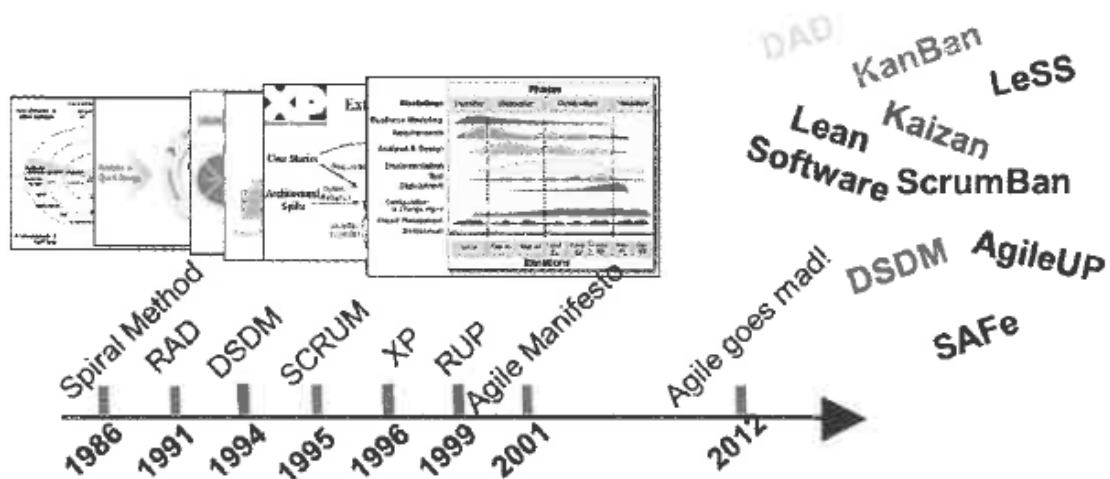
2.2 Agiilsuse komponendid

Agiilsuse kolm komponenti on printsiibid, metoodikad ja tehnikad.

Agiilse tarkvaraarenduse põhiprintsiipideks on koostöö kommunikatsiooni näol, iseorganiseeruvad meeskonnad, pidev täiustamine, interaktiivne arendamine ning järkjärguline tarne ja planeerimine, mis arvestab pidevalt asetleidvaid muudatusi [15]. Põhiprintsiipe kirjeldab autor detailsemalt alajaotises 2.2.1.

Agiilse tarkvaraarenduse metoodikad sisaldavad erinevaid agiilse tarkvaraarenduse tehnikaid [15]. Agiilse tarkvaraarenduse tehnikaid kirjeldab autor lähemalt alajaotises 2.2.2.

Agiilsete tarkvaraarenduse metoodikateevolutsioonilist kulgu näitab Joonis 1. Kõige populaarsemad agiilse tarkvaraarenduse metoodikad on XP, SCRUM, LSD ja DSDM [15]. Seejuures LSD ja DSDM on pigem lähenemised või metoodilised raamistikud, kui konkreetset metoodikad. See tähendab, et neid tuleb alati kasutada koos agiilse tarkvaraarenduse metoodikatega, nagu näites Scrum, XP. Sellest tulenebki agiilsete arendusmetoodikate evolutsiooniline elutsüklil. Populaarsemaid agiilse tarkvaraarenduse metoodikaid kirjeldab autor detailsemalt alajaotises 2.2.3.



Joonis 1. Tarkvaraarenduse metoodikate ajajoon

Allikas: [15]

2.2.1 Printsiihid

Tähtsaim agiilse tarkvaraarenduse printsiipt on inimesed ja nendevaheline koostöö. Seejuures töökulgu tellija ning arendaja vahel peab olema tihe ja paindlik, mis tähendab igapäevast koostööd kogu projekti vältel. Keskenduda tuleb väärtusele, mida luuakse kliendile, selle asemel, et olla kinni meeskonna rollides. Pidev koostöö peab olema teostatav näost näkku suhtlemise abil igapäevaste või iganädalaste kohtumiste kaudu. Pideva koostöö eesmärgiks on tagada tellijale kõrgel tasemel väärtuspakkumise võimekus, pakkudes talle vajalikku tarkvara võimalikult kiiresti ja tihti. Vajaliku all mõeldakse, et arendajate poolt loodav tarkvara on just see, mida tellija soovib. Selle saavutamiseks ja ühtlasi kliendi rahulolu tagamiseks on tema tagasiside igas tarkvaraprotsessi faasis igati teretulnud, et saaks arvesse võtta ka arenduse käigus muutuvaid ärinõudeid. Oluliseks printsiiptiks on ka see, et muutustele reageerimine ning arengusuundade ja eesmärkide kohandamine on tähtsam, kui esialgse plaani järgimine. Selle printsiipti järgmise aluseks on just töötajate hoiakud ja käitumisviis ning nende motiveeritus, kompetentsid ja mõtteviis. Agiilne lähenemine eeldab, et töötajad on oma tööle pühendunud ning nad on iseseisvad ja distsiplineeritud. Selleks on tarvis luua ka vastav keskkond ning toetada ja usaldada töötajaid [15], [17].

2.2.2 Tehnikad

Andmekaitse ja infoturbe leksikoni kohaselt on tehnika võtete kogum ja sooritusviis [16]. Osa agiilse tarkvaraarenduse tehnikaid on kasutatavad erinevates agiilse tarkvaraarenduse metoodikates ja teine osa on kasutatavad ainult kindla metoodika juures. Universaalsete agiilsete tarkvaraarenduse tehnikate näideteks on igapäevased kiirkoosolekud (*Stand-up meetings*), kasutuslood (*User Story*) ja planeeritavate tööde nimekiri (*Product backlog*). Populaarseimad agiilse tarkvaraarenduse tehnikad on välja toodud Tabelis 1.

Tabel 1. Populaarsemad agiilse tarkvaraarenduse tehnikad

Allikad: autori koostatud, kasutades [15], [18], [19], [20].

| Tehnika nimetus | Kirjeldus |
|--|--|
| Kiirkoosolek (<i>Stand-up meetings</i>) | Antud tehnika on omane Scrum metoodikale. Tegu on igapäevase koosolekuga, mille eesmärk on meeskonnaliikmetel anda ülevaade eelmise päeva tehtust, arutada ülesse kerkinud probleemid, rääkida tulevastest tegevustest. Sellised koosolekud on osa iteratsioonidest ehk sprintidest (vt iteratsioon). Taolised koosolekud soodustavad tihedat koostööd kliendi ja arenduspartneri vahel ja näost näkku rääkimist ning lisaks kiiret tegutsemist. Koosoleku viib läbi kas Scrum meister (<i>Scrum Master</i>), või kui kasutusel ei ole Scrum metoodika, siis iteratsiooni või meeskonna juhi poolt. Koosoleku kestus on tavaliselt 15 minutit, kus osalevad kõik meeskonna liikmed. Koosoleku peamised teemad on progress ning takistused. |
| Iteratsioon (<i>Iteration</i>) | Iteratsioon on arenduse töösükkel, mis võib olla keskusega kuni 3 kuud. Tavaliselt jääb see vahemikku 1-4 nädalat. Selle jooksul arendatakse välja töötav tarkvara. Ühe iteratsiooni käigus teostatakse meeskonna poolt analüüs, disain, arendatakse, testitakse ja paigaldatakse töötav funktsionaalsus. Nimetust „iteratsioon“ kasutatakse XP metoodika juures, Scrum-s on see nimetusega „sprint“ ning DSDM metoodikas „ajakast“ (<i>timebox</i>). |
| Iteratsioonikoosolek | Koosoleku põhitähtsus seisneb meeskonna töökiirusel. Koosolek on mõeldud kõikidele osalistele, kes on huvitatud väljaarendatud tarkvarast. Nendeks on nii kliendi sponsorid, kui ka arendajad. Iga iteratsioon algab iteratsioonikoosolekuga, kus pannakse paika millised kasutuslood on lahendatavad millise iteratsiooni juures ning millised ülesanded, vastuvõtukriteeriumid, analüüs ning disain on iga kasutusloo juures. Igale ülesandele antakse ligikaudne hinnang töötundides. |
| Iteratsiooni ülevaade (ka sprindi ülevaade) (<i>Show & Tell</i>) | Tegemist on koosolekuga, kus arendusmeeskond tutvustab tehnilise detailsusega kas toote omanikule (<i>Product Owner</i>) või kliendile seda funktsionaalsust, mis oli eelmise iteratsiooniga teostatud. Funktsionaalsust näidatakse ette ning selgitatakse see lahti. See on koosolek, kus saadakse tagasisidet, kas teostatud funktsionaalsusega |

| | |
|------------------------|--|
| | <p>sai täidetud püstitatud eesmärk ning korjatakse mõtteid järgneva iteratsiooniks. Scrum-i vaatevinklist on Scrum meister see, kes tutvustab koosoleku agendat, toote omanik räägib eesmärgist arenduse vaatevinklist ning arendusmeeskond tutvustab tehtu.</p> |
| <p>Agiilne töölaud</p> | <p>Samuti seostatakse Kanbani visuaalse töölauaga. Töölaua mõiste on kasutusse tulnud <i>Lean</i> metodoloogiast, mis aitab visualiseerida ning juhtida käesolevat töövoogu. Töölaud on tõhus viis iteratsiooni või relüüsi (<i>Release</i>) jooksul ülevaate saamiseks. See on kommunikatsioonivahend, mis algatab suhtlust meeskonnaliikmete vahel. Töölaud näeb välja kui tabel, kus iga rida on kasutuslugu (<i>User story</i>), mis on võetud planeeritavate tööde nimekirjast <i>Product backlog</i>. Kasutuslugu on jagatud ülesanneteks (<i>Task</i>). Iga ülesanne on paigutatud eraldi veergu. On viis veergu:</p> <ol style="list-style-type: none"> 1. Lugu (<i>story</i>) – kasutusloo kirjeldus 2. Plaanimised tegevused (<i>To do</i>) – hulk ülesandeid, mida on vaja ära teha, et kasutuslugu oleks valmis 3. Ülesanne, mis on käsil (<i>Work in progress</i>) – siia tõstetakse ülesandeid plaanimise tegevuste veerust. Seda teeb kesiganes meeskonnaliige, kes ülesandega tegelema hakkab. Enne tegevuse algust räägitakse see ülesanne läbi kiirkoosolekul. 4. Kinnitamiseks ülesanded (<i>To verify</i>) – selles veerus on need ülesanded, mida tarvis testida. 5. Tehtud (<i>Done</i>) – kõik tehtud ülesanded tõstetakse lõpuks siia veergu üle. <p>Iga iteratsiooni käigus on teretulnud kõik muudatused. Nendeks on ärinõuete täiendused detailsuse väljaselgitamisel. Seega, töölaud iteratsiooni käigus võib ülesannete poolelt täienduda.</p> <p>Töölaua kasutamise jooksul võib välja tulla ka fakt, et mõne ülesande tegemisel ei jätku mõnel meeskonnaliikmel teadmisi ülesande teostamisel. Siinkohal saab tööülesannet üle võtta muu meeskonnaliige. See on hea võtte teadmiste jagamisel.</p> |
| <p>Kanban töölaud</p> | <p>Töölaud, mis aitab saada ülevaadet töövoost ning tekkida võivatest pudelikaeladest. Nimelt, töölaud on jagatud tulpadeks, kus igas tulpas võib olla teatud arv ülesandeid:</p> <ol style="list-style-type: none"> 1. Planeeritavate tööde nimekiri (<i>Backlog</i>) – korruga võib siin olla 6 kasutuslugu. 2. Analüüs (<i>Analyse</i>) – see tulp sisaldab 3 ülesannet korruga. Tulp on jagatud kaheks alamtulbaks, kus ühes on need ülesanded, mis tegemisel ja teises, mis tehtud. 3. Arendus (<i>Development</i>) - see tulp sisaldab 4 ülesannet korruga. Tulp on jagatud kaheks alamtulbaks, kus ühes on need ülesanded, mis tegemisel ja teises, mis tehtud. |

| | |
|--|--|
| | <p>4. Testimine (<i>Test</i>) – kui testimistulpa on tekkinud 3 ülesannet, siis see viitab pudelikaelale, kus on tarvis tegutseda. Enne ei alustata teiste ülesannetega, kui testimiseülesandeid on alla 3.</p> <p>5. Paigaldamine (<i>Deploy</i>) – testitud arendusülesannete lõpppunkt. Antud töölaua eelis seisneb selles, et tekkinud pudelikaelad soodustavad meeskonnaliikmetel veelgi efektiivsemat omavahelist suhtlemist.</p> |
| Langustrendi graafik (<i>Burn down chart</i>) | <p>Graafiku põhimõte on näidata, mis tööd on veel tarvis meeskonnal ära teha, mitte aga, mis töö sai tehtud. Y teljel on kasutuslood, mis tarvis iteratsiooni käigus ära teha ning X teljel on aeg päevades. Kui kasutuslugu on lahendatud, lisatakse see graafikule, mis annab ülevaate sellest, palju veel jäänud teha. Igal kasutuslool on oma punktiskoor. See on arv, mis on eelnevalt meeskonna poolt hinnatud (vt <i>Planeerimipokker</i>). Selleks, et graafikut oleks efektiivne kasutada, peaksid kasutuslood olema väikesed. Kasutuslugude asemel võivad olla ülesanded. Ülesandeid määratakse tundides. Kui ülesanne lahendatud, lisatakse see graafikule. Ei ole soovitatav määrata ülesandeid, mis on 1 tunnised või väiksemad ning suuremad kui 12 tundi.</p> |
| Kasutuslugu (<i>User story</i>) | <p>Kasutuslugu kirjeldab süsteemi käitumist: kuidas süsteem peaks suhtlema tegutsejaga (<i>Actor</i>). Tegutseja alustab süsteemiga suhtlust, et täita mingit eesmärki (<i>Goal</i>). Süsteem peab vastama selliselt, et tulemus oleks kooskõlas tegutsejate, laiemalt organisatsiooni huvidega. Erinev käitumine võib sama kasutusloo puhul kaasa tuua erinevad stsenaariumid (<i>Scenario</i>), mis sõltuvad tegutseja päringutest ja süsteemi reaktsioonidest. Kasutusloo puhul on olulised järgmised mõisted:</p> <ol style="list-style-type: none"> 1. tegutseja (<i>Actor</i>) – keegi, kes omab käitumist, nt. inimene (roll), organisatsioon või süsteem. 2. huvigrupp (<i>Stakeholder</i>) – keegi, kellel on huvi süsteemi käitumise vastu. 3. põhitegutseja (<i>Primary actor</i>) – keegi huvigrupist, kes käivitab süsteemiga suhtluse, et täita oma eesmärgi. 4. skoop (<i>Scope</i>) – ütleb, mis on täpselt kirjeldatavaks süsteemiks. |
| Planeeritavate tööde nimekiri (<i>Product backlog</i>) | <p>Nimetus on tulnud Scrum metoodikast. Kuna kõik projektid ei ole seotud tootega, vaid mõned teenusega või protsessi täiustamisega, siis nimekirja nimetuseks võib olla samuti „planeeritavate lahenduste nimekiri“. Tööde nimekiri on süsteemi prioritseeritud nimekiri funktsionaalsustest, mis on nii kliendikesksed, kui ka tehnilised. Tööde nimekirja võib täiendada vastavalt kliendiga läbirääkimiste käigus tulnud täiendustega või lisaosovidega. Tööde nimekirja haldajaks on toote omanik, kuid ülesandeid toodetakse arendusmeeskonna poolt. Iga kord, kui planeeritavate tööde nimekiri täieneb, tööd vaadatakse uuesti üle ning antakse hinnangut. Hinnangu andmisel kasutatakse tavaliselt MoSCoW meetodit. Kõik planeeritud tööd liiguvad kord</p> |

| | |
|--|---|
| | tööde väljalaskme nimekirja (<i>Release backlog</i>) ning lõpuks jõuavad sprindi tööde nimekirja (<i>Sprint backlog</i>). |
| Planeerimispokker (<i>Estimating with planning poker</i>) | <p>Iga iteratsioon sisaldab kasutuslugusid, mis antud töötsükli jooksul on planeeritud ära teha. Iga kasutuslugu peaks olema enne meeskonna poolt läbi räägitud ja ära hinnatud. Hindamiseks on mitu lähenemist, kas:</p> <ol style="list-style-type: none"> 1. formaalne, kus hinnangut antakse tuginedes lahendustele, mis olid varem tehtud 2. eksperthinnang, kus funktsionaalsust arutatakse läbi ning otsustatakse arvamuse põhjal 3. kahe eelneva kombineeritud hinnang <p>Agiilne lähenemine kasutab hindamiseks pokkeri kaarte. Nimelt, tegemist on rühmatööga, kus liikmeteks on rühmatöö juht, kliendi esindaja, kas siis toote või äri omanik, arendusmeeskond. Valitakse kasutuslugu, kus kliendi esindaja teeb lühiülevaate. Arendusmeeskond esitab lisaküsimusi, et oletused ja riskid oleksid välja selgitatud. Seejärel valib arendusmeeskond ühe kaardi. Suurema ja väiksema arvuga kaardid arutatakse läbi. Protsessi korratakse, et kõik meeskonnaliikmed oleksid tulemustega päri. Numbrid, mis saadakse igale kasutusloole, võib hinnata kui päevi või inimest päevas või kasutusloo punktina.</p> |
| Retrospektiiv ehk tagasivaade (<i>Retrospective</i>) | <p>Koosolek, mida viiakse läbi iga iteratsiooni järgselt. Intensiivsemaid retrospektiive võib samuti läbi viia kas projekti lõpus või lõpliku tarkvara paigaldusejärgselt. Retrospektiiv on see koht, kus saab vaadata tagasi tehtule, tunnistada vigu ning nendest õppida. Põhifookus on töö tegemisel ning inimestevahelisel suhtlusel. Retrospektiiv koosneb viiest sammust:</p> <ol style="list-style-type: none"> 1. Ettevalmistus (<i>Set the stage</i>), mille eesmärk on mugava keskkonna loomine selleks, et meeskonnaliikmed tunneksid ennast avatuna ning kaasatud. 2. Info kogumise (<i>Gather data</i>) mõte seisneb selles, et meeskonnaliikmete uskumustest ning arusaamadest kooruks välja suur pilt, mis avardaks ka teiste vaadet. Läbirääkimise laual on nii sündmused, mõõdikud, valmis saadud kasutuslood, täiustatud kood. 3. Teadmiste loomise (<i>Generate insights</i>) juures arutatakse kogutud infot ning arutatakse miks oli see nõnda tehtud ning kuidas saaks parandada ning järgmisel korral teha paremini. 4. Järgneva on valitud põhiteemad ning koos mõeldakse mis nendega edasi teha (<i>Decide what to do</i>). Igast iteratsioonist võetakse tavaliselt kuni 2 põhiteemat. 5. Lõppsõnad (<i>Closing the retrospective</i>) – mis retrospektiivis meeldis, mida soovitakse järgmiseks retrospektiiviks muuta ning tunnustama meeskonnaliikmeid. |

2.2.3 Metoodikad

Käesolev alajaotis annab ülevaate kõige populaarsematest agiilsetest tarkvaraarenduse metoodikatest – XP, SCRUM, LSD ja DSDM. Iga metoodika kohta tuuakse välja kirjeldus, protsess ja selles osalevad rollid, kasutatavad agiilse tarkvaraarenduse tehnikad ning metoodika hinnang.

2.2.3.1 Scrum

Tegemist on arendusmetoodikaga, mis aitab juhtida projekte. Sellel on kindlad juhised, mis on suunatud toote kiirele ja pidevale tarnimisele. Ühest küljest on Scrum-i protsess, printsiibid ja reeglid sirgjoonelised ning arusaadavad. Teisest küljest võib niisugune lihtsus olla petlik, sest tegemist ei ole ettekirjutatud protsessiga, mis ütleb ette mida igas antud olukorras ette võtta. Vastupidiselt sellele on Scrum loodud eeskätt tarkvaraarenduse projektide jaoks, kus ei ole võimalik kõike ette aimata majandusturu, ühiskondlike ja tehnoloogia muutuste osas. Läbi muutuste õpitakse, leitakse probleemidele lahendusi ning jagatakse teadmisi meeskonna liikmete vahel. Scrumi keskmes on intuiitvne lähenemine probleemile, inimestevaheline suhtlus ja vajaliku väärtuse pakkumine töötava tarkvara näol. Metoodika etapid on meeskonnale nähtavad ning täpselt on teada, mis igal ajahetkel projektis aset leiab [15], [17], [21].

2.2.3.1.1 Protsess ja rollid

Scrum-s eristatakse järgmist kolme (3) faasi:

1. Esimene faas algab kliendi ja arendusmeeskonna vahelise koosolekuga, kus pannakse paika arendatava tarkvarasüsteemi eesmärgid, disain ja arhitektuur ning need nõuded, mida sprindis arendama hakatakse.
2. Arendusfaasis leiavad aset 15 minutilised kiirkoosolekud, kus osalejateks on arendusmeeskond ning klient.
3. Kolmandas faasis toimub sprindi ülevaade. Antud faasi eesmärk on õppida sprindi jooksul tekkinud vigadest, et tulevikus nendega arvestada [17], [22].

Scrum-s saab kaardistada viite (5) rolli:

1. Scrum meister (*Scrum Master*), kelle põhiülesandeks on lahendada ülessekerkivaid takistusi ning probleeme arendusmeeskonna ning toote omaniku (*Product Owner*) vahel.

2. Toote omanik, kellel peamiselt vastutada planeeritavate tööde nimekiri (*Product backlog*).
3. Arendusmeeskond, tavaliselt 5-9-st inimesest koosnev motiveeritud, kompetentse ning iseorganiseeruv.
4. Klient – tema ülesanne on anda sisendit planeeritavate tööde listi koostamiseks, teavitada arendusmeeskonda muutustest ning kiiresti reageerida ülessekerkivatele probleemidele/takistustele.
5. Juhtkond – lõppotsuste, õiguslike regulatsioonide, standardite jm eest vastutab juhtkond. Samuti võtab ta osa eesmärkide püstitamisel ning nõuete määramisel. [17], [22].

2.2.3.1.2 Metoodika hinnang

Scrum-i on lihtne kasutada väikeses kuni 9-liikmelises arendusmeeskonnas. Kui tekib ebaselgusi, siis kiired arendamine, prototüüp ja testimine annavad selguse. Tänu lühikestele sprintidele ja regulaarsetele koosolekutele saab arendusmeeskond kliendilt kiiret tagasisidet ning on võimalik kiiresti sisse viia muudatusettepanakuid. Probleemid võivad aga esile kerkida siis, kui klient ja mud võimalikud huvigrupid ei ole kaasatud ja meeskonnaliikmed ei ole motiveeritud ega kompetentsed. Probleemiks võib kujuneda ka see, et planeeritud tööde teostamise juures võib kaduda üldine ülevaade arendatavast tootest, mis omarda tähendab, et oodatav äriväärtus võib jääda saamata [17], [22].

2.2.3.2 XP

Metoodika XP keskendub eelkõige tehnilistele aspektidele nagu paarisprogrammeerimine, ühiktestid (*Unit tests*, arvutiprogrammi väikseimate osade testid), loodava programmi tarnimine väikeste osadena, pidev suhtlus kliendi ja arendajate vahel ning kliendi kiire reageerimine. XP eesmärgiks on tarnida kliendile funktsionaalsust, mis loob kliendile väärtust ning luua minimaalne elujõuline toode (*Minimal viable product*). Selle tagamiseks on meeskonna liikmed kõrge kvalifikatsiooni ja kogemustega. Kõik kliendi poolt ettepanud muudatused on metoodikas XP teretulnud, sest nii tagatakse kliendile just seda, mida ta vajab [15], [23].

2.2.3.2.1 Protsess ja rollid

XP-d võib jagada viite (5) faasi:

1. Uurimine (*Exploration*) – klient tagab arendusmeeskonnale kasutuslood, mida ta soovib toote esimeses versioonis (*Release*) näha. Selles faasis tutvutakse tööriistadega, tehnoloogiatega ning praktikatega, mida ta selle projekti jooksul kasutama hakkab.
2. Planeerimine (*Planning*) – see on koht, kus arendusmeeskond hindab ära kasutuslood ning seejärel seatakse paika esimese versiooni ajaplaan.
3. Iteratsioon (*Iterations to Release*) – ajaplaan, mis planeerimisfaasis sai paika pandud, tükeldatakse iteratsioonideks. Need kestavad 1-4 nädalat. Esimese iteratsiooni käigus valitakse need kasutuslood, mille põhjal saab välja arendada süsteemi arhitektuur. Järgnevatel iteratsioonidel kasutuslugude valikul saab sõna klient. Viimasel iteratsioonil on süsteem tootmiseks valmis.
4. Tootmine (*Productionizing*) – enne kliendile toote presenteerimist, teha testimine ja kontroll. Tootmisfaasis võivad tulla välja muudatused, millega tarvis arvestada. Kõik välkideed ning soovitud dokumenteeritakse ning need võetakse arvesse näiteks hooldusfaasis.
5. Hooldus (*Maintenance*) – niinimetatud klienditugi juba valmis saanud funktsionaalsusele, mis on paralleelne järgmiste iteratsioonidega.
6. Surm (*Death*) – see on faas, kus enam ei testita, ei viida sisse muudatusi ei disaini arhitektuuri ega koodi, ega kliendi poolt ole teostamisele anda juurde kasutuslugusid. See on koht, kus kas klient on tootega rahul või vastupidi. Vajalik dokumentatsioon luuakse selles faasis [15], [23].

XP-s saab kaardistada seitset (7) rolli:

1. Arendaja, kelle ülesandeks on lihtsasti loetava koodi loomine vastavalt ettekirjutatud kasutuslugudele, testimine.
2. Klient, kelle ülesandeks on kasutuslugude loomine ning prioritseerimine ja funktsionaalsete testide kirjutamine.
3. Konsultant on isik, kes suunab ning otsib lahendusi tehnilistele probleemidele, mis võivad ülesse kerkida.
4. Testija eesmärgiks on abistada klienti funktsionaalsete testide, nimelt kasutuslugude vastuvõtutestide (*acceptance testing*) kirjutamisel.

5. Jälgija (*Tracker*) – tema ülesandeks on mõõta ning tutvustada meeskonna edukust.
6. Treener (*Coach*) vastutab terve protsessi eest.
7. Juhhi roll on vastu võtta otsuseid [15], [23].

2.2.3.2.2 Metoodika hinnang

XP on küll paindlik ja kergekaaluline metoodika, kuid XP tarkvaraprotsess sõltub organisatsiooni äriprotsessidest, mis tähendab, et kui organisatsioon tervikuna kõnealust metoodikat vastu ei võta, siis võib selle rakendamine ebaõnnestuda. XP rakendamise puhul ei ole arendajatel vaja kirjutada tihti tarbetut dokumentatsiooni vaid nad saavad keskenduda koodi kirjutamisele ja kiirele toote tarnimisele. Paraku võib aga siin tekkida oht, et arendajad ei omanda oma töö käigus kõiki XP praktikaid ning muudatuste halduse puudumine ja vähene dokumentatsioon eeldab arendajalt kliendi rolli omandamist, mis aga võib osutada probleemiks näiteks arendusmeeskonna koosseisu muutumisel [15], [23], [24].

2.2.3.3 DSDM

DSDM ehk dünaamiline süsteemiarendusmetoodika on agiilsetest tarkvararenduse lähenemistest üks kõige põhjalikum. Selle rajajaks on DSDM-konsortsium Suurbritanniast, millesse kuuluvad Ühendkuningriigi suurimad IT-teenuste pakkujad. DSDM-i kasutamiseks peab läbima koolitusprogrammi ja lisaks on lähenemisel oma käsiraamat, mille tutvumisversioon on kättesaadav DSDM-konsortsiumi kodulehel [24], [25].

DSDM-i kohaselt on mõistlik esmalt keskenduda aja ja ressursside parandamisele ning seejärel järk-järgult kohandada toote funktsionaalsust. Edu saavutamine seisneb prototüüpimises, kus kõik muudatused ning täiendused on vältimatud ja neid viiakse arndusprotsessi sisse tagasiulatuvalt. Iga etapp peab olema piisavalt detailne, et saaks alustada järgneva etapiga. Eelnevaid etappe on võimalik lõpetada järgnevate etappide käigus.

DSDM-i kohaselt saavutab projekt edukuse, kui paigas on selged ärilised eesmärgid, tarded on tihedad ja meeskonnaliikmed on motiveeritud ning neile on tagatud kõik

volitused tegutsemiseks ja otsustamiseks. DSDM-i kohaselt võib ühes projektis olla mitu meeskonda, kuhu kuulub kuni kuus (6) inimest [15], [25], [26].

2.2.3.3.1 Protsessid ja rollid

DSDM koosneb järgnevatest faasidest:

1. Eelprojekt (*Pre-project*) – etapp, mis tagab, et ainult õiged projektid selgete eesmärkidega oleksid valitud.
2. Teostatavus (*Feasibility*) – antud projektis toimub väljavalitud projektide hindamine ning sobivus.
3. Vundament (*Foundations*) – siin etapis määratakse töö skoop, selle tegevused, rollid ning ajaplaan – projektiplaan.
4. Järk-järguline arendamine (*Evolutionary development*) – selles etapis rakendab arenduse meeskond järgnevaid praktikaid: iteratiivne arendamine, ajakast ning MoSCoW meetod nõuete prioritseerimisel.
5. Paigaldamine (*Deployment*) – faasi põhitegevusteks selles faasis on toote osi kokku panna, neid üle vaadata ning paigaldada.
6. Projekti tagasivaade ehk *retro* (*Post-project*) – faas, kus tehakse ülevaade projekti kulgemise kohta nii tehnilisest, protsessi ja ärilisest vaatevinklist. [15], [26].

DSDM-i rolle saab jagada nelja kategooriasse:

1. kliendi liikmed – sponsor, visionäär, nõustaja, saadik, ärianalüütik
 2. projekti liikmed – projektijuht, meeskonnajuht
 3. tehnilised liikmed – koordinaator, ärianalüütik, arendaja, testija, nõustaja
- protsessi liikmed – metoodika treener ja juhendaja [15], [26].

2.2.3.3.2 Metoodika hinnang

DSDM on väga paindliku protsessiga lähenemine. Ta küll katab suure osa arendusprotsessist, kuid ei ütle ette, kuidas arendamine peaks täpselt välja nägema. Arendusprotsessi paikapanemine on arendusmeeskonna liikmete ülesandeks. Kui aga meeskonnaliikmed ei ole piisavalt kaasatud arendusprotsessi või ei ole kompetentsed, siis on suur oht ebaõnnestuda. Teistest agiilsetest metoodikatest erineb DSDM selle poolest, et omanikuks on konsortsium ja metoodikat saab kasutada ainult konsortsiumi poolt

väljaantud litsentsi alusel. Litsentsi eest tasutakse aastamaksu näol, mis võimaldab ligipääsu metoodika täiskirjeldusele. Konsortsium viib läbi koolitusi, tagab kasutajatoe, tegeleb metoodika edasiarendamisega ja vastutab dokumentatsiooni ajakohasuse eest. [15], [25], [26].

DSDM-i peamine fookus on ajakastil, mis võimaldab kiirelt ja operatiivselt tarnida kliendile väärtust pakkuvat toodet, mis vastab tema vajadustele. Samas on niisugune lähenemine pigem projektijuhtimise raamistik kui tarkvaraprotsessi organiseerimise viis. DSDM-i projektis on oluline roll kliendi osalusel tomuval iteratiivsel arendusel, mis tähendab, et progress peab olema osapooltele selgesti jälgitav ja arusaadav. DSDM-i kasutamine on raskendatud ülikriitilise tarkvara arenduse puhul, kus on vajalik põhjalik tarkvara spetsifikatsioon, mille alusel toimub valideerimine ja verifitseerimine [15], [25], [26].

2.2.3.4 Lean metodoloogia ehk LSD

Lean-i peamiseks põhimõteteks on raiskamiste ja üleliigse kõrvaldamine (*Eliminating waste*) ning kliendile väärtuse lisamine. Lean-i lähenemise kohaselt ei tohi mitte miski, mis ei tooda kliendi jaoks väärtust, jõuda lõplikku tootesse. Antud lähenemine keskendub kogu arendustsüklile – tellimuse saamisest kuni arve esitamiseni. Lean-i kohaselt ebaefektiivsed tegevused eemaldatakse arendusprotsessist. Eesmärkide saavutamiseks kasutatakse mitut tehnikat ja meetodit, mida iseloomustavad järgmised omadused:

1. lihtne leida probleeme,
2. lihtne lahendada probleeme,
3. lihtne õppida vigadest.

Lean on alguse saanud tootmise valdkonnast, mis tähendab, et see metoodika ei pruugi sobida muudele valdkonadele nagu näiteks näiteks teenindus [15], [27], [28].

2.2.3.4.1 Protsess ja rollid

LSD ei keskendu sellele, kuidas tarkvara arendada. Antud lähenemine on tavaliselt kombineeritud teiste metoodikatega või praktikaga, sest selle põhiohk on järjepidev õppimine ning täiustamine. Üks täiendustest võib olla Kanban-i praktika, mis pakub

korrapärasest töövoogu. Ka Scrum, XP, DSDM või mõnda muud metoodikat võib täiendada LSD-ga.

Sellel põhjal seab antud lähenemine ette 5 (viis) põhiprintsiipi:

1. Määratleda kliendiväärtus ning arendatava lahenduse väärtus koos selle võimaluste ja ajaga.
2. Määratleda väärtusvoog ning teha selgeks selle etapid, kõrvaldades need etapid, mis väärtust ei too.
3. Muuta väärtust toovad etapid pingeliseks jadaks, mis võimaldaks lahenduse sujuvat tarnimist kliendile.
4. Tarnida ainult vastavalt kliendi soovile, kus ressursside kasutamine on vastavuses kliendi nõudmistega.
5. Korrata kõik ülesse loetletud sammud saavutamaks täiuslikkust, kõrvaldades üleliigsust [15], [29].

2.2.3.4.2 Metoodika hinnang

LSD-i lähenemine on küll pühendunud üleliigsuse vähendamisele, protseduuride lihtsustamisele ning tootmise ja tarnimise kiirendamisele, kuid paraku on seda metoodikat raske eraldiseisvalt kasutada. Metoodikat võib võtta kui hulk tööriiste, süsteemi või filosoofiat. LSD on suunatud tarkvaraprotsessi täiustamisele, mille saavutamiseks pakutakse tehnikaid aga mitte tarkvara arendamisele või töökultuuri väljakujundamisele esilekerkivate probleemide lahendamiseks. Kiire arendustööde tarnimine on küll LSD eelis, sest niimoodi saab toota rohkem funktsionaalsust, kuid arendamise tegelik kiirus sõltub siiski kliendi või huvigruppide osalemisest ja otsustest [15], [30].

2.2.3.5 Metoodikate erinevad rõhuasetused ja ühised omadused

Võrreldud metoodikad erinevad üksteisest nii ulatuse kui ka detailsuse poolest. Metoodikate rakendatavusel on erinevates olukordades erinevad piirangud. Näiteks Scrum keskendub peamiselt iteratsioonide juhtimisele ja projektihaldusele aga XP arendusprotsess määrab kindlaks detailsed praktikad, mida meeskond peab järgima. Võrreldes teiste metoodikatega ei seta XP rõhku tarkvara spetsifitseerimisele ega dokumenteerimisele aga kasutab XP mitmeid täpselt defineeritud arendustehnikaid nagu

näiteks paariprogrammeerimine ja testimise poolt juhitud arendus (*Test-driven development*) [15], [24].

Agiilsete metoodikate rakendamise kitsenduste osas saab välja tuua loodava tarkvara kriitilisuse, kus näiteks eriti DSDM-i puhul on metoodika rakendamine raskendatud kriitilise töökindlusega tarkvara arendusprojektides [24].

Agiilsed metoodikad põhnevad iteratiivsel arendusprotsessil, kusjuures iteratsiooni pikkus varieerub tugevalt. Lühim iteratsioon on XP-1, mis tähendab riskide kiiremat maandamist ja vahetumat tagasisidet kliendilt. Seetõttu on klient alati kaasatud XP arendusprotsessiga ja kliendi esindajad on arendusmeeskonna igapäevased liikmed [24].

Ühiste omaduste osas sooviks autor rõhutada seda, et nõuete kogumine erinevates metoodikates on sarnane selles mõttes, et nõudeid kogutakse ja esitatakse väikeste ühikutena nagu näiteks kasutuslugudena (*User stories*) [24].

Erinevate agiilsete metoodikate sarnasused ja erisused on välja toodud Tabelis 2.

Tabel 2. Agiilsete metoodikate erinevad rõhuasetused ja ühised omadused

Allikad: autori koostatud, kasutades sisendiks [7], [15], [17], [31].

| | Scrum | XP | DSDM | Lean |
|--|--------------------------------|--|---|---|
| Metoodika sobivus tarkvara arendusprojektidele või äriprojektidele | Põhirõhk arenduse juhtimisel | Põhirõhk arenduse meetoditel või praktikatel | Metoodika on pigem kohaldatav äriprojektidele, kui arendusprojektidele | Arendusprojektid |
| Projektide suurus | Väikestele arendusprojektidele | Väikesed või keskmised arendusprojektid | Metoodika sobib pigem arendamiseks suuri süsteeme, mille kestusperiood võib ulatuda kuni 3 aasta. Sellisel juhul on projektis mitu meeskonda. Võimalik rakendada ka | Nii väikestele (3-6 kuud kestvusega), kui ka suurtele aastate kestvusega projektidele |

| | | | | |
|--|---|--|---|--|
| | | | väikestele projektidele. | |
| Käsitleb valitsemist laiemalt kui ainult tarkvaraarendus protsessi | Ei | Ei | Jah. Valitsemise eelduseks on defineeritud rollid ning paika pandud struktuur. | Kui kasutusele võetakse kas Scrum või XP metoodika, siis tagatakse pidev täiustamine ja väärtusvoog. |
| Defineeritud rollid | Jah | Jah | Jah | Ei |
| Keskendub inimestele või protsessidele | Inimestele | Inimestele | Protsessile | Protsessile |
| Meeskonna suurus | Väike meeskond, 5-9 inimest. Kui on võimalik kaasata rohkem inimesi, siis saab moodustada mitu meeskonda. | Väike meeskond, 3-20 inimest. | Väike meeskond, 2-6 inimest. Ühes projektis võib olla mitu meeskonda, mis võib ulatuda kuni 100 arendajani. | Ei ole defineeritud. |
| Tehnoloogia tähtsus | Madal | Kõrge | Sõltub millise metoodikaga koos kasutatakse. | Sõltub millise metoodikaga koos kasutatakse. |
| Dokumenteerimise tähtsus | Keskmine | Madal | Keskmine | Sõltub millise metoodikaga koos kasutatakse. |
| Arendustehnikate tähtsus | Madala | Kõrge | Madala | Madal |
| Kliendi pidev kaasatus | Jah | Jah | Jah | Ei |
| Algatus- ning lõpetamisfaas | Alustatakse sprindi planeerimisega ning lõpetatakse sprindi ülevaatega. | Alustatakse kasutuslugudega ning lõpetatakse retrospektiiviga. | Projekti alguses pannakse paika vaid üldine eesmärk ja projekti ulatus ning ei „külmutata“ kohe kõiki | Sõltub millise metoodikaga koos kasutatakse. |

| | | | | |
|---|---|---|--|---|
| | | | nõudeid. Arendusprotsess i käigus määratakse nõuded detailsemalt, kuid projekti ulatus ei tohiks seejuures enam oluliselt muutuda. | |
| Lühikesed lahenduse tarned | Jah | Jah | Jah | Jah, kui kasutatakse koos Scrum-i või XP-ga. |
| Iteratiivne ja järg-järguline arendusprotsess | Jah | Jah | Jah | Jah, kui kasutatakse koos Scrum-i või XP-ga. |
| Ette defineeritud arendusprotsess | Jah | Jah | Jah, kui kasutatakse koos Scrum-i või XP- ga. | Jah, kui kasutatakse koos Scrum-i või XP-ga. |
| Muudatustega arvestamine | Jah, kui järgnevates iteratsioonides. | Jah, põhineb pideval koodi ülevaatusel. | Jah, muudatusi võib detailides teha, kuid nõuded on paigas enne arenduse alustamist. | Jah ning nii hilja, kui võimalik |
| Testimine | Jah | Jah, testimisel põhinev arendus. | Jah | Jah, testimine läbi kogu elutsükli. |
| Järjepidev õppimine ning täiustamine | Jah | Jah | Jah | Jah |

2.2.3.6 Maailma ja Eesti praktikad

Ainuüksi mõte, et agiilne lähenemine tuleb omaks võtta, võib töötajatel põhjustada psühholoogilist stressi. Agiilse lähenemisega on samuti raske kohaneda, kui eelnev projekti juhtimisstiil on põhinenud kontrollil. Siit tuleneb ka töötajate vähene motiveeritus omaks võtta uuendusi nagu agiilne lähenemine. Selle juures on väga olulisel kohal juhtkonna toetus ja eestvedamine. Muudatuste läbiviimiseks on kõigepealt vaja arusaadavat visiooni. Juht on muudatuste protsesi strateegiliseks eestvedajaks, tehes organisatsiooni strateegilise visiooni töötajatele arusaadavaks ning motiveerides inimesi kaasa tulema, pakkudes töötajatele väljaõpet ja koolitusi ning luues suhteid arenduspartneri ja huvigruppide vahel. Samuti tuleb tegeleda agiilse mõtteviisi ja väärtuste juurutamisega ning omaks võtmisega juhtimise eri tasanditel ning alles siis hakata agiilset lähenemist praktikasse rakendama [32], [33].

Nuottila et. al. uuringust [32] selgub, et väikesed organisatsioonid kohanduvad agiilsete muudatustega paremini kui suured organisatsioonid ja seda just seetõttu, et suures ettevõttes ei saa projekt olla täiesti iseseisev, sest omab organisatsioonisiseseid seoseid teiste projektide ja meeskondadega ning seda eriti olukorras, kus teised projektid ei järgi agiilset lähenemist. Tavaliselt töötab suures organisatsioonis ühe projektiga mitu meeskonda, mistõttu inimestevaheline pidev kommunikatsioon, nagu seda eeldab agiilne lähenemine, ei pruugi töötada. Samuti on suurtes organisatsioonides muudatuste kontrollnõukogu süsteemsete või arhitektuuriliste muudatuste läbiviimiseks, mis paraku vähendab paindlikkust ning kliendile tajutavat väärtuse pakkumist. Mis puudutab avalikku sektorit, siis agiilset lähenemist siin peaaegu üldse ei rakendata. Nuottila et al. poolt läbiviidu uuringu [32] raames viidi ühes Soome avaliku sektori agentuuris läbi küsitlus, mille tulemusena selgus, et kui avalik organisatsioon soovib võtta kasutusele agiilse tarkvaraarenduse meetodika, siis peaks muutma hankepoliitikat ning õiguslikke regulatsioone nii, et need seda võimaldaksid. Lisaks peaks sisse viima arusaadavat rollide jaotust. Samuti peaks juhtkond olema valmis muudatusi juhtima, klient kaasatud, töötajad kogemustega oma töösse pühendunud. Kuna avaliku sektori süsteemid on loomu poolest suured ja keerulised, siis võtab innovatsiooniga kaasaskäimine avalikus sektoris paraku omajagu aega [32]

Aastal 2017 viis KPMG [34] läbi küsitluse 62 Hollandi ning Belgia organisatsiooni seas, välja selgitamaks agiilse lähenemise rakendamise edukust. 11% küsitletutest olid

avalikust sektorist. Küsitlusest selgus, et paljud organisatsioonid kasutavad oma projektides agiilseid meetodikaid, mis aitavad sisse viia muutusi IT süsteemidesse. Traditsioonilise lähenemise kasutamisele siiski jätkatakse, kuid seda hübriidselt, nt Water-Scrum ehk Kose- ja Scrum-metoodikate kooskasutamisega. Küsitlusest selgus, et populaarseim meetodika on Scrum. KPMG poolt läbiviidu küsitlus [34] toob samuti välja, et põhilisteks väljakutseteks agiilsuse kasutamisel on juhtimine ja organisatsiooniline küpsus.

Samal aastal viis KPMG [35] läbi ka ülemaailmalise küsitluse, milles osales 188 vastanut. Vastustest [35] selgus, et Scrum-metodika oma 40%-ga on kõige populaarsem ja et ainult 10% organisatsioonidest kasutab XP- ning 11% LSD-metoodikat.

Medvedska ning Berzisa oma 2015. aastal avaldatud artiklis [36], et Kose mudel, Scrum ja DSDM on ühed enim kasutatavamad meetodikad avaliku sektori organisatsioonides. DSDM-i puhul on tegu uue lähenemisega, mille parimaid praktikaid leidub vähe. Lisaks on tegemist keerulise ning raskestimõistetava meetodikaga.

Eesti avaliku sektori organisatsioonidest on agiilsete meetodikate eesrindlikuks rakendajaks olnud SMIT. SMIT-i ajakirjas 2016. [37] aastal ilmunud ühest artiklist selgub, et organisatsioon on võtnud kasutusse *Lean* lähenemise tarkvaraprotsessile. Siseministeriumi poolt tellitud PWC aruandest [8], mis sai valmis aastal 2018, saab teada, et SMIT kasutab oma arendusprotsessis Scrum/Kanban lähenemist. Viimatimainitud uuringu [8] käigus on läbi viidud organisatsioonisisene küsitlus, mille eesmärgiks oli saada hinnang valitud arendusmeetodikate juurdumisest. Küsitluse tulemustena toodi välja järgmised peamised probleemid:

1. Arendustöö on vähe automatiseeritud, mis tähendab, et vanemate süsteemide puhul ei ole näiteks versioonihaldus lõpuni automatiseeritud.
2. Arendustöö on vähe automatiseeritud, mis tähendab, et vanemate süsteemide puhul on rohkem manuaalset tegevust, mille tõttu täisautomatiseeritud tarnimine pole võimalik."
3. Ei viida läbi regulaarseid retrospektiive.
4. Meeskonnatöö ei ole viidud otsesesse vastavusse konkreetsete äriliste mõõdikutega.
5. Ligikvadi pooled vastajatest avaldasid arvamust, et arendusmeeskonnad on enestele ise kehtestanud kvaliteedi, koostöö ja meistrioskuste standardid, millest järeldub, et tiimide töö kvaliteet ei ole tugevalt juhitud.

6. Klient on küll kaasatud esialgse visiooni loomise faasis, kuid vähe kõigis järgnevates arendusprotsessi sammudes, mis tähendab, et tarkvaraarendamisel võib jääda saamata kliendi tagasiside, mistõttu lõpptoode ei pruugi vastata kliendi vajadusele.

Küsitluse läbiviija hinnangul esilekerkinud probleemid ei tundu olevat kriitilised, mistõttu agiilsete arendusmetoodikate rakendamist võib pidada rahuldavaks.

Taani Ärikorralduse Amet soovis aastal 2009 digitaliseerida tol hetkel paber kandjal põhinevat firma registreerimisprotsessi, valides traditsioonilise tarkvaarenduse metoodika – Kose mudeli. Kahe aasta möödudes selgus, et antud metoodika ei viinud soovitud lahenduseni, kusjuures peamiseks takistuseks olid pidevalt muutuvad nõuded ja aeglane otsuste vastuvõtmine. Aastal 2011 otsustati agiilse metoodika Scrum kasuks. Antud metoodika kandis vilja ja 2014. aastaks oli saavutati kliendile vastuvõetav tulemus. Antud projektis saavutati agiilse lähenemise rakendamise järgmist:

1. kliendikesksus,
2. tugev juhtimine ja kiire otsustamine,
3. IT arhitektuur, mis võimaldab järg-järgulist süsteemi muutmist,
4. selge süsteemi arendamise tegevuskava,
5. meeskonnaliikmete vaheline usaldus [38].

Rootsi ning Hispaania uuringutest [30], [39] selgub, et avaliku sektori organisatsioonid on oma arendusprotsessides rakendanud *Lean* lähenemist. Rakendamisel on peamine rõhk olnud avaliku sektori poolt pakutavatel kultuurilistel ja rahalistel aspektidel. Lean on aidanud parandada protsesse ja kvaliteeti. Kvaliteedi parandamisel tuleb tähelepanu pöörata eelkõige töötajate toetamisele ja väljaõppele, mille läbi on võimalik paremini mõista lõppkasutajat ja tema väärtusi. Takistustest on uuringutest [30], [39] välja tulnud, et agiilse metoodika rakendamisel on puudunud arusaam sellest, miks antud metoodikat rakendatakse ning ei osata õigesti kasutada selle tehnikaid. Samuti on takistustena välja toodud välised poliitilised muutused ja see, et kolme tüüpi juhtimise aspektid – strateegilised, operatiivsed ja poliitilised – ei ole olnud omavahel korrigeeritud.

3 Metoodika

Metoodika valikul lähtus autor eelkõige uurimistöö eesmärgist ja uurimisküsimustest. Uurimismetoodika hõlmab vajalike andmete kogumise viisi, valimi koostamist ning kogutud andmete analüüsi ja tõlgendamist [40].

Uurimistöö eesmärgid ja uurimisküsimused toodi välja esimeses peatükis. Käesolevas peatükis kirjeldab töö autor, kuidas ta kavatseb saavutada püstitatud eesmäärke. Alajaotises 3.1 kirjeldab autor töös kasutatud uurimismeetodit, alajaotises 3.2 – töös kasutatud andmekogumismeetodeid ning alajaotises 3.3 – valimi valikut, alajaotises 3.4 andmete töötlemist. Kogutud andmete tõlgendamist kirjeldatakse neljandas (4) peatükis.

3.1 Uurimismeetod

Vastavalt andmete kogumisele ja töötlemisele, jagatakse uurimistöö meetodid üldjuhul kvantitatiivseteks ja kvalitatiivseteks uurimismeetoditeks. Kvantitatiivsed uurimismeetodid näevad ette tõestuse leidmise teooriale või statistilistele arvandmetele. Kvalitatiivsed uurimismeetodid seevastu põhinevad mitteamvulistel andmetel [41], [42].

Käesoleva töö eesmärgiks oli uurida agiilse lähenemise rakendamist Keskkonnaministeeriumi haldusala tarkvaraprotsessi jaoks. Kuna uurimisobjektiks on tarkvaraprotsess, siis töö autori hinnangul sobib töö teostamiseks kvalitatiivne uurimismeetod, sest selle meetodi rakendamise käigus keskendutakse ühele objektile ning uuritakse selle detailset sisu [43].

3.2 Andmekogumismeetodid

Andmeid on võimalik koguda nii teisestest, kui ka esmastest andmetest.

Teisesed võivad olla kas:

1. Sisemised allikad – kirjalikult või muul viisil eelnevalt dokumenteeritud kellegi poolt.
2. Enam kui ühest allikast kombineeritud – võivad olla kas geograafilise piiranguga, või aluseks on näiteks periood, mille jooksul või mille kohta on andmeid kogutud.
3. Välised allikad – raamatutest ning artiklitest saadud informatsioon, samuti igaaastased väljaanded, statistika, raportid, erinevad uuringud (näiteks ajaloo uuringud) [42], [44], [45].

Esmasteks võivad olla:

1. Vaatlusuuring (*Observation*), kus toimub uuringuobjekti vaatlemine tema loomulikus keskkonnas ja tegevuses [44].
2. Eksperiment (*Experiment*), kus katse käigus kontrollitakse püstitatud hüpoteesi [45].
3. Kaardistusuuringud (*Survey*), mida kasutatakse peamiselt kvantitatiivsetes sotsiaalteaduslikes uurimistöodes [45].
4. Juhtumiuuring (*Case study*) – tegemist on meetodiga, kus uurimisobjektiks on kas üksikjuhtum või üksikjuhtumid. See tähendab, et detailsemalt uuritakse üht või mitut indiviidi, programmi, sündmust, tegevust või protsessi. Uuringut ei tohiks mõista kitsamas kontekstis andmekogumismeetodina. See võib toetuda nii kvalitatiivsetele, kui ka kvantitatiivsetele andmetele, kus võib kasutada ühte tüüpi andmeid. Uuring teostaja kogub teatud aja jooksul üksikasjalikku informatsiooni, kasutades erinevaid andmekogumismeetodeid [41], [46].
5. Intervjuud, mis jagunevad kaheks:
 - Struktureeritud, mis tähendab, et intervjuerija loeb ette küsimused ning märgib vastuseid ette lugemisel ja vastuste ära märkimisel, mida on hiljem võimalik arvuti abil töödelda.
 - Struktureerimata intervjuud jälgivad mitteformaalset joont ning tavaliselt kindla teema uurimisele, kus intervjueritav räägib vabalt toimunud sündmustest, enda käitumisest, seisukohtadest. Struktureerimata intervjuusid tasub kasutada siis, kui
 - uuringu eesmärk on koguda põhjalik informatsioon,
 - teema eeldab sügavat isiklikku kontakti või
 - avatud/keeruliste küsimuste puhul [42], [44].

Käesolevas töös kirjeldatud uuringu läbiviimisel kasutas töö autor nii teiseid kui ka esmaseid andmeid. Kirjalikest allikatest kasutati PWC uuringut [8], erinevaid agiilsetest meetodikaid käsitlevaid raamatuid [18], [19], [23], [21] ja agiilse ärianalüüsi koolitusmaterjale [15] ning teadusartikleid [6], [22], [29], [30], [32], [36], [39]. Viimaseid otsis autor peamiselt teadusinfo otsinguvahendist Google Scholar [47] ja EBSCO andmebaasist [48]. Samuti on autor kasutanud statistikat agiilsete arendusmeetodikate kasutamise ja rakendamise kohta, mis on välja toodud erinevates ettevõtte KPMG poolt läbiviidud uuringutes [34], [35].

Selleks, et autor saaks saavutada käesoleva töö eesmärgi, kasutas ta lisaks ülalmainitud teistele andmetele ka KEMIT-i tarkvaraarendusprotsessi kirjeldavat korda [11] ning viis KeM haldusala töötajatega läbi struktureerimata intervjuud. Intervjuu küsimused võimaldasid autoril saada vastused järgmistele uurimisküsimustele:

1. Kuidas agiilne lähenemine võiks aidata kaasa KeM haldusala tarkvaraarendusprojektidele?
2. Kuidas rakendada agiilset lähenemist KEMIT-i tarkvaraarendusprotsessi?

Küsimusi koostas autor tuginedes kirjalikele allikatele [7], [49], [50], [51]. Küsimustik tervikuna on esitatud Lisas 1.

Intervjuude käigus esitati 23 küsimust. Intervjuule vastamine kestis üks tund ja mõne vastaja puhul ka vähem. Iga intervjuu alguses tutvustas autor intervjuueeritavatele uuringu eesmärgi ning selgitas, miks intervjuud tehakse, miks valiti just antud konkreetne intervjuueeritav ja miks vastaja arvamus on oluline. Küsimustik oli kavandatud temaatiliste plokkide kaupa, välja selgitamiseks vastaja igapäevast tööprotsessi ja selle detailid. Intervjuu käigus esitati samuti küsimusi, uurimaks vastaja arvamust organisatsiooni võimalikke muutuste kohta juhul, kui rakendatakse agiilset lähenemist. Autor esitas nii avatud kui ka kinniseid küsimusi stiilis „Kas?“, „Kuidas?“, „Millised?“, „Mis?“, „Kui?“. Mõned küsimused algasid esmalt agiilse lähenemise ja meetodika selgitustega, mis võimaldasid vastajal paremini mõista küsimuste sisu. Osa küsimusi sisaldasid loetelu vastusevalikut. Intervjuude läbiviimise tehnikatest on autor kasutanud vastuste üleskirjutamist ja märkmete tegemist [41].

3.3 Valim

Intervjuude läbiviimiseks kasutas autor eesmärgipärast valimit [41]. See tähendab, et KeM haldusala töötajatest võeti valimisse need, kes igapäevaselt puutuvad või puutusid kokku arendusprojektidega. Intervjuu viidi läbi 11 töötajaga, kus KEMIT-st oli neli (4) IT projektijuhti, üks (1) ärianalüütik, üks (1) süsteemianalüütik, üks (1) vanemarendaja, üks (1) taristu arhitekt, üks (1) rakendusadministraator ning KeM-i haldusalast kaks (2) projektijuhti.

3.4 Andmete töötlemine

Uuringu käigus läbiviidud andmete tõlgendamise mõte oli saada ülevaade uuritavast tervikust, välja lugeda tekkinud mustrid ning korrastada küsimuste kirjastiili. Saadud intervjuude vastuste korduvlugemise käigus puhastas autor vastuste sisu, sest avatud küsimustele sai autor mitmetelt intervjuueeritavatelt vastuseid, mis olid huvitavad uuringu mõttes kuid ei haakunud antud uurimistöo skoobiga. Intervjuude abil kogutud andmed koondas autor kokkuvõteteks ja illustratsiooniks, kasutades vastavaid graafikuid.

4 Tulemused ja analüüs

Käesolevas peatükis esitab autor agiilse lähenemise rakendamise võimalused KEMIT-i tarkvaraarendusprotsessi. Alajaotises 4.1 analüüsib autor intervjuude käigus saadud tulemusi, alajaotises 4.2 teeb järeldusi ning omapoolse ettepaneku KEMIT-is hetkel agiilse lähenemise rakendamise võimaluste kohta KEMIT-i tarkvaraprotsessi.

4.1 Intervjuude tulemused

Uuringust selgusid nii probleemid kui ka riskid, mis võivad tekkida agiilse lähenemise juurutamisel KEMIT-i tarkvaraprotsessi. Samas on autor leidnud mitmeid võimalusi, mis soodustaksid agiilse lähenemise rakendamist.

Uuringust selgus, et nii KEMIT-st, kui ka kliendi juures leidub poolehoidjaid agiilsele lähenemisele. Sellegipoolest ei oleks agiilset metoodikat võimalik täielikult juurutada kõikides KeM haldusala arendusprojektides.

Uuringu autori poolt tõlgendatud üksikasjalikud tulemused on esitatud alljärgnevas teemade kaupa liigendatud alajaotises 4.1.1.

4.1.1 Tulemuste tõlgendamine

4.1.1.1 Suhtlusvahendid, retrospektiivid ning arenduprojekti meeskonnaliikmed

Uuring näitas, et arendusprojektides kasutatakse põhiliste suhtlusvahenditena Skype-i, jooksvate probleemide lahendamiseks ning küsimuste esitamiseks ja neile vastamiseks, elektronposti kokkulepete fikseerimiseks, Jira-t arendustööde ülevaate saamiseks Kanban-i töölaua stiilis ning koosolekuid, mida korraldatakse vastavalt vajadustele. Kliendi ja IT projektijuhid on samuti välja toonud Slack'i ja selle, et kokkuleppe kohaselt oodatakse meeskonnaliikmetelt nimetatud suhtluskeskkonnas kiiret reageerimist lahenduste leidmiseks.

Tulemustest selgus, et arendusprojekti jooksul kasutatavate suhtlusvahenditega püüakse lahendada kõik projekti jooksul tekkinud küsimused ning probleemid ning seetõttu

projekti või selle etapi lõpus ei viida läbi retrospektiive, millest saaksid õppida kõik meeskonnaliikmed. Põhjendusena on välja toodud, et taoliseks koosistumiseks ei ole aega, kuid selleks nähakse vajadust, sest retrospektiiv parandaks kindlasti arenduste kvaliteeti. Samas selgus, et hankepartneri peamine huvi on töö eest tasu saamine ning põhjenduste kohaselt taolised koosolekud ei ole ette nähtud arenduse skooopi. Vastaja arvamusel oleks huvi retrospektiivide korraldamiseks ainult juhul, kui arendajad kuuluksid tellija meeskonda.

Tavaliselt kuuluvad arendusmeeskonda süsteemi- või rakendusadministraator KEMIT-st, kliendi, IT (KEMIT) ja välise partneri projektijuhid, samuti hankepartneri arendajad. Klienti esindavad samuti ärinõunikud ja ärispetsialistid. Kui tegemist on analüüsihankega, siis on meeskonda kaasatud ka välise partneri analüütikud. Mõnest tulemusest oli märgitud kliendi, kui ka hankepartneri poolelt testijad.

4.1.1.2 Eesmärgid, projektist huvitatus, pühendumus, kommunikatsioon ning rollikohased teadmised

Uuringu tulemustest selgus, et üks suurematest probleemidest on inimressurssi puudus. See tähendab, et meeskonnaliikmed ei jõua keskenduda mitmele projektile korraga ning peale arendusprojektide näevad nende ametijuhendid ette ka muude töökohustuste täitmist. Vastavalt sellele näitasid uuringu tulemused, et kui projektri algul ollakse teadlikud arendatava lahenduse eesmärgist, siis projekti jooksul see paraku kas ununeb või kaob teadmine, kuhu lahendusega soovitakse jõuda. Mõned meeskonnaliikmed, nagu näiteks administraatorid, ei jõua projekti lõpuni süveneda, vaid lahendavad projektis ette tulnud ülesandeid. Vahest saadakse eesmärkidest hästi aru, kuid lõpptulemusena selgub, et meeskonnaliikmete huvid on erinevad, kusjuures mõndasid neid neist motiveerivad ainult isiklikud eesmärgid.

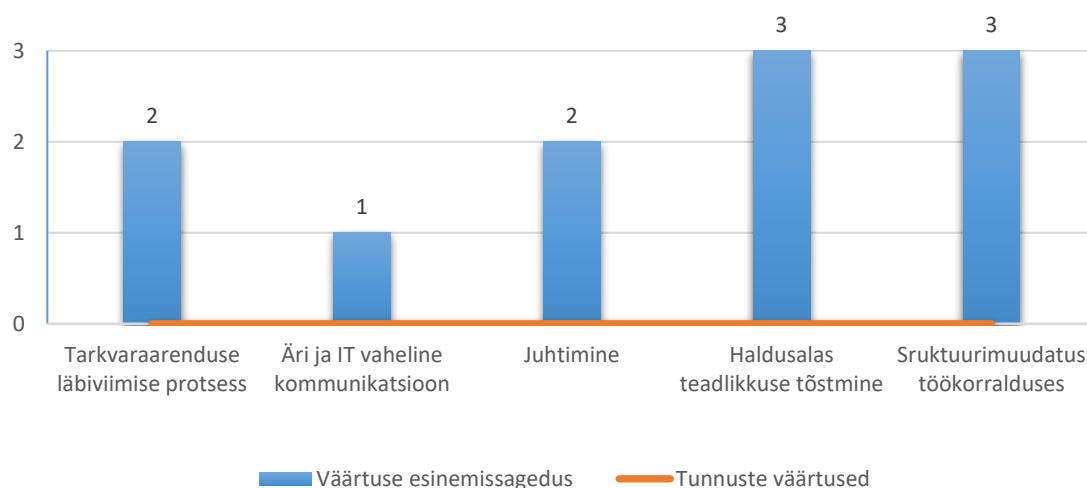
Vähene inimressurss puudutab paraku ka projektist huvitatus, pühendumuse ja teadmiste aspekte. Leidub IT projektijuhte, kellel ei ole sügavaid tehnilisi teadmisi nagu seda on süsteemi- või rakendusadministraatoril. Samuti leidub ka klienti esindavaid spetsialiste, kelle valdkondlikud teadmised ja oskused on vähesed. Need spetsialistid, kellel on vastava valdkonna teadmised, näiteks kliendi ja IT projektijuhid ja administraatorid,

paraku ei jõua ajapuuduse ning ülekoormatuse tõttu projektile lõpuni pühenduda, mis seletab ka nende vähest huvi projekti vastu.

Mis puudutab kommunikatsiooni, siis see on projektiti ja meeskonniti erinev. Tihti tuleb ette takistusi arendusprojekti hankepartnerilt ülevaate saamisel tööprotsessi kulgemise, probleemide ja edenemise kohta. Samuti esineb kommunikatsioonitõrkeid kliendi poolel, kus vajalik info ei jõua õigeaegselt teatud spetsialistideni. Paraku tuleb ette ka seda, et lõpetatud tööetapist ei teavitata, sest liigse töökoormuse tõttu asutakse koheselt täitma järgmiseid ettenähtud ülesandeid.

4.1.1.3 Agiilse lähenemise rakendamine ja juurutamine ning väljaõpe

Uuringus on käsitletud agiilse lähenemise rakendamise ja juurutamise küsimusi. Autor on uurinud, milliseid muudatusi oleks organisatsioonis vaja agiilse lähenemise rakendamiseks ning saadud vastused on kokkuvõtlikult välja toonud Joonisel 2.



Joonis 2. Vajalikud organisatsioonilised muudatused agiilse lähenemise rakendamiseks

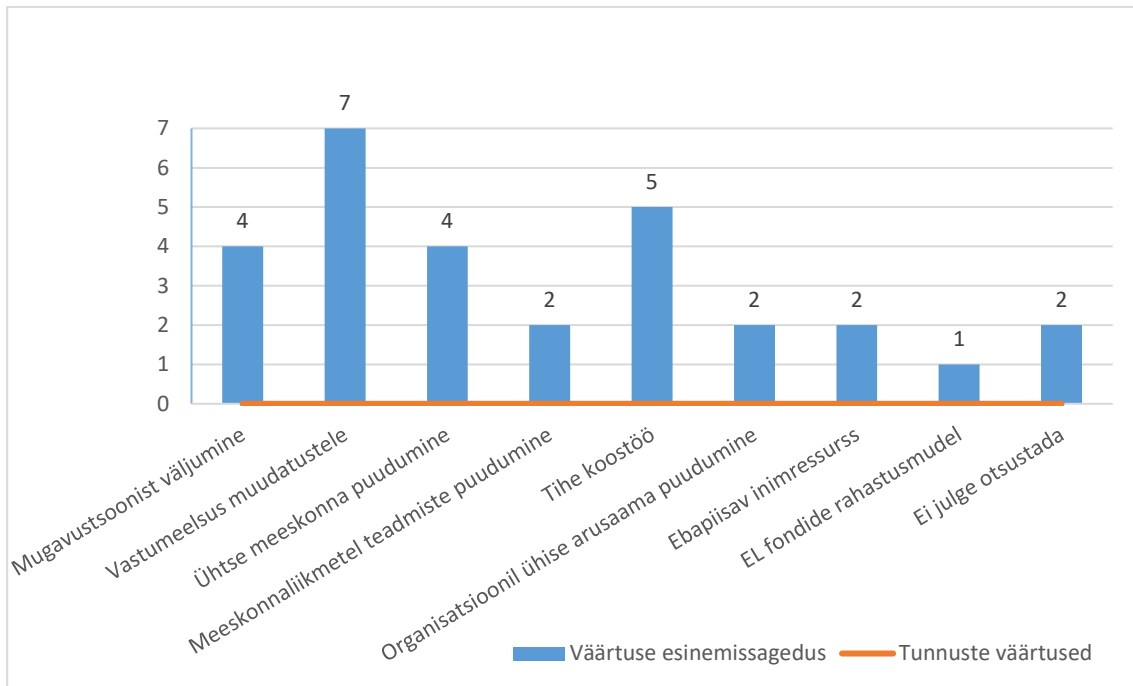
Allikas: autori koostatud vastavalt uuringu tulemustele

Kõige enam arvatakse, et agiilse lähenemise rakendamiseks tuleks muuta töökorraldust ning haldusalas tuleks tõsta teadlikkust. Töökorralduse muutmise all mõeldakse arendusmeeskondade moodustamist ja rollidele täpsete kohustuste määramist. Olemasolevate meeskondade ja töökohustuste juures tuleks paraku kindlasti kaaluda

lisaressursi värbamist ja töö prioritseerimist ajapuuduse ning töökoormatuse tõttu. Samuti oleks tarvis haldusalas tervikuna teadvustada töötajatele nende funktsiooni ja ülesandeid arendusprojektides. Selleks, et tööülesanded saaksid olla täidetud, peab töömetoodika olema kõikidele osapooltele arusaadav ning järgitav nii kliendi, kui ka IT poolt. Kuigi tarkvara arendusprotsess on Keskkonnaministeeriumi haldusalas defineeritud, kõik arendusprojektid ei järgi seda protsessi. Vastajate arvates peaks tarkvara arendusprotsessi uuesti üle vaatama ning sisse viima korrekture ja uuendusi ning arendusmeeskondade kõikidele liikmetele sellest teada andma. Hetkel takistab just töötajate ülekoormatus kliendi ja IT vahelist ladusat koostööd, mistõttu meeskonnaliikmete vaheline kommunikatsioon on nõrk.

Mis puudutab strateegilisi otsuseid, üldisi eesmärke ja tegevuskava, siis nendest teavitamine ja nende arusaadavaks tegemine peaksid kindlasti toimuma organisatsiooni juhtkonna eestvedamisel. Selle tulemusena peaksid töötajad aru saama töö suunast ja prioriteetidest. Ka Nuottila [32] uuringust selgus, et just juht on muudatuste sisseviimise motiveerija ja eestvedaja, omades kindlat visiooni .

Järgmisena uuris autor võimalikke organisatsioonilisi barjääre agiilse lähenemise juurutamisel. Uuringu tulemused selles osas on toodud Joonisel 3.



Joonis 3. Võimalikud organisatsioonilised barjäärid agiilse lähenemise juurutamisel

Allikas: autori koostatud vastavalt läbiviidud intervjuude tulemustele

Intervjueeritavate vastused näitavad, et agiilse lähenemise omaksvõtmine nagu mistahes muudatus võib töötajates tekitada vastumeelsust, sest eeldab oma mugavustsoonist väljumist. Sama tõsiasja on rõhutanud ka Nuottila [32] enda poolt läbiviidud uuringus.

Uuring näitab samuti, et üheks agiilsuse juurutamise barjääriks võib olla vajadus teha tihedamat koostööd, sest agiilsus eeldab tihedat suhtlust meeskonnaliikmete vahel, rohkemat panustamist, arusaamist agiilse meetodika põhimõtetest ja meeskonnaliikmete kokkusobivust. Väljatöötatud ja arusaadav meetodika aitaks samuti kaasa meeskonnaliikmete koostööle.

Probleemina nähakse ka seda, et kliendi ja IT esindajad täidavad oma hõivatuse tõttu ainult oma rollidele vastavaid kohustusi ning paraku puudub ühtne meeskond, mida agiilne lähenemine ette näeb. Ka ainult ühtsest meeskonnast ei piisa, sest töötajaid on tarvis koolitada ja suunata, mille eestvedajaks peaks olema organisatsiooni juhtkond. Uuringus osalejate arvates võib üheks takistuseks agiilsuse juurutamisel olla ka see, et iga muudatus agiilses arendusprojektis eeldab otsuste vastuvõtmist meeskonnaliikmete poolt.

Uuringus osalejad on samuti olnud arvamusel, et barjääriks võib olla asjaolu, et kogu organisatsiooni tasandil ei mõisteta agiilsuse printsiipe, sest tegemist võib olla organisatsiooni jaoks uudse lähenemisega, millest ei saada aru ja mida ei osata kasutada.

Takistusena on välja toodud ka alajaotises 4.1.1.2 käsitletud inimressurss, mille osas ollakse arvamusel, et agiilse lähenemise juurutamisel tuleks kindlasti suurendada meskonda eelkõige arendajate osas. See parandaks oluliselt huvi projektide vastu ning võimaldaks eesmärkide paremat mõistmist. Samuti oleks ühtse meeskonnaga lihtsam saavutada niisugust lahendust, mida klient tõeliselt vajab.

Üks vastajatest on välja toonud ka rahastusmudeli aspekti. Pääaegu kõiki arendusprojekte finantseeritakse erinevatest fondidest (näiteks Euroopa Liidu omast), millede jaoks on raha kasutamine vaja üksikasjalikult ära põhjendada. Põhjendusena aga sobib ainult ühe või teise lubatud funktsionaalsuse väljaarendamine. Niisuguse rahastusmudeli juures on muudatuste sisseviimine lahendusse raskendatud, sest raha taotlemise eelduseks on ära kirjeldatud valmislahenduse teostamine.

Uuringu tulemused näitavad selgelt, et enne agiilse lähenemise rakendamist on kindlasti vaja suurendada töötajate valmisolekut ja parandada nende teadmisi, mis vajavad eelnevat väljaõpet. Sellegipoolest on osad uuringus osalejad leidnud, et väljaõpe on agiilse arendusprotsessi juurutamisel vajalik ainult siis, kui selleks tekib eluline vajadus. Arvatakse, et koolitustest oleks kasu ainult juhul, kui töötajatel on nendega reaalselt mida ette võtta ja tegemist on reaalsete vajadustega. Koolitusesse peaks samuti olema kaasatud organisatsiooni juhtkond, sest just tema peaks olema muudatuste algatajaks ja suunajaks organisatsioonis.

4.1.1.4 Agiilse arendusmetoodika juurutamise riskid, arenduse kvaliteet ja soovitud lahendused

Iga muudatuse ja uuenduse juurutamisega võivad kaasneda teatud riskid. Selle teadmise juures pidas töö autor samuti vajalikuks selgitada uuringus osalejate arvamust selle kohta, kas agiilse lähenemise juurutamisega riskid suureneksid või väheneksid, võrreldes Kose mudeliga. Leiti, et kuna agiilsus eeldab järkjärgulist arendust, siis taoline lähenemine võimaldab lahenduse kohest proovimist, millega väheneb mittevajaliku lahenduse väljatöötamise risk. Samas suureneks liigse ajaressursi risk, sest ootustele vastava

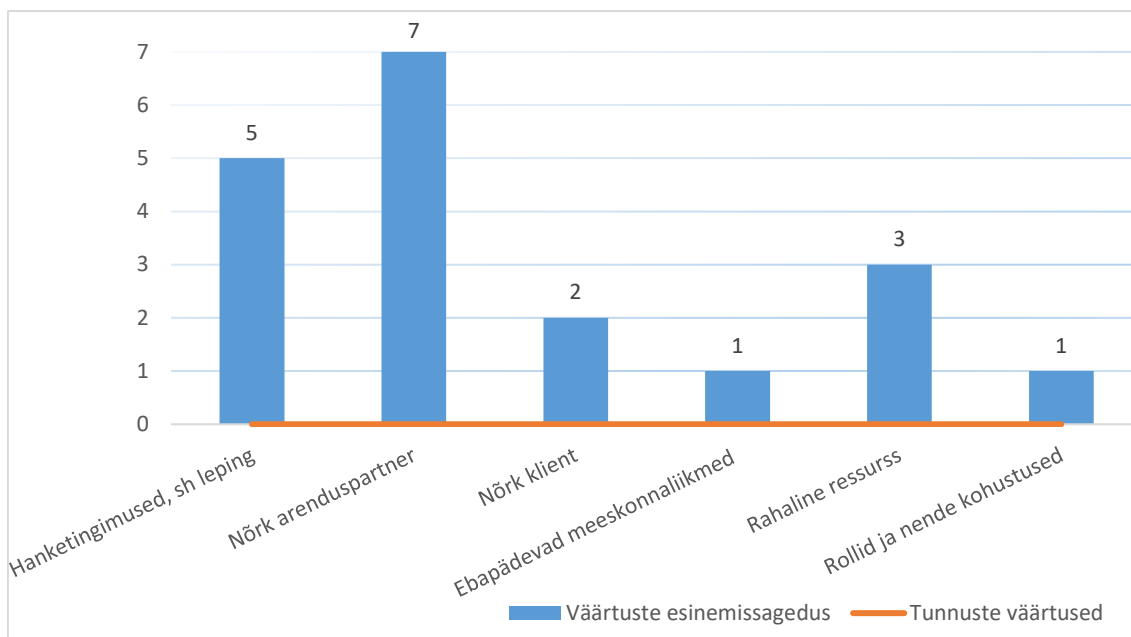
tulemuse väljatöötamine nõuab rohkem panustamist. Kindlasti väheneks kommunikatsioonirisk, aga ainult juhul, kui töökeskkond vastab agiilse lähenemise põhimõtetele. Samuti, kui rakendada agiilset lähenemist olemasolevas organisatsioonis, siis vastajate arvates ei saa see toimida enne, kui organisatsiooni struktuuri on sisse viidud vastavad muudatused. Selleks aga peaks tõstma töötajate teadlikkust ja väljaõpet, värbama juurde inimesi ja prioritseerima tööd. Kindlasti suureneks eelarve risk, kuid jätkuarenduste vaates oleksid need kulutused õigustatud.

Agiilse lähenemise juurutamisel tuleb kindlasti ette riske, mida peab teadvustada ja tegelema, kuid enamik uuringus osalejaid olid siiski arvanud, et agiilsuse juurutamine aitab nii tõsta arenduse kvaliteeti, kui ka arendada lahendusi, mis vastavad äri vajadustele paremini, kui Kose mudeli kasutamisel.

Enamiku uuringus osalejate arvates peaks agiilse lähenemise juurutamist avalikus sektoris soodustama, aga seda ainult nende arendusprojektide juures, kus see oleks mõistlik. Uuringus osalejad on ka märkinud, et soodsate tingimuste juures, nagu seda oleksid näiteks vastavad riigieelarveline ressurss ja organisatsioonilised ümberkorraldused, võiks agiilset lähenemist kasutada pea iga arendusprojekti juures.

4.1.1.5 Arendusprojektide läbiviimist takistavad tegurid, esialgsete nõuete muutmise vajadus, riigihangete seadus, EL rahastusnõuded ja kasutatavad arnedusmeetodid

Joonisel 4 on välja toodud uuringu tulemused arendusprojektide läbiviimist takistavate tegurite osas.



Joonis 4. Arendusprojektide läbiviimist takistavad tegurid

Allikas: autori koostatud vastavalt läbi viidud intervjuude tulemustele

Uuringust selgus, et kõige rohkem arendusprojektide läbiviimist takistavaid tegureid on seotud arenduspartneriga. Uuringus osalejad väitsid, et hankega saadud partner ei ole alati motiveeritud täisulatuses projekti panustama. Selle peamise põhjusena on välja toodud projekti fikseeritud rahaline summa. Seetõttu muutuste sisseviimine ei kuulu projekti skoopi ega vasta rahastusmudelile. Tõepoolest, näiteks Euroopa Liidu rahastusmudeli puhul peab kirjeldama täitmiseks ettemääratud nõuded. Nendeks on kas ärianalüüs või tegevuskava, mis toovad välja funktsionaalsused, mida arenduse käigus soovitakse realiseerida. Siit tulenevad ka hankelepingus esitatud tingimused, mis ei soodusta muudatuste sisseviimist, sest põhinevad fikseeritud nõuetel.

Uuringus osalejad on takistava tegurina välja toonud ka nõrgad teadmised. Ebakompetentsust esineb nii arenduspartneril kui ka kliendil. Ette on tulnud, et klient ei hoi oma lubadustest kinni just ajapuuduse tõttu. Samuti on tihti esinenud, et töö tegemiseks ei leidu piisavalt inimressurssi. Tööjaotuski ei ole alati kõikidele selge ning puudulik on arusaam rollide kohustustest.

Eeltoodud takistustele vaatamata ollakse arvamusel, et agiilist lähenemist avalikus sektoris peaks siiski soodustama. Põhjuseks tuuakse välja, et klient saaks jooksvalt sisse viima muudatusi, saades kiiremat tagasisidet, kui see on hetkel võimalik ning saaks

maandada osasid riske, nagu nt kommunikatsioon ja rahalised ressursid ning kõrvaldada üleliigsust.

Uuringus osalejate arvates ei takista agiilset arendamist ka riigihangete seadus. Lisaks selgus, et kuigi KeM haldusala on ette määratud tarkvaraprotsessi kord [11], viiakse ka praegu antud haldusala projekte tegelikult läbi kasutades nii Kanbani praktikaid kombinatsioonis Kose mudeliga, kui ka Scrumi metoodikat koos Kose mudeliga.

4.2 Agiilse lähenemise rakendusvõimaluste ettepanekud KEMIT-i tarkvaraprotsessi

Käesolevas alajaotises teeb autor ettepanekud agiilse lähenemise rakendamiseks KEMIT-i tarkvaraprotsessi. Ettepanekute tegemise aluseks võetakse teooriad, mis välja toodud teises (2) peatükis ning autori poolt läbiviidud uuringu tulemused, mida on käsitletud alajaotises 4.1.1 ning KEMIT-i tarkvaraprotsessi kord [11].

Ettepanekud on üles ehitatud lähtudes käesoleva töö sissejuhatavas peatükis 1 väljatoodud eesmärgist – uurida agiilse lähenemise rakendamisvõimalusi KEMIT-i haldusala tarkvaraprotsessis.

4.2.1 Järeldused ning ettepanekud

Töö sissejuhatavas peatükis 1 puudutati nii era- kui ka avaliku sektori olemust ja seost. Erasektorile omased tunnusjooned on konkurents ning kliendile pakutava toote või teenuse pealt kasumi teenimine, järgides turumajanduslikke reegleid. Konkurents eeldab muudatustega kohanemist ja innovatsiooniga kaasaskäimist organisatsioonikultuuri [1]. [2]. Teenuse või toote arendamine ja kohandamine erasektoris toimub vastavalt kliendi tagasisidele ja kliendi vajadustele. Avalik sektor erineb erasektorist selles osas, et eesmärgiks ei ole kasumi vaid avaliku huvi teenimine, järgides nii poliitilisi otsuseid kui ka erinevatel tasemetel seadusandlust.

Üheks muudatustrendiks arendusmaastikul on agiilne lähenemine [5]. See sai ka läbivaks teemakäsitletuseks käesolevas töös. Ei saa väita, et avalik sektor ei saaks üldse kaasas käia muudatustega ning innovatsiooniga. See on võimalik, kuid paraku võtab rohkem aega kui erasektori puhul [32] l.

Järgnevalt toob autor oma peamised ettepanekud agiilse lähenemise rakendamiseks KEMIT-i tarkvaraprotsessis:

1. Juht on eestvedaja.

Margretta ja Stone [3] ning Nuottila et.al. [32] väidavad, et muudatuste elluviimine organisatsioonis peaks algama juhtkonnast. Osad uuringus osalejatest on arvamusel, et selleks, et organisatsiooni uuendusi sisse tuua, nagu seda on näiteks agiilse lähenemise juurutamine, peaks muudatustest eelkõige aru saama ning need omaks võtma organisatsiooni juhtkond. Seejärel oleks autori arvates mõistlik jätkata terve organisatsiooni väljaõppega. Agiilne treener ehk *agile coach* oleks see roll, mille täitjad võiksid organisatsioonil ja selle juhtkonnal aidata mõista agiilsuse põhimõtteid ja agiilse arendamise toimimist organisatsioonikultuuri mõttes. Analoogiliselt oli ka Assist Knowledge Development'i koolituse koolitaja [15], millest töö autor 2018. aastal osa võttis, arvamusel, et agiilne lähenemine peaks alguse saama agiilsest treenerist.

2. Meeskonnad ja koostöö on tähtsad. Kommunikatsiooni peab parandama.

Magistritöö alajaotis 2.2.1. käsitleb agiilse lähenemise põhimõtteid. Üheks tähtsamatest nendest on inimesed ja inimestevaheline koostöö. Inimeste all mõeldakse kõiki arendusprojekti meeskonnaliikmeid nii IT-teenuse pakkuja kui ka kliendi poolel. Autori poolt läbiviidu uuringu tulemustest selgus, et kõikidele arendusprojekti meeskonnaliikmetele ei ole alati selge nende roll ja kohustused ning kommunikatsioon ei ole piisav. Taoline olukord tuleneb töötajate liiga suurest hõivatusest. Töötajad täidavad oma töökohustusi, et tööd õigeks ajaks valmis saada, tihti mõtlemata eesmärkidele ja lõpplahendusele. Agiilne lähenemine aga eeldab tihedat ja paindlikku suhtlust ning mitte kinniolemist meeskonnaliikmete rollides, vaid pigem keskendumist kliendile pakutavale väärtusele. Ilmselgelt parandaks olukorda töökorralduse muutmine meeskondade loomise näol, kuhu kuuluksid need rollid (näiteks toote omanik, analüütik, arendaja ja testija), mida agiilsed meetodikad tüüpiliselt ette näevad. Selleks tuleb olemasolevate töötajate teadmisi tõsta ja juurde värbata kompetentseid spetsialiste. Samuti võiks projekte ning vastavaid töid prioritseerida, et saaksid suurenda pühendumus, motiveeritus, iseorganiseeruvus, eesmärkidest arusaamine ja kliendi lahendusele keskendumine.

3. Reeglite järgimine.

Agiilse lähenemise järgimine on võimalik vaid juhul kui kõik meeskonnaliikmed järgivad väljavalitud meetodika juhiseid, st oskavad kasutada tehnikaid ning tegutsevad vastavalt

ettenähtud korrale ja kokkulepetele. Selle eeldusteks on töötajate harimine ning ühine arusaam ja usaldus ning vajalik suunamine ja toetus ja vastava keskkonna loomine juhtkonna poolt.

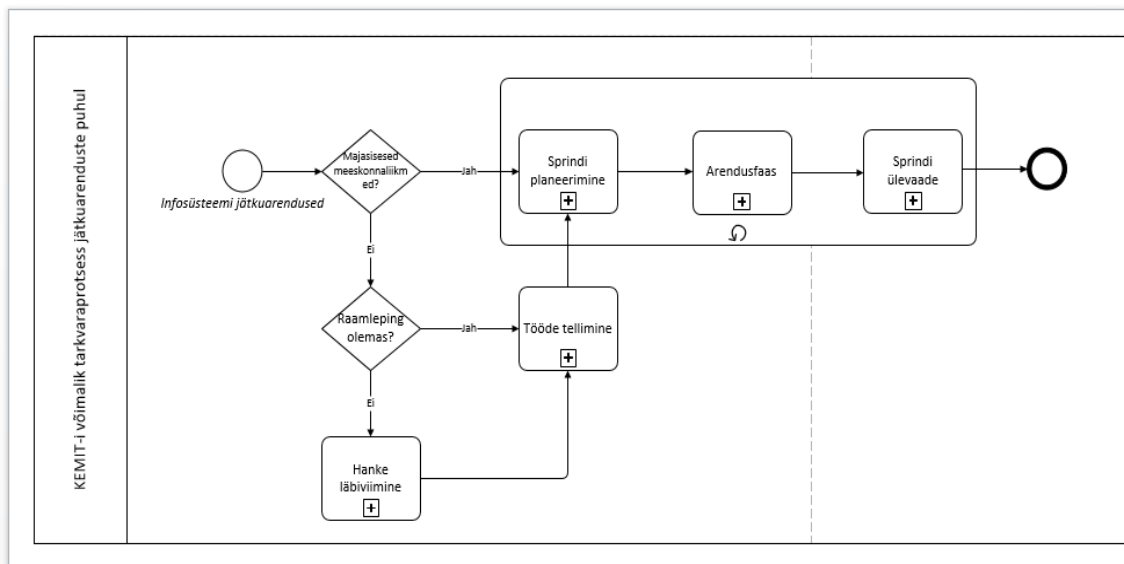
4. Muutuvad ärivajadused – Scrum, kui üks võimalustest jätkuarenduste puhul.

KeM haldusalas tuleb tihti sisse viia muudatusi ja täiendusi juba olemasolevasse infosüsteemi, vastavalt seadusandlusest tulenevatele nõuetele. Vastavalt KEMIT-i tarkvaraprotsessile [11], muudatuste elluviimine KeM haldusalas näeb ette viie (5) etapi läbimist arendusvajaduse esitamisest kuni selle teostamiseni. Kõige aeganõudvamad nendest etappidest on autori arvates ärivajaduste kogumine, Euroopa Liidu fondidelt raha taotlemine ning vajadusel arenduspartneri hankimine juhul, kui eelnevalt ei ole sõlmitud raamhankeleping, mis võimaldab arendustööde kohest tellimist. Tuginedes alajaotistele 2.2.3.1. ning 2.2.3.4. on autori arvates üheks sobivaimaks agiilse arenduse meetodikaks KeM haldusala jaoks Scrum, mida võiks kasutada koos *Lean* lähenemisega. Viimane aitaks eemaldada üleliigseid tegevusi. Scrum võimaldaks KEMIT-i tarkvaraprotsessi viie (5) etapi läbimist operatiivsemalt, pakkudes kliendile kiiremat lahendust seadusandlike muudatuste sisseviimise korral.

Scrum-metoodika toimiks KeM haldusalas eeldusel, et on olemas oma arendusmeeskond, kuhu muuhulgas kuuluksid KEMIT-i arendajad. Vastav lähenemine sobiks ka väljast hangitud arendajatega. Meeskonnaliikmetel võiks olla võimalik vastu võtta otsuseid ja koheselt tegutseda. Selleks peaks enamikul juhtudel muutma ka rahastusskeeme. Arendusprojekti peaks olema kaasatud kogu arendusmeeskond, alustades projekti eesmärgi paikapanemisega ning jätkates tööde prioritseerimisega, järk-järgulise arendamise ja testimisega ning vajadusel muudatuste kohese sisseviimisega. Selle juures on üheks võtmeteguriks meeskonnaliikmete omavaheline suhtlus.

Uuringust selgus, et mitte kõikide projektide puhul ei ole mõistlik ega otstarbekas kasutada agiilset meetodikat. Näiteks uute süsteemide arendamine vajab Euroopa Liidu fondide rahastust, mis aga eeldab teatud tingimuste täitmist. See aga on takistuseks operatiivsete otsuste vastuvõtmisel ning muudatuste sisseviimisel. Samas kasutatakse agiilseid tehnikaid nagu näiteks kasutuslood (*User stories*), Kanbani töölaud ja ärinõute prioritseerimine, ka juba praegu kehtivas KEMIT-i tarkvaraprotsessis [11]. Kindlasti võks lisaks eelmainitud tehnikatele samuti kasutada kiirkoosolekuid ja langustrendi graafikuid ning retrospektiive, kuid see peaks olema kõikide arendusmeeskondade liikmete poolt aktsepteeritud. Juba väljaarendatud infosüsteemi puhul oleks igati võimalik

jätkata lisaarendustega rakendades näiteks Scrum-metoodikat. Joonisel 5 on autori poolt välja toodud võimalik KEMIT-i tarkvaraprotsess jätkuarenduste tegemiseks, järgides Scrum-metoodikat.



Joonis 5. KEMIT-i võimalik tarkvaraprotsess jätkuarenduste puhul

Allikas: Autori koostatud

5 Kokkuvõte

Magistritöö eesmärk oli uurida agiilse lähenemise rakendusvõimalusi Keskkonnaministeeriumi haldusala tarkvaraprotsessi jaoks. Tulemuse saavutamiseks on autor läbi töötanud teaduslikku kirjandust ning läbi viinud kvalitatiivse uuringu, saades vastuseid läbi viidud intervjuudest, kasutades struktureerimata küsimustikku. Vastused leiti kõigile püstitatud uurimisküsimustele:

1. Milliseid agiilse arendusprotsessi praktikaid kasutatakse avalikus sektoris?

Varasemalt läbi viidud KPMG uuring on näidanud, et Hollandi ning Belgia avaliku sektori organisatsioonides on peamiselt kasutuses Scrum meetodika ning Scrum ja Kose mudeli kooskasutus. KPMG ülemaailmisest küsitlusest selgub, et 40% kasutatakse Scrum meetodikat ning 11% LSD meetodikat. Enim kasutatavamad, Medvedska ning Berzisa väite kohaselt, on Kose mudel, Scrum meetodika ning DSDM lähenemine.

2. Milliseid agiilse arendusprotsessi praktikad kasutatakse Eesti avalikus sektoris?

PWC uuringu kohaselt agiilsete meetodikate eesrindlik kasutaja on SMIT, kes on võtnud kasutusse *Lean* metodoloogia lähenemist, Scrum meetodikat ning Kanban praktikat.

3. Kuidas agiilne lähenemine võiks aidata kaasa KeM haldusala tarkvaraarendusprojektidele?

Autori poolt läbi viidud uuringust selgus, et peamine probleem on IT ja kliendi vaheline suhtlus. Agiilne lähenemine kindlasti aitab kommunikatsiooni parandada, aga ainult juhul, kui keskkond vastab agiilse arendamise eeldustele. Vastajad on samuti arvanud, et agiilsuse juurutamine aitakst tõsta nii arenduse kvaliteeti, kui ka arendada lahendust, mis vastaks paremini defineeritud ärivajadustele, kui seda Kose mudeli juures. Samuti agiilsuse järg-järguline arendamine võimaldaks lahenduse kohest proovimist ning vastavate korrektuuride sisse viimist.

4. Kuidas rakendada agiilset lähenemist KEMIT-i tarkvaraarendusprotsessi?

Agiilne mõtteviis peaks tulema organisatsiooni läbi juhtkonna. Agiilsetest printsiipidest võiks aidata aru saada agiilne treener ehk *agile coach*. Agiilsete juhiste järgimiseks peab eelkõige olema komplekteeritud kompetentsed ning iseorganiseeruvad meeskonnad. Kui keskkond loodud ning inimesed ette valmistatud, oleks võimalik agiilset lähenemist tarkvaraarendusprotsessi rakendada jätkuarenduste juures.

Autor leiab, et magistritöö eesmärk saavutati, kuna töö käigus valmis soovitud tulem – võimalik ettepanek agiilse lähenemise rakendamiseks Keskkonnaministeeriumi (KeM) haldusala tarkvaraprotsessi jaoks.

Autori arvates saab antud tööd aluseks võtta Scrum metoodika juurutamiseks KeM haldusala tarkvaraprotsessi. Magistritöö tulemused aitaksid kindlasti kaasa nii tudengitele, kui ka organisatsioonidele.

Kasutatud kirjandus

- [1] J. Edler ja J. Fagerberg, Innovation policy: what, why, and how, *Oxford Review of Economic Policy*, Volume 33, Number 1, 2-23, 2017.
- [2] R. Alas ja M. Pramann Salu, Muudatuste meistrikläss, Äripäeva Kirjastus, 2005.
- [3] J. Margretta ja N. Stone, Mis on juhtimine? Kuidas see töötab ja miks on see igapäevane asi, Pegasus, 2003.
- [4] Ametniku soovitusõnastik, Eesti Keele Instituut, 2013.
- [5] D. Orusalu, Agiilsus kui eduka ettevõtte märksõna, SEB Foorum, 2015. [WWW] <https://www.seb.ee/foorum/ettevotlus/agiilsus-kui-eduka-ettevotte-marksona> [22.09.2018].
- [6] P. Serrador ja J. K. Pinto, Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051, 2015.
- [7] J. Lazartšuk, Agiilsete arendusmetoodikate kasutamise probleemid Eesti avalikus sektoris: bakalaureusetöö. Tallinn: Tallinna Tehnikaülikool, Tallinn, 2017.
- [8] M. Lauk, Lõpparuanne. Siseministeeriumi haldusala IKT teenuste arendamine ja haldamise finantseerimise jätkusuutlikkus ning mõju siseturvalisuse tagamisele, PWC, 2018. [WWW] https://www.smit.ee/pdf/SiM%20IKT%20finantseerimine_PwC_L%C3%B5pparuande%20avalik%20versioon.pdf [02.10.2018].
- [9] G. Jonnalagadda, S. Shafey, W. Lynah, M. Massetti, P. Maerten, S. Wiczorek, S. Landzaat, M. Probst, A. Schuster, C. Oxborough, R. Bains, M. Bonser, Agile Project Delivery Confidence. Mitigate project risks and deliver value to your business, PWC, 2017. [WWW] <https://www.pwc.com/gx/en/actuarial-insurance-services/assets/agile-project-delivery-confidence.pdf> [04.02.2018].
- [10] Keskkonnaministeeriumi Infotehnoloogiakeskuse põhimäärus. (Vastu võetud, muudetud, täiendatud, viimati jõustunud 01.01.2013). – Elektrooniline Riigi Teataja, [WWW] <https://www.riigiteataja.ee/akt/109032018023> [31.12.2018].
- [11] K. Viik, KEMIT tarkvara arendusprotsess, Versioon 1, Keskkonnaministeeriumi Infotehnoloogiakeskus, 2018.
- [12] M. Markvardt, Kvaliteedijuhtimine ja tarkvaraprotsesside parendamine, Konspekt, IT Kolledž, 2011. [WWW] http://enos.itcollege.ee/~mmarkvar/I365/8_ProtsessideParendamine.pdf [01.01.2019].
- [13] I. Petuhhov, Tarkvara elutsükli mudelid, Konspekt, Tallinna Ülikool, 2015. [WWW] http://www.cs.tlu.ee/~inga/SE_materjal/Protsessimudelid_slaidid_2015.pdf [11.11.2018].

- [14] M. Fowler, UMLi kontsentraat. Objektmodelleerimise standardkeele UML 2.0 lühijuhend, 3. redaktsioon, Cybernetica AS, 2006.
- [15] A. Privitera, Agile Business Analysis Workshop, Koolitus, Assist Knowledge Development LTD, London, 2018
- [16] Andmekaitse ja infoturbe leksikon, Cybernetica, 2011-2019.
- [17] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, Agile software development methods. Review and analysis, VTT Publications 478, Otamesia Oy, Espoo, 2002.
- [18] J. Shore ja S. Warden, The Art of Agile Development, 1. ed, O'Reilly Media Inc, 2007.
- [19] E. Derby ja D. Larsen, Agile retrospectives. Making good teams great. The Pragmatic Programmes, 2006.
- [20] R. Öpik, Kontseptuaalne süsteemianalüüs, Konspekt, Tallinna Tehnikaülikool, 2009.
- [21] K. Schwaber, Agile Project Management with Scrum, Microsoft Press, 2004.
- [22] P. Huberts, Software Development for the Public Sector: Master Thesis, Radboud University, 2017.
- [23] K. Beck ja C. Andres, Extreme Programming Explained: Embrace Change, 2. ed, Addison-Wesley, Boston, Pearson Education, Inc, 2003.
- [24] V. Valdmees, Ekstreemprogrammeerimine: ülevaade, praktikad ja võrdlus teiste väledate arendusmeetoditega: bakalaureusetöö. Tallinn: Tallinna Ülikool, Tallinn, 2006.
- [25] V. Otsason, Ülevaade väledatest tarkvaraarenduse meetodikatest: bakalaureusetöö. Tartu: Tartu Ülikool, Tartu, 2017.
- [26] DSDM Handbook, Agile Business Consortium Limited, 2018. [25.11.2018],
- [27] M. Bankiir, Mis on Lean ja kuidas see ettevõttele kasulik on?, Leanway, 2013. [WWW] <https://leanway.ee/blogi/mis-on-lean-ja-kuidas-see-ettevottele-kasulik-on/> [30.11.2018].
- [28] P. Leis, Infosüsteemi strateegiline juhtimine ning arendamine, Konspekt, Tallinna Tehnikaülikool, 2017.
- [29] N. Maarse ja M. Janssen, The Need to Adjust Lean to the Public Sector, Hal, 2017.
- [30] E. Drotz, Lean in the Public Sector. Possibilities and Limitations: Thesis No. 1677, Sweden: Linköping University, Linköping 2014.
- [31] T. Sepp, Ettevõtte tarkvaraarenduse üleviimine koskmeetodilt agiilsele Elio Ettevõtte AS näitel: magisträtöö. Tallinn: Tallinna Tehnikaülikool, Tallinn, 2010.
- [32] J. Nuottila, K.Aaltonen, J.Kujala, Challenges of adopting agile methods in a public organization, *International Journal of Information Systems and Project Management*, Vol. 4, No.3, 65-85, 2016
- [33] V. Valkiainen, Eva-Maria Kangro: agiilne juhtimine toob eksperimenteerimise esikohale, Juhtimisblogi, Valkiainen Konsultatsioonid, 2018. [WWW] <http://valkiainen.ee/blogi/eva-maria-kangro-agiilne-juhtimine-toob-eksperimenteerimise-esikohale> [01.12.2018].
- [34] J. Groosmann, P. Olieman, S. Meijs, J. Brummelen, Agile project delivery. How to increase project success in a hybrid world?, Survey on project and programme management, KPMG, 2017.

- [35] G. Barlow, A. Tubb, G. Riley, Driving business performance, Project Management Survey, KPMG, 2017.
- [36] O. Medvedska ja S. Berzisa, Selection of Software Development Project Lifecycle Model in Government Institution, doi: 10.1515/itms-2015-0001, De Gruyter, 2015/18
- [37] M. Laussoo, Lean ei saa mitte kunagi otsa, Siseministeriumi infotehnoloogia- ja arenduskeskuse ajakiri, 3. väljaanne, 2016. [WWW] https://issuu.com/smit2008/docs/smit_talv_2016_net [01.12.2018].
- [38] S. Fjeldstad, M. Lundqvist, P. Olesen, From waterfall to agile: How a public agency launched new digital services, McKinsey's , Oslo, 2016.[WWW] <https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/from-waterfall-to-agile-how-a-public-agency-launched-new-digital-services> [20.12.2018].
- [39] M. Barraza, T. Smith, S. Dahlgard-Park, Lean-kaizen public service: an empirical approach in Spanish local government, *The TQM Journal*, Vol 21, No. 2, Emerald, 2009.
- [40] K. Niglas, The Combined use of Qualitative and Quantitative methods in Educational research: doctor dissertation, TPÜ Kirjastus, Tallinn, 2004.
- [41] R. Yin, Case study research design and methods, 5th ed, SAGE, 2014.
- [42] P. Ghauri ja K. Gronhaug, Research Methods in Business Studies. A Practical Guide, 3rd. ed., Prentice Hall, Financial Times, 2005.
- [43] M. Laherand, Kvalitatiivne uurimisviis. Tallinn, 2008.
- [44] M. Saunders, P. Lewis, A. Thornhill, Research Methods for Business Students. 4th ed. England: Pearson Education Limited., 2007.
- [45] L. Õunapuu, Kvalitatiivne ja kvantitatiivne uurimisviis sotsiaalteadustes, Tartu Ülikool, 2014.
- [46] K. Creswell, Research design: Qualitative, Quantitative and Mixed Methods Approaches, 3rd. ed., SAGE Publications, Inc, 2009.
- [47] Google Scholar, Bibliographic database, 2004.
- [48] EBSCO host, Database, EBSCO Industries, Inc, 2018.
- [49] A. Ribeiro ja L. Domingues, Acceptance of an agile methodology in the public sector, Elsevier, Procedia Computer Science 138, 621–629, 2018.
- [50] H. Gnadenteich, Waterfallist agiilseks: magistritöö, Tallinn: Tallinna Ülikool, Tallinn, 2017.
- [51] A. Tammik, The possibility of agile development in the public sector: the case of Estonia: bachelor thesis, Tallinn. Tallinn University of Technology, Tallinn, 2016.

Lisa 1 – Intervjuu küsimused

1. Milliseid suhtlusvahendeid (e-mail, Jira, koosolek, muu) kasutate/kasutasite arendusprojektis(-des), millest osa võtate/võtsite?
2. Kes on/olid Teie meeskonnaliikmed arendusprojektis?
3. Kas arendusprojekti etapi lõpus või projekti lõpus teete/olete teinud retrospektiivi?
4. Kas kogu projektimeskonnal on/oli arendatava lahenduse eesmärk selge?
5. Kas olete märganud, et kliendi projektijuht/IT projektijuht ei ole projekti läbiviimises huvitatud?
6. Kas olete tundnud, et projektimeskonna liikmel(-tel) puuduvad tema rolliga seotud teadmised?
7. Kas olete tundnud, et projektimeskonna liikmel(-tel) puudub valmisolek (tahtmine) pühenduda projektile?
8. Kas projekti käigus juhtub, et projektimeskonna liikmete vahel puudub kommunikatsioon? Kui jah, siis kelle vahel sagedamini?
9. **Agiilselt läheneda tähendab olla paindlik ja kiire. Agiilse lähenemise protsessid on rohkem avatud ja vabamad, kui traditsioonilise lähenemise protsessid. Agiilne lähenemine lubab proovida uusi lahendusi, teha vigu ja neist õppida.**
Milliseid muudatusi oleks organisatsioonis vaja agiilse lähenemise rakendamiseks?
10. **Agiilselt läheneda tähendab olla paindlik ja kiire. Agiilse lähenemise protsessid on rohkem avatud ja vabamad, kui traditsioonilise lähenemise protsessid. Agiilne lähenemine lubab proovida uusi lahendusi, teha vigu ja neist õppida.**
Millised võivad teie arvates olla organisatsiooni barjäärid (vastuseis muudatustele, kliendi ja IT vahelise kommunikatsiooni puudumine, ebapiisav ressurss, ei mõisteta agiilse metoodika olemust, ei osata kasutada agiilse metoodika vahendeid, meeskonnatöö, muu) agiilse lähenemise juurutamisel?
11. **Agiilselt läheneda tähendab olla paindlik ja kiire. Agiilse lähenemise protsessid on rohkem avatud ja vabamad, kui traditsioonilise lähenemise protsessid. Agiilne lähenemine lubab proovida uusi lahendusi, teha vigu ja neist õppida.**
Agile coach ehk agiilne treener on isik, kes aitab organisatsioonil mõista agiilse lähenemise põhimõtteid ja neid järgida. Ta aitab meeskonnaliikmetel

ka ise optimeerida meeskonna tööprotsesse ning keskenduda lahenduse eesmärkidele. Agiilne treener aitab ka kõrvaldada takistusi ja selgitab lahenduse detaile.

Kas arvate, et agiilse arendusprotsessi juurutamisel peaks kaasama agiilse treeneri abi?

12. Agiilselt läheneda tähendab olla paindlik ja kiire. Agiilse lähenemise protsessid on rohkem avatud ja vabamad, kui traditsioonilise lähenemise protsessid. Agiilne lähenemine lubab proovida uusi lahendusi, teha vigu ja neist õppida.

Kas agiilse lähenemise juurutamise riskid (kommunikatsiooni, tulemuse, skoobi, eesmärkide või muu osas) suureneksid või väheneksid, võrreldes kose mudeliga?

14. Kas Teie arvates agiilse arendusmetoodika juurutamine aitab kaasa arenduse kvaliteedi tõstmisele?

15. Kas Teie arvates agiilse arendusmetoodika juurutamine aitab saavutada just kliendi vajadustele vastavat lahendust?

16. Kas arendusprojekti käigus on ette tulnud vajadust muuta esialgseid nõudeid?

17. Mis on arendusprojekti läbiviimisel takistavateks teguriteks (nõrk arenduspartner, hanketingimused, muu)?

18. Agiilselt läheneda tähendab olla paindlik ja kiire. Agiilse lähenemise protsessid on rohkem avatud ja vabamad, kui traditsioonilise lähenemise protsessid. Agiilne lähenemine lubab proovida uusi lahendusi, teha vigu ja neist õppida.

Kas agiilset lähenemist saab kasutada kõikides arendusprojektides?

19. Agiilselt läheneda tähendab olla paindlik ja kiire. Agiilse lähenemise protsessid on rohkem avatud ja vabamad, kui traditsioonilise lähenemise protsessid. Agiilne lähenemine lubab proovida uusi lahendusi, teha vigu ja neist õppida.

Kas agiilsete metoodikate kasutamist avaliku sektori projektides peaks soodustama? Miks?

20. Kuidas riigihangete seadus hetkel soodustab või takistab agiilset lähenemist?

21. Milliseid nõudeid (põhjalik analüüs, tegevuskava olemasolu ja selle järgimine, muu) seavad EL fondid arendusprojektide finantseerimisele?

22. Milliseid arendusmetoodikaid kasutavad hankega saadud arenduspartnerid?

23. Kas KeM haldusalas järgitakse KEMIT-i tarkvaarendusprotsessi?