

TARTU ÜLIKOOL

Pärnu kolledž

Ettevõtlusosakond

Sten Viljus

**AGIILSETE PRINTSIIPIDE RAKENDAMINE  
KESKKONNAMINISTEERIUMI HALDUSALA  
PROJEKTIDE JUHTIMISEL**

Lõputöö

Juhendaja: Arvi Kuura, PhD

Pärnu 2021

Soovitan suunata kaitsmisele

(allkirjastatud digitaalselt)

Arvi Kuura

Kaitsmisele lubatud

TÜ Pärnu kolledži programmijuht

(allkirjastatud digitaalselt)

Margus Kõomägi

Olen koostanud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, põhimõttelised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

(allkirjastatud digitaalselt)

Sten Viljus

## SISUKORD

|   |    |
|---|----|
| Sissejuhatus .....  | 4  |
| 1. Tarkvaraarenduse juhtimine .....   | 7  |
| 1.1. Projektid ja tarkvaraprojektid .....                                       | 7  |
| 1.2. Levinuimad tarkvaraarenduse juhtimise meetodikad .....                     | 9  |
| 1.3. Agiilsete arendusmetoodikate ülevaade .....                                | 14 |
| 1.4. Eeldused agiilsete printsiipide rakendamiseks .....                        | 19 |
| 2. Agiilse tarkvaraarenduse rakendamine Keskkonnaministeeriumi haldusalas ..... | 23 |
| 2.1. Uurimisobjekti tutvustus .....   | 23 |
| 2.2. Uurimismetoodika ja valim .....  | 25 |
| 2.3. Uuringu tulemused .....  | 27 |
| 2.4. Järeldused ja ettepanekud .....  | 34 |
| Kokkuvõte .....   | 37 |
| Viidatud allikad .....  | 39 |
| Lisad .....   | 46 |
| Lisa 1. Küsimustik .....  | 46 |
| Lisa 2. Originaalküsimustik (Shastri et al., 2021b) .....                       | 58 |
| Lisa 3. Tulemuste korrelatsioonitabel .....                                     | 70 |
| Summary .....   | 72 |

## SISSEJUHATUS

Erinevalt äriettevõtetest on avalikus sektoris domineerivaks praktikaks pikkade analüüsidega algav ning arendusfaasi jõudes tihtipeale juba vananenud nõuetega süsteemide arendamine. Riigikontroll on leidnud, et Eestis on probleeme riigi IT projektides tulemuste tarnimisega.

Tarkvaraarenduses on viimaste aastate jooksul toimunud suured muutused ning uute arendusmetoodikate kasutamiselevõtt. Järjest enam liigutakse klassikaliselt nn kosemudelilt agiilsele arendusele, mis peaks tagama parema tulemuse tarnimise. Agiilse arenduse juures on oluline muuta ka organisatsioonide käitumist ning autor otsib uuritavale organisatsioonile sobivat lahendust kuidas agiilset arendust praktiseerides asutust efektiivsemaks muuta. Agiilse meetodika juurutamise juures on oluline, et muutuks ka juhtimine (Hamunen, 2016).

Sarnasel teemal on kirjutatud viimasel ajal mitmeid töid, mis kajastavad näiteks DevOps printsiipide rakendamist Transferwise näitel (Rush, 2016) ning Telia näitel (Vene, 2017). Aastal 2019 ilmus magistritöö, kus uuriti agiilse lähenemise rakendusvõimalusi Keskkonnaministeeriumi (KeM) haldusala tarkvaraprotsessis (Lemberg, 2019). Autor leiab, et viimati nimetatud töös toodud järeldused ja ettepanekud ei pruugi enam uuemate teooriate valguses olla parimad ning teeb omapoolsed argumenteeritud ettepanekud koos konkreetsete rakendusettepanekutega (arendus- ja tarneprotsessi muudatused).

Teema on Keskkonnaministeeriumi Infotehnoloogiakeskuse (KEMIT) võtmes aktuaalne, kuna värskelt on läbi viidud struktuurimuudatused ning sellega seoses ka rollide ümber jagunemine. Kui hetkel kehtiv mudel on osakonnapõhine ja rangete ITIL (*IT Infrastructure Library*) reeglite alusel jäikade teenuse üleminekuprotsessidega ühe osakonna vastutusalast teise, siis uus struktuur on avatum ja kasutab tiimimudelit. Agiilse mudeli alusel toimetavat organisatsiooni võib vaadata kui improvisatsioonilist kollektiivi, kellel on küll üks eesmärk, kuid tee sinnani ei ole ette määratud (Du et al., 2019).

Keskkonnaministeeriumi haldusala projektid, nagu ka paljude teiste avaliku sektori IT majade projektid, kipuvad minema üle aja (Viira et al., 2019), kuna nõuded on muutumises, kuid kasutusel olev tarkvaraarenduse meetodika on jäik. Kuna ka tarkvaraarenduse trendid on muutumises, siis esitab autor küsimuse, et milline agiilse arenduse printsiipe kasutatav meetodika oleks KeMIT'is rakendamiseks mõistlikeim ning kas asutus on selleks valmis. Lisaküsimus on, et kas on võimalik juurutada agiilseid printsiipe nii, et arendus (*development, dev*) on mujal ja haldus (*operations, ops*) on KeMIT'is.

Uurimuse eesmärk on välja selgitada võimalused ja teha ettepanekud agiilse arenduse printsiipide juurutamiseks konkreetselt Keskkonnaministeeriumi haldusalas ja laiemalt avalikus sektoris, mis oma olemuselt on hierarhiline ja konservatiivne. Et mitte jääda vaid teoreetilisele tasemele, teeb autor ka konkreetsed ettepanekud protsesside muutmiseks. Kui olemasolevate protsesside asemele uusi alternatiivseid protsesse ei kehtestata, tingib see vanade protsesside ja uue struktuuri paralleelse toimimise, mis tõenäoliselt on teineteisele vasturääkivad (Vene, 2017).

Uurimisülesanded:

- kaardistada ning tuua välja võimalikud riskid, mis kaasnevad agiilsete protsesside kasutuselevõttuga;
- anda ülevaade erinevatest arendusmeetodikatest, tuginedes akadeemilisele kirjandusele;
- selgitada välja IT teenusepakkuja organisatsiooni valmisolek agiilseks arendusprotsessiks;
- teha ettepanekuid tarkvara arendusprotsessi muutmiseks;
- teha ettepanekuid tarkvara tarneprotsessi parendamiseks;

Uurimisülesannete lahendamiseks kasutab autor lisaks küsitlusele ka kvalitatiivset uuringut, et saada parim võimalik sisend järeldusteks.

Töö koosneb kahest peatükist. Esimeses peatükis annab autor ülevaate tarkvaraarenduse projektide juhtimise teoreetilisest taustast, tuginedes teaduskirjandusele ja mainekate organisatsioonide publikatsioonidele. Kirjeldatakse tarkvaraprojekti olemust ja erinevaid tarkvaraarenduse juhtimise meetodikaid. Autor annab ülevaate agiilse tarkvaraarenduse

printsiipidest ning kirjeldab organisatsioonilise ja tehnoloogilise muutumise vajalikkust agiilsete printsiipide kasutuselevõtuks.

Töö teine peatükk on empiiriline. Autor tutvustab uuritava organisatsiooni keskkonda, toodud on uuringu metoodika kirjeldus, tulemuste analüüs koos võimalike riskide ja probleemkohtade välja toomisega. Lisaks eelnevale on ära toodud ka autoripoolsed ettepanekud tarkvara arendus- ja tarneprotsessi parendamiseks.

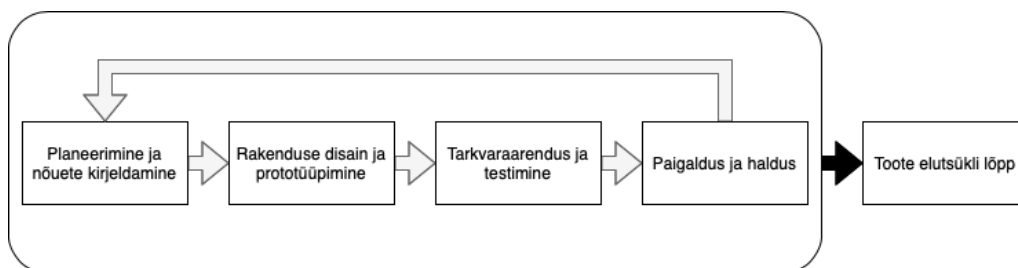
# 1. TARKVARAARENDUSE JUHTIMINE

## 1.1. Projektid ja tarkvaraprojektid

Projektid on kõikjal meie ümber, tajume me seda või mitte. Projektiks on nii kardinat panek kui kulleriga toidupoest kauba toomine. Akadeemilises mõistes on projekt ajutine tegevus, mis võetakse ette, loomaks unikaalset toodet, teenust või tulemust (Project Management Institute, 2017a). Projekti teine tähendus on ajutised organisatsioonid ehk projektiorganisatsioonid - vastavalt vajadusele ja eesmärgile kokku pandud inimeste ja ressursside hulk, mille ühine kestvus on määratud kas tähtaja või eesmärgiga. Samas ei ole ka erand see, kui organisatsioonid, eriti korduvate koostööprojektide korral, omavad püsivat sama, näiteks kliendipõhist projektimeeskonda.

Tihti on olnud kuulda väidet, et projekt on projekt ja ei ole vahet, mis projekti juhtida. Autor saab selle väitega nõustuda vaid osaliselt ja seda juhul, kui tegemist on lühiajalise, hästi piiritletud ja planeeritud ning väikest muudatust põhjustava projektiga - sellisel juhul on võimalik klassikalist ehk nn kosemeetodit (*waterfall*) kasutades hõlpsasti tulemuseni jõuda (Boutin, 2020). Sama eelduste hulk sobib nii ehitus- kui tarkvaraarenduse projektile. Ehitus- ja tarkvaraarenduse projektide omavaheline võrdlus on ka üks levinuimad, algavad ju mõlemad visioonist ning liiguvad edasi kavandamise ja arhitektuuri loomisega ning lõppevad toote või teenuse valmimisega, millele järgneb lammutamine (tarkvaraprojekti mõttes toodangust eemaldamine).

Kuna autor keskendub oma töös tarkvaraarendusele on joonisel 1 ära toodud lihtsustatud tarkvaraprojekti elutsükkel. Kui tavapäraselt kasutatakse seitset sammu tarkvara elutsükli kirjeldamisel (Jevtic, 2019) ning toodangust mahavõtmist ei käsitleta eraldi sammuna, siis autor on lihtsustanud joonist, arvestades agiilsete praktikate puhul erinevate arenduse juures kasutatavate praktikate suuremat integreeritust.



**Joonis 1.** Tarkvara elutsükel. Autori koostatud Jetvic, 2019 alusel

Triebert (2005) on toonud hea võrdluse tarkvaraarhitektuuri ja ehitiste arhitektuuri vahel. Ta toob välja, et ehituse puhul kavandatakse ehitise lõplikuna, tugevate seinte ja kindla vundamendiga ning keegi ei mõtle kunagi selle peale, et mõne aasta pärast võiks näitkeks 20-korruselisele majale ehitada juurde veel 20 korrust. Hea tarkvaraarhitektuur on aga vastupidiselt algusest peale kavandatud sellisena, et kasutatavad komponendid oleks hõlpsasti vahetatavad, lisatavad ja hooldatavad. Heal tarkvaral peab saama vahetada ka vundamenti, kui selleks peaks vajadus tekkima.

Kui klassikaline tarkvaraprojekti elukaar lõppeb tootest maha võtmise ja vajadusel uue tarkvaraga asendamisega, siis järjest kasvav mikroteenuseid kasutav arhitektuur (Fritsch et al., 2019; Mazlami et al., 2017) toetab kasutatava tarkvara funktsionaalsuste kaupa uuendamist, pikendades sellega tarkvara eluiga ning vähendades kulutusi ning riske. Rahvusvaheline projektijuhtimise assotsiatsioon IPMA on aasta 2021 kuulutanud vastutustundliku projektijuhtimise aastaks ning sellega kutsutakse üles väiksemale raiskamisele projektides (Thompson & Williams, 2021). Autor leiab, et see üleskutse on mõistlik ning mikroteenuste arhitektuuri kasutamine ning agiilsete printsiipide juurutamine aitab selle eesmärgi saavutamisele kaasa.

Tihti peale tellitakse rahastuse nõuetest tingituna avalikus sektoris agiilset arendust nii, et sõnastatakse hange, mille tulemuseks on fikseeritud hinna ja ulatusega pakkumine (*fix price*) ning seejärel hakkatakse agiilsuse nime all asju ringi tõstma, mis ei ole ju tegelikult keelatud. Kui aga lõpuks saavad tunnid otsa, siis otsitakse üles leping ja nõutakse seda, mis alguses pakkumuses oli kirjas. (Morales et al., 2018; Nuottila et al., 2016). On teada, et tarkvaraarenduse projekti käigus on nõuete muudatused pea vältimatud (Alsanad et al., 2019) ning sellega seoses on vältimatud ka muudatused esialgses lähteülesandes (Adamson, 2020). Muutused tarkvaranõuetes on klassikalise kosemudeli puhul lisaks



ebameeldivusele ka protsessi pärssivad, kuid on hädavajalikud, kuna muudavad lõpptoote kvaliteeti ja tagavad parema funktsionaalsuse. Klassikalise seadusemuudatusest tuleneva probleemina võib tuua välja ka ebamõistliku ajakava aktsepteerimise ilma suurema aruteluta (Sommerville, 2016), mis võib olla tingitud asjaolust, et rahalist katet muudatustele pole planeeritud või seaduse kehtima hakkamisel pole tehtud IT-alast analüüsi.

## **1.2. Levinuimad tarkvaraarenduse juhtimise meetodikad**

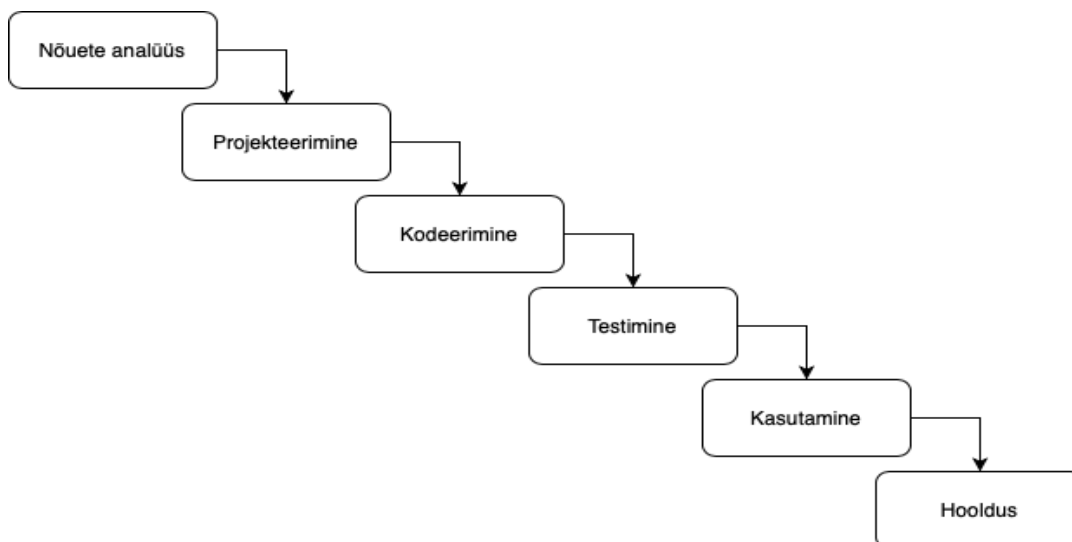
Tarkvaraarendus on loominguline töö ning seetõttu on aegade jooksul tekkinud ka suurel hulgal erinevaid lähenemisi, kuidas tarkvaraarendust juhtida. Siiski saab tarkvaraarenduse juhtimise meetodikad jagada konsultatsioonifirma PWC uuringu (Jonnalagadda et al., 2017) alusel suures plaanis traditsiooniliseks kosemudeliks ja agiilseks - agiilne omakorda mitmeteks alamhulkadeks, milliste juurde tuleb autor hiljem. Nii kosemudelil kui ka agiilsel lähenemisel on oma kindlad kasutuskohad, kus nad võiksid töötada paremini kui teine mudel ning seetõttu ei ole võimalik üheselt otsustada, milline mudel on teisest parem (Špundak, 2014). Vastukaaluks traditsioonilisele tarkvaraarendusele sai 90ndatel alguse agiilne lähenemine. Agiilsest tarkvaraarendusest hakati rohkem rääkima ja see muutus järjest levinumaks 2000. aastate algul, mil avaldati „Agiilse tarkvaraarenduse manifest“ (Beck et al., 2001), mille avaldamise olulisimateks põhjuseks oli asjaolu, et paljud tarkvaraprojektid ebaõnnestusid hoolimata sellest, et kasutasid kindlat plaani ning järgisid ettemääratud protsessi. „Agiilse tarkvaraarenduse manifest“ sõnastas agiilse tarkvaraarendusprotsessi üldised põhimõtted ja põhiväärtused. Agiilne tarkvaraprotsess on oma ülesehituselt paindlik ning selles üheks oluliseks tunnuseks on nii tarkvaraarendusmeeskonna liikmete kui ka kliendi kaasamine tavapärasest suuremal määral (Špundak, 2014). Agiilsete meetodikate juures rõhutatakse, et tarkvaraarendusprojekti õnnestumiseks kõige tähtsamad tegurid on projektis osalejate omadused ja nendevahelise koostöö kvaliteet, oluliselt vähem oluliseks peetakse kindlaksmääratud protsessi ja töövõtete kasutamist (Beck et al., 2001; Bhuiyan & Baghel, 2005). Tabelis 1 on autori poolt välja toodud põhilised erinevused nende vahel.

**Tabel 1.** Klassikalise ehk kosmeetodi ja agiilse meetodikate võrdlus

| Võrdlusalus                | Kosmeetod  | Agiline meetod   |
|----------------------------|--|--|
| Nõuded                     | väga selged esialgsed nõuded; vähene muutuste arv              | uuenduslik ja loov, ebaselged nõuded                     |
| Kasutajad                  | ei kaasata   | tihe ning pidev omavaheline koostöö                      |
| Dokumentatsioon            | nõutud kindel dokumentatsioon                                  | dokumenteerimise aste madal, teadmised meeskonna sees    |
| Projekti suurus            | suured projektid   | väiksemad projektid                                      |
| Organisatsioonipoolne tugi | suuremad organisatsioonid; kasutatakse olemasolevaid protsesse | võime austada agiilset lähenemist ja sellega kaasa minna |
| Töörühma liikmed           | liikmed määratud; liikmed vahetuvad; meeskond hajutatud        | väiksem meeskond; samas asukohas                         |
| Süsteemi kriitilisus       | tagajärjed süsteemi tõrke korral tõsised                       | vähemkriitilised süsteemid                               |
| Projektiplaan              | lineaarne, ettemääratud  | keerukas ning pidevas muutumises                         |

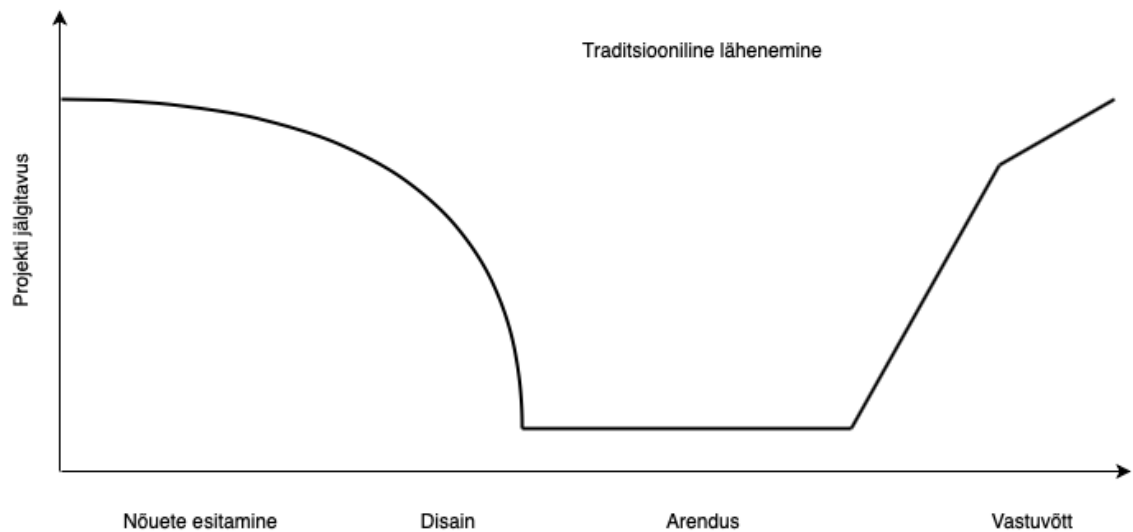
Allikas: Špundak, 2014

Tavapärane tarkvaraarenduse protsess on üksteisele järgnevate tegevuste jada, mis algab kliendipoolsete vajaduste defineerimisest ning lõpeb vajadusi enam mitte rahuldava tarkvara toodangukeskkonnast eemaldamisega (Špundak, 2014). Agiilne tarkvaraprotsess on välja arenenud traditsioonilisest lineaarsest lähenemisest, tuntuim traditsiooniline tarkvaraprotsessi mudel on nn kosemudel (*Waterfall*). Traditsiooniline tarkvaraprotsess järgib kosemudelit (Joonis 2), mille puhul sooritatakse tarkvara ehitamisel lineaarselt (tsüklite kaupa) arenduseks vajalikud tegevused nagu süsteemi nõuete analüüs, disainimine ehk projekteerimine, kodeerimine ehk tarkvara ehitamine, tulemite testimine, tarkvara kasutamine ja järelhooldus. Töö järjestuse loogika on selline, et korraga tehakse ära üks terve tsükkel ja alles seejärel võetakse töösse uus. (Cohn et al., 2009)



**Joonis 2.** Traditsiooniline tarkvaraarenduse mudel ehk kosemudel. Allikas: Autori koostatud (Cohn et al., 2009) alusel

Kosemudelit kasutatakse siis, kui lahendamist vajav probleem ja lõpptulemusele esitatavad nõudmised on selgesti defineeritud või juhul, kui tehakse üheselt arusaadavaid muudatusi olemasolevale süsteemile. Kosemudelit järgides on kokkulepitud lähteülesande muutmise tülikas, kuna kosemudeli meetoodika näeb nimelt ette, et kliendi soovide korje ja nende arvesse võtmine ning ka kõik muud olulised arendusprotsessi puudutavad otsused langetatakse ühekordselt tarkvaraprotsessi varajases faasis. Kosemudeli suurimaks puuduseks on paindumatus projekti eri etappides ning selle tulemuseks on see, et klient näeb projekti tulemust alles peale arenduse lõppu ehk üleandmisel. (Serrador & Pinto, 2015; Špundak, 2014). Kosemudeli puhul on tarkvaraarenduse projekti jälgitavus (*visibility*) väga kõikuv, mida ilmestab Joonis 3. Nagu jooniselt nähtub, siis on jälgitavus suurim projekti kavandamisel ehk nõuete esitamise ja disainimise faasis ning seejärel väheneb läbipaistvus drastiliselt realiseerimise ajal ning taastub alles vastuvõtufaasis (Jonnalagadda et al., 2017).



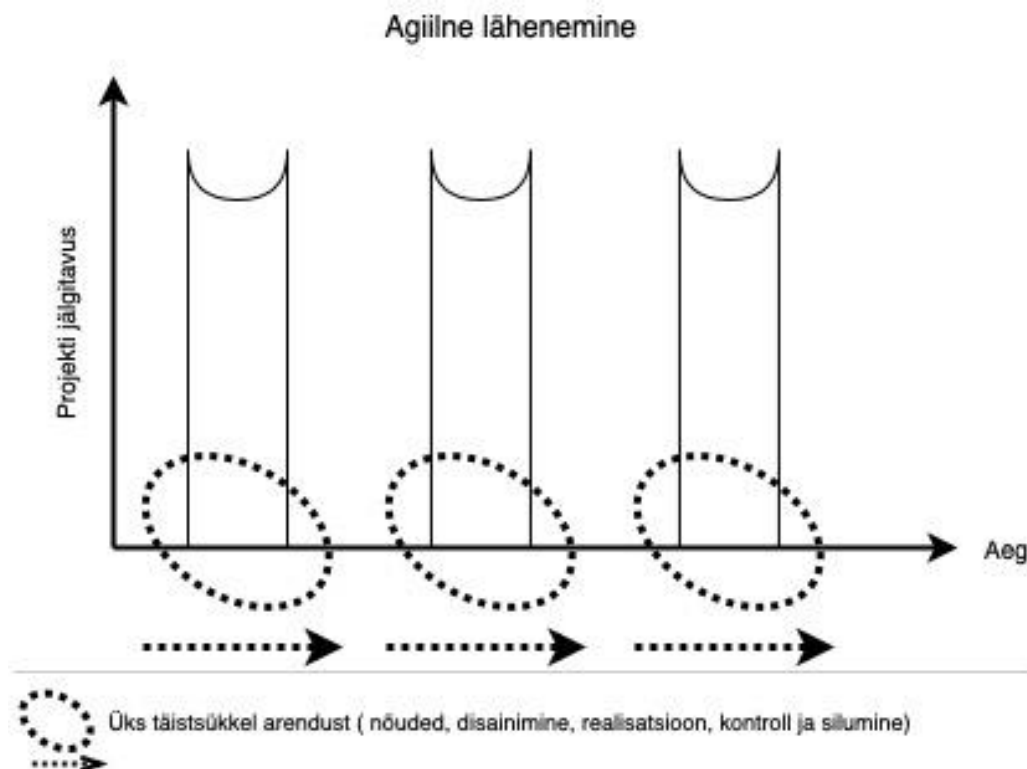
**Joonis 3.** Projekti jälgitavus kosemudeli puhul eri etappides. Allikas: (Jonnalagadda et al., 2017)

Serrador ja Pinto (2015) analüüsisid 1386 projekti ning leidsid, et 6% neist olid läbi viidud kas läbinisti agiilse või selle lähedase metoodikaga ja üle 65% projektidest omas mõnda agiilset komponenti. PMI poolt koostatud raporti järgi kasutab 79% vastanutest oma projektides agiilseid metoodikaid (Project Management Institute, 2019) ning aasta 2000 andmetel on 53% organisatsioonidest seadnud kõrgeks prioriteediks organisatsioonikultuuri viimise agiilseid metoodikaid toetavaks (Project Management Institute, 2020). Metoodikad sisaldavad endas protsesside kirjeldusi, erinevaid standardeid ning tööriistu ja tehnikaid (Cybernetica AS, 2020). Agiilsed metoodikad määratlevad, et projektimeeskonna liikmete omadused ning nende koostöö on olulisem kindlaksmääratud protsessist (Beck et al., 2001; Bhuiyan & Baghel, 2005).

Agiiilse arenduse puhul on kliendi ja kliendile lahendust loova teenusepakkuja, kas siis sisse ostetud partneri (*outsourced team*) või oma organisatsiooni inimestest koosneva (*inhouse team*) arendusmeeskonna suhtlus viidud tihedamaks ja töötatakse ühe meeskonnana sama eesmärgi nimel. Agiilsetes metoodikates tuuakse välja erinevad rollid, mida tuleb meeskonnal ühiselt täita, kuid klassikalise projektijuhtimise roll ei ole agiilsetes metoodikates defineeritud (Project Management Institute, 2017b; Shastri et al., 2021a). Üks põhimõtteid, mida agiilse arenduse juures järgitakse on, et väga selgelt

defineeritud tööd saab ja lausa peab automatiseerima (Project Management Institute, 2017b).

Agiilse arenduse puhul on trumbiks arendustsüksi lühiajalisus ning iga tsükli järel toodangusse paigaldamiseks valmis oleva tarkvarakomponendi või -funktsionaalsuse valmimine. See võimaldab kiiresti viia sisse vajalikke muudatusi ning arendust tellivale ettevõttele oma investeeringut tagasi teenima hakata. Võrreldes kosemudeliga on projekti jälgitavus (*visibility*) tellijale oluliselt suurem, nagu Joonis 4 vahendusel on näha.



**Joonis 4.** Projekti jälgitavus agiilse arendusmudeli puhul. Allikas: (Jonnalagadda et al., 2017)

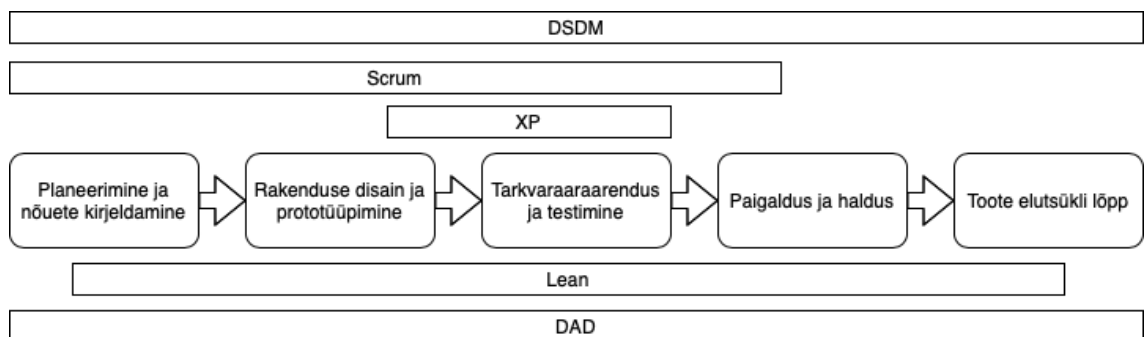
Agiilset arendust võib vaadata kui järjestikuliste ja lühiajaliste Demingi poolt defineeritud PDCA (*Plan-Do-Check-Act*) tsüklite korduvust (Jonnalagadda et al., 2017), mis on veidi kohendatud kujul, kuid kannavad sama eesmärki.

Kõikide organisatsioonide küpsust hinnatakse küpsusmudelitega, mis on välja töötatud vastavalt organisatsiooni tüübile või hinnatavale väärtusele. Organisatsioonide agiilse arenduse küpsust mõõdetakse agiilsete küpsusmudelitega (*agile maturity models, AMM*),

milliseid on akadeemikute, agiilsuse propageerijate ja tarkvaraekspertide poolt välja pakutud üle neljakümne (Nurdiani et al., 2019). Autor leiab, et erinevate küpsusmodelite kirjeldamine ja nende tööpõhimõtete lahti rääkimine vääriks eraldiseisvat tööd ning käesolevas töös on see toodud lihtsalt informatsiivse äramärgimisena, kuna käesoleva töö eesmärk ei ole kasutada organisatsiooni hindamiseks küpsusmodelit, vaid hinnata pigem organisatoorsete ja tehnoloogiliste eelduste olemasolu.

### 1.3. Agiilsete arendusmetoodikate ülevaade

„Agiilne manifest“ (Beck et al., 2001) on välja antud juba 20 aastat tagasi. Seda ajaraami arvestades on loogiline, et „Agiilse manifesti“ põhimõtteid järgides ja arvesse võttes on tekkinud palju erinevaid lähenemisi, mis kõik pakuvad omapoolset tõe, et kuidas oleks kõige õigem agiilselt tarkvara arendada. Arvestades suurt hulka erinevaid populaarseid agiilseid arendusmetoodikaid on autor trendidele (Himmat & Osman, 2020) tuginedes valinud neist tutvustamiseks välja enamlevinud metoodikad. Metoodikate kattuvus on toodud Joonis 5, mis annab visuaalse üldvaate, millisele tarkvara elutsüklist on konkreetne metoodika keskendunud.



**Joonis 5.** Enamlevinud metoodikate tarkvara elutsükli katvus. Autori koostatud (Agile Business Consortium, 2014; Ambler & Lines, 2019, 2020; Bell, 2017; Dalla, 2020; Digital.ai Software Inc., 2020; Rajkumar et al., 2016; Saltz et al., 2020; Shastri et al., 2021a; Sohaib et al., 2019; Špundak, 2014; Wiedemann et al., 2019) alusel

**Scrum** näol on tegemist arendusmetoodikaga, mis seostub paljudel sõnaga agiilsus. Viimase State of Agile raporti järgi kasutab Scrumi lausa 58% kõigist agiilse metoodika kasutajatest (Digital.ai Software Inc., 2020). Scrum on välja töötanud selged juhised, mis on suunatud eelkõige toote kiirele valmimisele ja pidevale tarnimisele. Ühest küljest on

Scrumi protsess oma printsiipide ja reeglitega on sirgjooneline ning arusaadav, samas võib niisugune lihtsus olla teisest küljest petlik, sest tegemist ei ole päris üheselt käsiraamatuna kasutatava juhendiga. Scrumi protsess keskendub pigem ideaalolukorrale ja ei ütle ette, kuidas eriolukordade juhul käituda. Vastupidiselt kõigi elualade erinevate projektide tüüpe katvale klassikalisele projektijuhtimisele on Scrum-metoodika loodud ennekõike silmas pidades tarkvaraarenduse projektide spetsiifikat - ei ole ju arendusprojektides alati võimalik näha ette kõiki võimalikke muudatusi, ei majanduses toimuvate, ühiskondlike või tehnoloogiliste muudatuste tõttu. Scrumi keskmes on avatud suhtumine probleemidesse ja nende ühine lahendamine, saadud õppetundide ja teadmiste jagamine, inimeste omavaheline suhtlus eesmärgiga pakkuda kliendile vajaliku väärtust töötava tarkvara näol. Scrum-metoodika puhul on tööd jaotatud väikesteks etappideks ehk sprintideks ning see tagab, et nii klient kui meeskonnaliikmed on projekti vältel kursis, kes millega tegeleb või milliseid ülesanded hetkel lahendatakse. (Dybå & Dingsøyr, 2008; Khmelevsky et al., 2017; Shastri et al., 2021b). Scrumi puhul on väga olulisel kohal kindlad rutiinsed tegevused. Scrumis nimetatakse selliseid tegevusi tseremooniateks (Paasivaara, 2021) ning nende kestvus, toimumissagedus ja neist osavõtu olulisus on puhtakoelise Scrum-metoodika puhul äärmiselt tähtsad (Griffin, 2021) ning nendest mitte kinni pidamine tähendab mittepuhta Scrummetoodika kasutamist. Nii Griffin (2021) kui ka Paasivaara (2021) leiavad, et tegelikult saaks ja isegi peaks Scrum-tseremooniaid vähendama või muutma ehk tuleks kasutada Scrum-metoodika peal Lean-võtteid.

**XP** (*eXtreme Programming*) metoodika on välja töötatud 90ndate lõpus Kent Beck'i poolt ajal, kui ta töötas Chrysleri palgaarvestuse projekti juures (Bell, 2017). XP keskendub arenduse tehnilisele realisatsioonile, kasutades paarisprogrammeerimist, ühikteste (*Unit tests*), väikeste osade kaupa tarnimist. XP eesmärgiks on tarnida kliendile ainult sellist funktsionaalsust, mis loob väärtust ning selleks kasutatakse minimaalse elujõulise toote loomist (*Minimal viable product, MVP*). Selle tagamiseks on meeskonna liikmed kõrge kvalifikatsiooni ja kogemustega. XP puhul, nagu ka teiste agiilsete lähenemiste puhul, on väga oluline pidev suhtlus kliendi ja arendajate vahel. Kliendilt oodatakse kiiret reageerimist ning kohest testimist. Kliendi kiire tagasiside võimaldab tagada, et kliendile tarnitakse just selline toode, mida klient tegelikult vajab. (Beck, 1999; Sohaib et al., 2019). XP-metoodika on arendusmeeskonna sisemise paindlikkuse näol antud vastus kliendi muutuvatele soovidele.

**DSDM** ehk dünaamiline süsteemiarendusmetoodika on agiilsetest tarkvararenduse metoodikatest ehk kõige põhjalikum. Selle rajajaks ja haldajaks on DSDM-konsortium Suurbritanniast. Konsortiumi moodustavad Suurbritannia suurimad IT-teenuste pakkujad (Agile Business Consortium, 2014). DSDM-i järgi tuleb esmajoones keskenduda ressursside ja ajakasutuse parandamisele ning alles peale seda kohandada väikeste sammudena toote funktsionaalsust. Eduka projektitulemi saavutamiseks on DSDM järgi vajalik pidev prototüüpimine. Prototüüpimine tagab, et kõik muudatused ning täiendused on kirjeldatud, et seejärel viia need arendusprotsessi abil sisse. Iga etapp koos oma tulemiga peab olema piisavalt lõplik, et saaks alustada järgneva etapiga. Iga järgneva etapi käigus on võimalus eelnevates etappides lõpetamata tööde lõpetamiseks. Projekti saadab DSDM'i järgi edu juhul, kui paika on pandud selged ärilised eesmärgid, tarkvaratarned toimuvad piisavalt tihedalt ja meeskonnaliikmete hulgas hoitakse kõrget motiveerituse taset. Oluline on, et meeskonnaliikmetele oleks tagatud kõik vajalikud volitused nii tegutsemiseks ka otsustamiseks. (Agile Business Consortium, 2014). Erinevalt muust maailmast ei ole DSDM autori hinnangul Eestis levinud.

**Lean**-i peamisteks põhimõteteks on raiskamiste ja üleliigse kõrvaldamine (*eliminating waste*) ning kliendile väärtuse lisamine ning Lean-i kohaselt ebaefektiivsed tegevused eemaldatakse arendusprotsessist (Dalla, 2020; Saltz et al., 2020). Eesmärkide saavutamiseks kasutatakse erinevaid tehnikaid ja meetodeid, mida kirjeldavad järgmised omadused (Dalla, 2020) :

- probleemide leidmine on lihtne,
- probleemide lahendamine on lihtne,
- tehtud vigadest õppimine on lihtne.

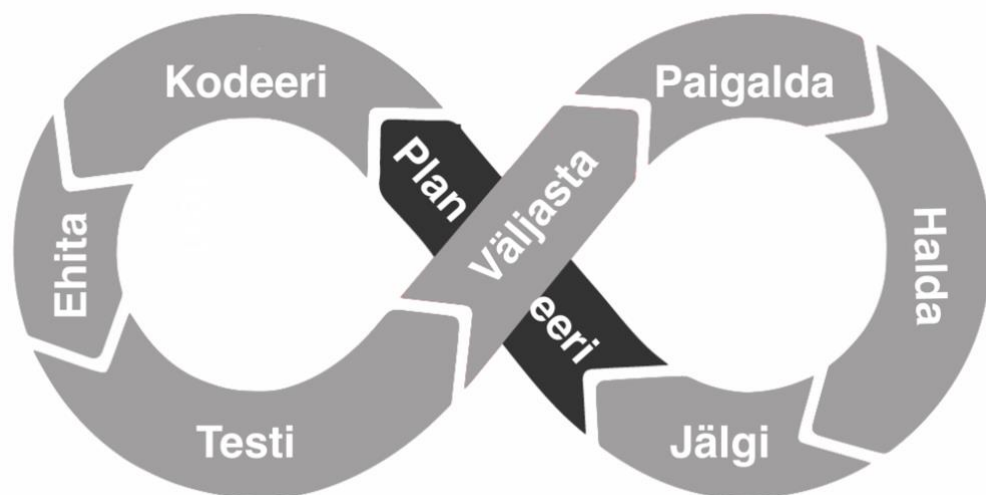
Lean-i lähenemise kohaselt ei tohi lõplikku tootesse jõuda mitte ükski funktsioon, mis ei tooda kliendi jaoks väärtust. Selline lähenemine rakendub kogu arendustsüklile ja seda alates tellimuse saamisest kuni vastuvõtuakti ja arve esitamiseni. (Saltz et al., 2020). Leani põhifilosoofia ongi ebaolulise eemaldamine (lean terminoloogias *muda*) protsessist.

**DevOps** määratletakse enim metoodika või tehnika või protsessina, kuid tegelikult on see mõtteviis, mis on arenenud vastusena arendus- ja haldusmeeskondade vähesele



koostööle. Pidev vastasseis tarkvara arendusprotsessi ja tarkvara paigaldamise praktikate üle on viinud protsesside ja tavade revolutsioonini, mis on osa DevOps'i kultuurist. Selline areng pole mitte ainult aidanud ületada lõhet muidu traditsiooniliselt eraldiseisvate arendus- ja haldusmeeskondade vahel, vaid DevOps'i kultuur on toiminud võimaldajana, et pakkuda pidevalt rohkem funktsioone, säilitades samal ajal stabiilsuse. (Rajkumar et al., 2016)

Samuti võib DevOps'i printsiipe kirjeldada ka erinevate tarkvaraarendusmetoodikate kogumina (Rush, 2016). Mõiste DevOps on tekkinud ingliskeelsete terminite arendus (*development*) ja haldus (*operations*) kokku liitmisest. DevOps ei ole raamistik, vaid selle all mõistetakse agiilse arenduse viisi, kus osakondade vahelised barjäärid on eemaldatud ning tegutsetakse ühise eesmärgi nimel multifunktsionaalsetes meeskondades (*crossfunctional teams*). („What Is DevOps?“, s.a.; Wiedemann et al., 2019). DevOps'i eesmärk on pidev parendamine ja täiustumine ning sellist lõputut jada nimetatakse DevOps elutsükliks (Porras, 2021). Joonis 6 ilmestab, kuidas arendus (vasakpoolne) ja haldus (parempoolne) on DevOps tsükli mõttes selges sümbioosis ning ühe eesmärgi täitmiseks.



**Joonis 6.** DevOps elutsüklil. Autori koostatud Porras, 2021 alusel

Eelkõige tähendab DevOps tarkvaraarendusspetsialistide (*development*) tihedat koostööd haldusega tegelevate spetsialistidega (*operations*), tihtilugu ka traditsiooniliselt nende erialaks olnud tööde enda kanda võtmist, näiteks andmebaaside või süsteemide

administreerimine. Vene (2017) on leidnud, et võiks kasutada isegi väljendit *Developing Operations* ehk süsteemide halduse arendamine. Ettevõtte need liikmed, kelle tööks ei ole otseselt tarkvara arendamine, näiteks süsteemi-, võrgu- või andmebaasiadministraatorid, kasutajatugi ja tarkvaratestijad on DevOps maailmas asendamatuteks abilisteks arendajatele toe ja parandusettepanekute tegemisel (Wiedemann et al., 2019). Laiemas käsitluses ei puuduta DevOps koostööd vaid IT arendus- ja haldusosakonna vahel, vaid kaasab ka ettevõtte äripoole. (Olagunju, 2018; Špundak, 2014)

Viimasel ajal on ka Projektijuhtimise instituut (PMI) järjest enam hakanud tähelepanu pöörama agiilsetele printsiipidele ja metoodikatele. Algselt IBM poolt välja töötatud ja hiljem PMI poolt koordineeruna on Disciplined Agile Consortium välja pakkunud lähenemise (*toolkit*) mida nimetatakse distsiplineeritud agiilsuseks (*Disciplined Agile, DA*)(Project Management Institute, s.a.). PMI keskendub täpsemalt distsiplineeritud agiilsele tarnele (*Disciplined Agile Delivery, DAD*) (Project Management Institute, s.a.), mis on ka loogiline, kuna kogu senine PMI poolt propageeritud tarkvaraarenduse loogika on tarnepõhine.

**DA/DAD** katab ära kogu protsessi alates meeskonna moodustamisest lõpetades tarkvara tarnimisega ja ta on segu (*hybrid*) Scrum, XP, Lean ja teiste meetodite parimatest osadest (Ambler & Lines, 2019). Värskema publikatsiooni järgi on PMI laiendanud DA/DAD ulatust ning võtnud uue tasemena (*level*) kasutusele ka termini distsiplineeritud DevOps (*Disciplined DevOps*), mis rikastab DA/DAD'd DevOps printsiipidega (Ambler & Lines, 2020).

Projektijuhtimises toob hübriidmetoodika kasutamine endaga tõenäoliselt kaasa niinimetatud Pareto<sup>1</sup> parendused, võimaldades neid saavutada sama ajakava, eelarve ja kvaliteediga, mis traditsioonilised metoodikad (Gemino et al., 2021). Hübriidmudel on ka autori kogemusele tuginevalt kõige kergemini juurutatav, kuna võimaldab vastavalt organisatsiooni valmisolekule järk-järgult lisada järgmiseid agiilsuse komponente.

---

<sup>1</sup> Kaupade või muude ressursside ümberpaigutamine mille tagajärjel vähemalt üks inimene tajub olukorra paranemist, seejuures kellegi teise olukord ei halvene (Gemino et al., 2021).

Kui DevOps, DAD ja Lean on oma olemuselt pigem mõtteviisid kui käitumisreeglid, siis Scrum ja DSDM on teineteisele suhteliselt sarnased ning annavad kasutajale ette suhteliselt selgete reeglitega protsessi (*flow*), mida järgides on tulemuse saavutamise tõenäosus suurem.

#### **1.4. Eeldused agiilsete printsiipide rakendamiseks**

Agiilsete printsiipide rakendamine, nagu iga teine muudatus organisatsioonis vajab hoolikat planeerimist ning läbi kaalumist. Muudatused, mis peavad organisatsioonis toimuma, et agiilsed printsiibid kasutusele võtta, saab jagada kaheks – organisatoorsed ja tehnoloogilised.

Organisatoorsete muudatuste all mõeldakse, et agiilsete printsiipide, eriti aga DevOps printsiipide juurutamise üheks esmaseks eelduseks on see, et kõik organisatsiooni eri tasandid – nii juhid kui spetsialistid saavad üheselt aru lahendatavast probleemist, meetmetest, millega probleemi plaanitakse lahendada tulemuse saamiseks ning on teadlikud ka takistustest, mis tuleb lahenduse edukaks lõpule viimiseks ületada (Vene, 2017). Agiilse arenduse juurutamise juures on hea juhtimisoskus kohustuslik, et integreerida vajaduspõhine ja agiilne lähenemine, nii juhtkond kui ka iga meeskonnaliige peavad aktiivselt edendama agiilsete arendustiimide läbipaistvust ja avatud suhtlemist (Niever et al., 2020). Klassikalises projektijuhtimises on meeskonna kooshoidmise ja koordineerimise roll olnud projektijuhil, agiilses arendusmeeskonnas on aga meeskonnad isekohenduvad ja juhtimisfunktsiooni täidetakse ühiselt. Agiilses arenduses on projektijuhi roll häägustunud ning laialt kasutatavas Scrum-metoodikas ei ole sellist rolli üldse (Shastri et al., 2021a). Oma artiklis leidsid Shastri ja teised (2021a), et projektijuhi roll on säilinud tegelikult endiselt ja projektijuht võib aidata meeskonda, olles üks täisväärtuslik liige ja täites mitteagiilseid ülesandeid nagu näiteks eelarvestamine ning projekti jälgimine.

Maailm on muutumises ning klassikalise projektijuhtimise ja sirgjooneliste projektide asemel tuleb järjest rohkem keskenduda kiirete muudatuste juhtimisele (Adamson, 2020). Kiire muudatuste juhtimine toob endaga kaasa ka mõttemaailma muutuse

organisatsioonides, kus organisatsioonid peavad muutuma paindlikumaks ehk rohkem improviseerima (Kuura & Sandoval, 2019), et jääda konkurentsisis ellu.

Du jt (2019) eristavad tarkvaraarenduses kahte suuremat improvisatsioonilist tegevust, milleks on improvisatsiooniline otsing ning improvisatsiooniline ülesehitus. Improvisatsioonilise otsingu sisuks on pideva tagasiside tõttu muudatusvajaduste leidmine. Improvisatsioonilise ülesehituse all peetakse silmas, et keskkond, milles arendusmeeskond töötab, on üles ehitatud nii, et see võimaldab paindlikult ja mugavalt reageerida improvisatsioonilise otsingu käigus leitud muudatustele. (Du et al., 2019)

Järjest enam leiame, et tarkvaraarendajate ja rakendusadministraatorite puhul otsitakse lisaks headele tööalastele oskustele ka niinimetatud pehmeid oskuseid (Totah, 2021). Ettevõtte töötajate mõtteviisi saab muuta ainult siis, kui mõlema osapoole (arendus ja haldus) vahel on õiged suhtluskanalid. Agiilse arenduse, eriti DevOps printsiipide puhul näeme, kus disainimõtlemine ja agiilne metodika aitavad viia mõlema osapoole eesmärgid vastavusse nii, et mõlemad osakonnad on üksteise huvigrupid teise poole tehnilises realisatsioonis. (Rajkumar et al., 2016; Sohaib et al., 2019). Inimestele pakutavad töövahendid peaksid vähendama käsitsi asjade tegemist, võimaldades tõhusaid ja lühemaid paigaldustsükleid standardsete tootmisesse paigalduse, kvaliteedi tagamise ja arendamisprotsessiga. Otsustajate seisukohast võivad protsessi paremaks muuta järgmised strateegiad (Rajkumar et al., 2016):

- tehke kindlaks protsessid ja tööriistad, mis võimaldavad DevOps meeskonnad teie äristrateegiaga heasse vastavusse viia,
- suurendage meeskonna koostöötahet, et saavutada mastaabiefekteri teadmiste koondumisest,
- kõik meeskonnaliikmed ehk mängijad on sprindis vastavate lugude mõlema osakonna võtmepartneriks,
- protsesside lihtsaks ja mänguliseks viimine stressi vähendamiseks ja meeskondade agaruse suurendamiseks.

Lisaks organisatorsetele muudatustele tuleb suurt tähelepanu pöörata ka tehnoloogiliste eelduste täitmisele. Avaliku sektori organisatsioonid on järjest kasvava surve all, et nad oma teenuseid paremaks muudaks (Vestues et al., 2021), kuid põrkuvad selle otsa, et

avaliku sektori süsteemid on loomu poolest suured ja keerulised, mistõttu innovatsiooniga kaasaskäimine võtab juba partneri valiku ehk hankeprotsessi tõttu kauem aega kui erasektoris (Nuottila et al., 2016).

Alljärgnevalt on toodud autoripoolne valik kõige enam esinevaid takistusi, mis võivad väga reguleeritud keskkonnas DevOps printsiipide juurutamisel ette tulla (Morales et al., 2018):

- ligipääs tarkvara paigalduskeskkondadele piiratud,
- aeglane projektiga seotud otsuste kinnitamine,
- riist- ja tarkvara aeglane hankeprotsess,
- arendajal puudub võimalus suhelda otse lõpptellijaga,
- juurdepääs tootmiskeskonnale on osaline või puudub üldse,
- puudub automaatpaigalduse süsteem.

Tehnoloogiliselt on leitud lahendused, mis on tõestanud ennast suurtes toodangusüsteemides ja võimaldavad ka pidevat pidevat turvamonitoringut (Amoroso, 2018), see tähendab, et paljud rakendatud meetmed ei ole tänapäeval enam kohased. Arvestades seda, et maailm liigub mikroteenuste poole (Gil & Díaz-Heredero, 2018; Yuan, 2019), siis klassikaline kaitsemeede ehk võrkude piiramine on tänapäeval pigem segav faktor. Vene (2017) on leidnud, et *“kuni tehnoloogia pole arenenud tasemele, kus kõik on automatiseeritud, vajavad tarkvaraarendajad partnerit platvormide ning toetavate funktsioonide valdkonnas”*. Konteksti arvestades pidas Vene (ibid.) autori hinnagul oma lausega silmas seda, et klassikaline arvamus, et administraatorite järgi vajadus kaob, ei vasta tõele, nende töö profiil aga muutub.

On väga oluline, et pideva integratsiooni ja tarne käigus toimuks ka pidev testimine. Harjumuspärase kosemudeli puhul toimus testimine vastuvõtmisel, kuid agiilse arenduse puhul ning kasutades järjest enam populaarsust koguvat mikroteenustel põhinevat ülesehitust (Gil & Díaz-Heredero, 2018; Yuan, 2019), mis tähendab suurt kogust teenusliidesid (API), peab testimine olema protsessi sisse kirjutatud juba algusest peale (Cope, 2020).

Üheks oluliseks ideeks DevOps liikumise taga on lisaks asutuste sisesesele tarkvara arendust soodustavale efektiivsele suhtlus- ja koostöömudeli pakkumisele ka totaalne

automatiseerimine. Süsteemide automatiseerimine ja integreerimine toimub eesmärgiga minimeerida aeg arendaja poolt valminud koodi jõudmiseks toodangukeskkonda. Kiire tarne vajaduse tõttu on DevOps tihedalt seotud pideva integreerimise (*Continuous Integration, CI*) ja pideva tarne (*Continuous Delivery, CD*) põhimõtetega. Sellised pidevale tarnele suunatud põhimõtted aitavad ettevõtetel oma arendustegevuses kiiremini reageerida turuolukorra muutustele. (Olagunju, 2018; Rush, 2016; Wiedemann et al., 2019). Pideva integreerimise ja pideva tarne protsessid (*pipelines*) on ühed oluliseimad tehnoloogilise valmisoleku teemad, millele tähelepanu pöörata, et vältida inimfaktori masinaga asendamise tulu ja sellest tuleneva võimaliku kahju minimeerimist (Paule et al., 2019). Autor leiab, et nendel teemadel tuleb koostada ka küsimused, mida esitada intervjueeritavatele, et avada paremini nende kogemust agiilse meetodikaga.

## **2. AGIILSE TARKVARAARENDUSE RAKENDAMINE KESKKONNAMINISTEERIUMI HALDUSALAS**

### **2.1. Uurimisobjekti tutvustus**

Antud töös viiakse autori poolt läbi uuring Keskkonnaministeeriumi Infotehnoloogiakeskuses (KEMIT) ja KEMITi poolt teenindavates Keskkonnaministeeriumi allasutustes. Töö uurimisobjektiks on agiilsete arendusmetoodikate kasutamine ja juurutamine tarkvaraarenduse projektide juhtimisel.

KEMIT kuulub Keskkonnaministeeriumi haldusalasse ehk KEMITit koordineeriv valitsusasutus on Keskkonnaministeerium. KEMIT moodustati 1. jaanuaril 2013, uue asutuse loomiseks ühendati Keskkonnaministeeriumi haldusala asutuste IT-töötajad ja toodi üle nende funktsioonid. Infotehnoloogiliselt on süsteemide konsolideerimine lõppenud, kuid kõik funktsioonid ei ole veel lõplikult keskselt KEMITi kätte koondunud, on endiselt asutusi, kus on oma infotehnoloogilise arendusega tegelevad rakukesed. Asutused, mida KEMIT teenindab, on Keskkonnaministeerium, Maa-amet, Keskkonnaagentuur, Keskkonnaamet, Loodusmuuseum, ja KEMIT ise (KeMIT, 2021). Organisatsiooni struktuur, ülesanded ja juhtimise korraldus on kindlaks määratud KEMITi põhimääruses. Vastavalt põhimäärusele on Keskkonnaministeeriumi Infotehnoloogiakeskuse põhiliseks ülesandeks „ministeeriumi ja selle valitsemisalas asuvate allasutustele infosüsteemide arendamine ja haldamine ning tehnoloogilist komponenti sisaldavate hangete läbi viimine“ (*Keskkonnaministeeriumi Infotehnoloogiakeskuse põhimäärus*, 2020).

KEMITi struktuuris tegeleb tarkvaraarendusega teenuste valdkond, mis on jagatud meeskondadeks vastavalt kliendiprofiilidele, see tähendab, et meeskonnad on püsivad. Meeskonnad on ideaalis täisfunktsionaalsed ehk katavad ära kõik tarkvara elutsükli protsessid alates arenduse planeerimisest, lõpetades selle haldamise ning tootest maha

võtmisega. Meeskondasid juhivad meeskonnajuhid, kes on ka autori poolt valitud küsitletavateks.

Keskonnaministeeriumi haldusalas toimuvat ja KEMITi poolt korraldatavat tarkvaraarendust reguleerivad IT-korralduse raamistik (Arula, 2020) ning KEMIT-i tarkvara arendusprotsess (Viik, 2018), mis on hetkel klassikaliselt rangete ITIL (*IT Infrastructure Library*) reeglite alusel ja jäikade teenuse vastutuste üleminekutega. Olemasolev arendusprotsess ei sobi autori hinnangul muutunud struktuuri alusel toimiva arenduse kirjeldamiseks ja vajab muutmist. Hetkel kehtivas arendusprotsessis on viis etappi (Viik, 2018) ja nendeks on:

- arendusvajaduste esitamine,
- ärianalüüs,
- arendamine,
- testimine,
- rakendamine.

KEMITi arendusprotsess on kose mudelipõhine, selle erinevad etapid on lahti kirjutatud tegevusteks ja iga tegevuse juures on välja toodud antud tegevusega seotud rollid, et oleks üheselt selge, kes mida tegema peaks (Lemberg, 2019; Viik, 2018). Oma töös on Lemberg (2019) välja toonud soovitus, et KEMIT võiks võtta kasutusele Scrum-metoodika. Autor kahtleb, kas ühe konkreetse metoodika kasutuselevõtt ikkagi annab soovitud tulemuse, kuna organisatoorsed või tehnoloogilised eeldused võivad olla täitmata.

KEMIT on avaliku sektori asutus ja avalikus sektoris, nagu ka erasektoris, on töö tellija nägu. Kolp ja Mylopoulos (s.a.) on öeldnud, et tarkvara arhitektuur on üldjoontes sama keeruline kui seda telliva organisatsiooni struktuur. Sama mõtet võib laiendada tarkvara arendusele laiemalt. Nii palju, kui on erinevaid tellijaid, on ka erinevat lähenemist tarkvara tellimisele ning sellega kaasnevale arendusmetoodika sobibusele või ette kirjutamisele. Riigikontrolli (Viira et al., 2019) auditist nähtub, et suuremal või vähemal määral kasutavad kõik avaliku sektori suuremad tellijad mingit agiilset komponenti. Päris ühte kindlat metoodika eelistust ei ole, kuid näiteks Siseministeeriumi Info-tehnoloogiakeskuses (SMIT) on tarkvarateenuste osutamiseks juurutamisel DevOps meetod (AS PricewaterhouseCoopers, 2018).



Avaliku sektori üks olulistest ülesannetest on seaduste ja määruste väljatöötamine ja kehtestamine. Viimatimainitu tõttu on avaliku sektori tarkvaraarendused väga suures osas just mõjutatud erinevate määruste- ja seadustega toimuvatest muudatustest (Viira et al., 2019). Viimati nimetatu on ka Riigikontrolli (ibid.) 2019. aastal avaldatud auditi tulemuste põhjal üheks peamiseks riski- ehk ohuteguriks tarkvaraarenduse hangetes, kuna üheks oluliseks arendusele esitatavate nõuete muutumise põhjuseks on just arendusprotsessi vältel jõustuvad uued määrused ja seadused, mis võivad halvata arendusprotsessi ning juhul, kui on algselt valitud mittesobiv / vale arendusmetoodika, siis võib see kaasa tuua lausa arenduse täieliku ebaõnnestumise ehk mittetöötava või vajadusele mittevastava lahenduse valmimise.

## 2.2. Uurimismetoodika ja valim

Uurimismetoodikad jagunevad kvantitatiivseteks ja kvalitatiivseteks uurimismeetoditeks. Autor on otsustanud kasutada kvalitatiivset uuringut, viies läbi andmete kogumiseks poolstruktureeritud intervjuud. Lisaks kvalitatiivsele uuringule kasutab autor ka kvantitatiivset uuringut ehk küsimustikku, mille eesmärgiks on koguda taustainfot intervjuude läbiviimiseks ja leida seoseid intervjuude ning ankeetküsimustiku vastuste vahel. Uuringu läbiviimise ajakava on toodud tabelis 2. Lisaks eelmainitule soovib autor tuvastada, et kas on ilmselgeid mustreid agiilses arenduses ennast tugevami tundvate ja agiilsusega vähem kokku puutunud vastajate vastustes.

**Tabel 2.** Uuringu läbiviimise ajakava

| Tegevus                            | Läbiviimise aeg         |
|------------------------------------|-------------------------|
| Ankeetküsimustiku edastamine       | 10.03.2021 – 16.03.2021 |
| Ankeetküsimustiku analüüs          | 17.03.2021 – 21.03.2021 |
| Intervjuude läbi viimine           | 21.03.2021 – 29.04.2021 |
| Intervjuude kodeerimine ja analüüs | 30.04.2021 – 10.04.2021 |
| Kokkuvõtete tegemine               | 11.04.2021 – 16.04.2021 |

Allikas: Autori koostatud

Enne intervjuusid edastas autor intervjuueeritavatele kvantitatiivse uuringu küsimustiku, mis on tõlgitud versioon Shastri ja teiste (2021b) poolt koostatud küsimustikust. Autor on lisanud omalt poolt statistilise mõõte saamiseks küsimused vastja agiilsete metoodikatega hakkama saamise ning vastaja arvamuse KEMITi valmisoleku kohta. Intervjuueeritavatele saadetud küsimustk on lisas 1. Shastri jt (2021b) loodud ja uurimistöös kasutatud

originaalküsimustik on toodud lisas 2. Kvalitatiivse uuringu puhul kasutab autor sisendina kvantitatiivse uuringu tulemusi, et piiratud aja jooksul (30 minutit vastaja kohta) saada võimalikult asjakohast informatsiooni ning küsida küsimusi, mis on vastaja vastustega kooskõlas — ehk saab välistada agiilsete meetodikate küsimused kui vastaja ei ole agiilseid meetodikaid kasutanud. Sellist lähenemist on pakkunud oma töös välja ka Shastri ja teised (2021a) ning jõudnud järeldusele, et see töötab. Kvalitatiivse uuringu eesmärgiks on leida täidetud ja täitmata eeldused agiilsete printsiipide rakendamiseks Keskkonnaministeeriumi haldusala projektide juhtimisel, kaardistada riske ja saada sisend protsesside muudatusettepanekuteks.

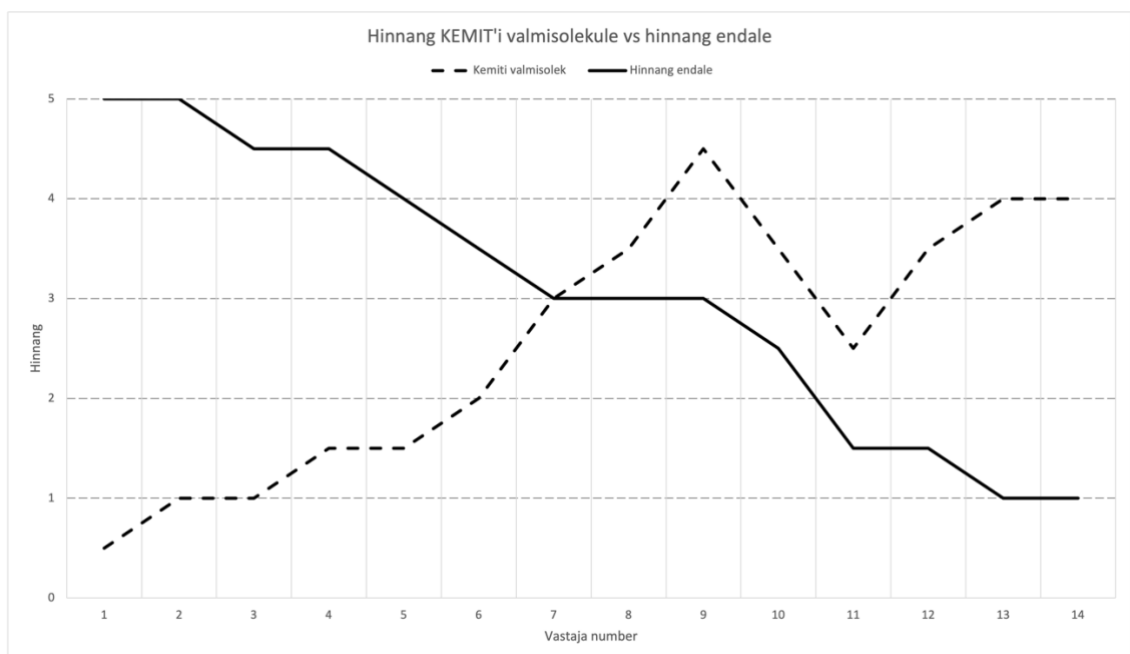
Intervjuud ehk kvalitatiivne uuring on valitud, kuna tavapärast kasutatavat küsitlust on küll hea näiteks küpsustaseme kiireks hindamiseks, kuid organisatoorse eripärade välja toomiseks jääb nendest väheseks (Gren et al., 2015). KEMITis puutub tarkvaraarendusega otseselt kokku 28 inimest. Valimisse on autor võtnud Keskkonnaministeeriumi Infotehnoloogiakeskuse (KEMIT) arendusega kokku puutuvad töötajad, kes on ka läbinud mõne agiilset arendust puudutava koolituse (8 isikut) ja 4 projektijuhti, kes pole saanud agiilse arenduse koolitust. Lisaks kaasatakse kolmest KEMITi poolt (projektide arvu poolest suurimast) teenindatavast asutusest vähemalt üks äripoolte tellija (4 isikut) ja KEMITi-poolsed tarkvara arendusega kokku puutuvad valdkonnajuhid (4 isikut). Valdkonnajuhtidele küsimustikku ei edastata, nendega toimuvad ainult intervjuud. Valimi koostamisel on autor hinnanud võimalikku vastuste kvaliteedi ja mõistliku mahu suhet. Küsimustiku valimis on seega kokku 16 isikut ja intervjuu valimis 20 isikut. Autor ootab küsimustiku täitmise osas minimaalselt 75% suurust osavõttu. Intervjuud viiakse lisaks valdkonnajuhtidele läbi vaid küsimustiku täitnud isikutega.

Kvantitatiivse uuringu läbi viimiseks kasutab autor tasuta veebirakendust Microsoft Forms. Kvalitatiivse uuringu läbiviimisel kasutab autor helisalvestist, mida hoitakse kuni transkriptsiooni valmimiseni. Transkriptsiooni koostamisel kasutatakse algmaterjali tootmiseks Tallinna Tehnikaülikooli poolt loodud täisautomaatset veebipõhist transkribeerimise tarkvara (Tallinna Tehnikaülikooli Küberneetika Instituut, 2018). Autor kasutab kvalitatiivse uuringu kodeerimiseks ning tulemuste leidmiseks tarkvara MAXQDA. Kvantitatiivset uuringut analüüsib autor, kasutades MS Excel andmeanalüüsi funktsioone. Kuna küsitluseks kasutatud veebipõhine töövahend kasutas vastuste

salvestamisel mitte numbrilisi vaid tekstilisi väärtusi, pidi autor enne analüüside alustamist kodeerima laekunud tekstilised vastused enne ringi. Ringi kodeerimine toimus tarkvarapaketi SPSS.

### 2.3. Uuringu tulemused

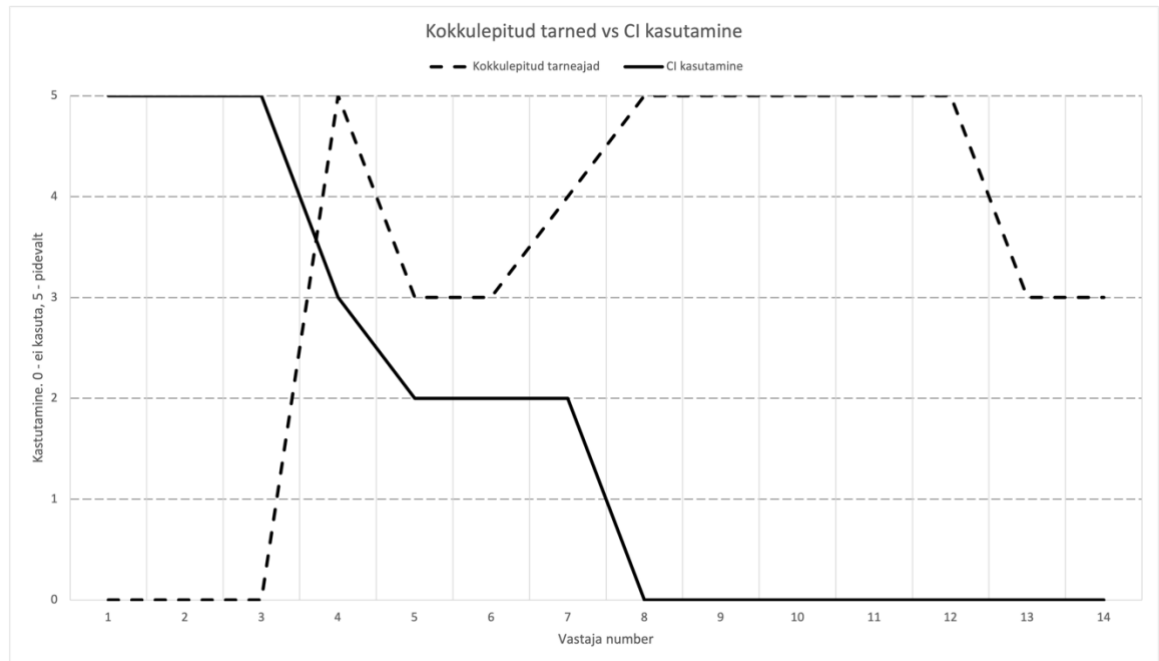
Uuringu küsimustikule vastas kuueteistkümnest valimis olnud isikust neliteist isikut ehk summaarselt 87,5% ning see on autori arvates piisavalt hea tulemus ning peaks õiglaselt kajastama organisatsioonipoolset üldvaadet. Peale küsimustikele vastuste laekumist analüüsis autor neid eesmärgiga leida laekunud vastuste hulgast tugevalt positiivseid ja tugevalt negatiivseid korrelatsioone (lisa 3), et neid konkreetseid vastuseid lähemalt uurida. Väga tugevalt negatiivselt ( $r = -0,824$ ) korreleerusid näiteks Joonis 7 välja toodud hinnang endale agiilsele lähenemise osas ja hinnang KEMITi valmisolekule agiilsete meetodikate kasutuselevõtuks.



**Joonis 7.** Hinnang enda valmisolekule võrreldes arvamusega KEMITi valmisolekust. Autori koostatud

Nagu jooniselt on ka visuaalselt näha, siis mida kõrgema hinnangu vastajad oma agiilsele võimekusele andsid, seda madalamalt nad hindasid KEMITi kui organisatsiooni valmisolekut agiilsete meetodikate kasutuselevõtuks. Sarnase tugevusega tugevalt

negatiivne korrelatsioon ( $r = -0,830$ ) ilmnes ka kindlaksmääratud tarneaegadel toimuvate tarnete ning pideva tarne (CI, *Continuous Integration*) vahel, mida ilmestab Joonis 8 peal nähtav selge erisus.

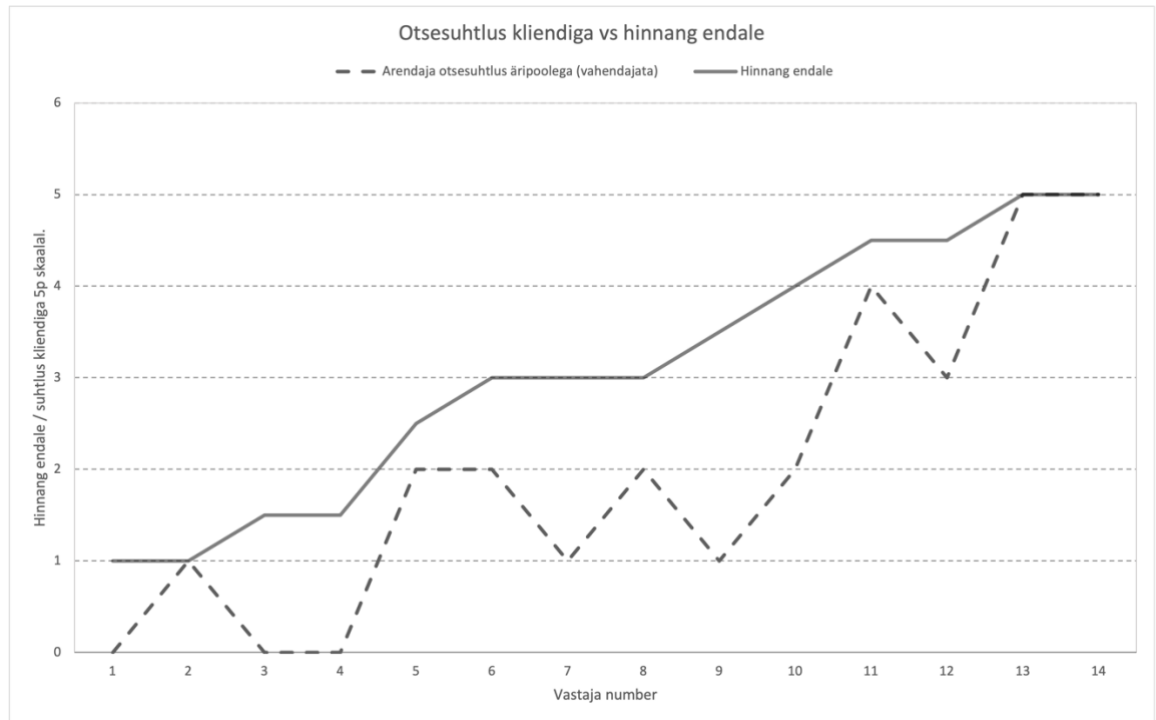


**Joonis 8.** Continuous Integration (CI) kasutamine võrdluses kokkulepitud tarneaegade kasutamisega. Autori koostatud

Joonisel nähtav tulemus oli ka autori poolt oodatav ning igati loogiline, kuna agiilsuse üks tehnoloogilisi eeldusi on just pideva tarne protsessi kasutamine (Fritzsich et al., 2019). Pideva tarne protsess on oluline, et ennast lahti siduda kindlates tarnetsüklistest, mis pärsivad loovust (Nutalapati & Amaradri, 2016), nagu autor ka esimeses peatükis allikatele toetudes väidab.

Lisaks tugevatele negatiivsetele korrelatsioonidele leidis autor ka mõned huvitavad tugevalt positiivsed korrelatsioonid, mis kinnitavad teooria osas toodud printsiipe ja väiteid. Joonis 9 ilmestab hästi, kuidas on väga tugevas positiivses korrelatsioonis ( $r = 0,884$ ) vastajate poolt enda agiilse võimekusele antud hinnang ning nende poolt läbi viidavates projektides arendaja ja kliendi vaheline suhtlus ilma vahendajata. Arendusmeeskonna ja kliendi vaheline otse ehk ilma vahendajata suhtlus on äärmiselt oluline, et kogu arendusmeeskond, kes kliendi heaks töötab, saaks esitada oma küsimused ning ettepanekud ja saaks neile kohese ning adekvaatse tagasiside otseallikast ehk

kliendilt. Nii üles ehitatud suhtlusprotsessi ajal on kliendil võimalik veenduda, et arendusmeeskond saab tema poolt esitatud soovidest aru sama moodi nagu need kliendi ettekujutustes on.



**Joonis 9.** Arendaja poolne otsesuhtlus kliendiga ja intervjuueeritavate hinnang endale. Autori koostatud

Sellist olukorda saab seletada autori arvates lihtsalt, et mida kindlamini inimesed ennast tunnevad agiilsete meetoditega, seda enam on nad valmis niiõelda käest andma ja meeskonda usaldama ilma, et peaksid osapoolte vahel filtriks olema. Seda toetab ka tugev positiivne seos enese agiilse võimekuse hinnangu ning nii kogu meeskonna otsuste ( $r = 0,780$ ) kui ka agiilsete meetodikate üldise toimimise ( $r = 0,798$ ) hinnangute vahel.

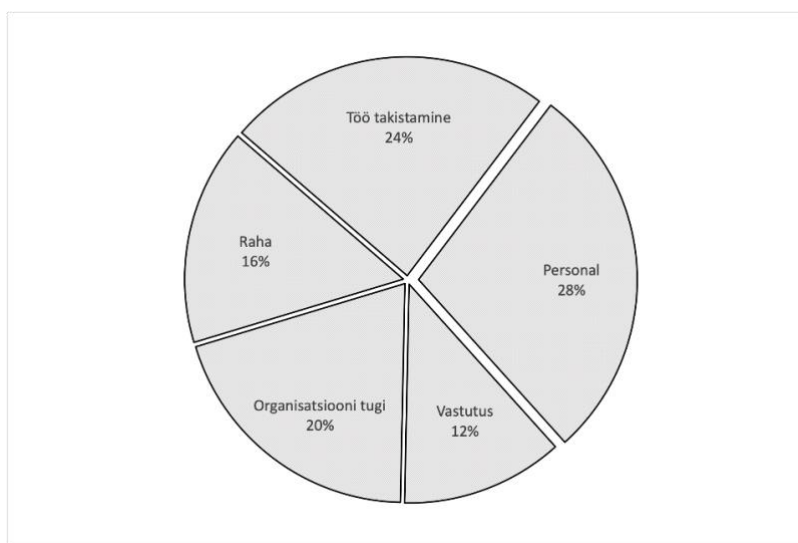
Huvitava nüansina tuli andmeid analüüsidest välja, et hoolimata infost, et Scrum-metoodika puhul on tseremooniad äärmiselt olulised ning kindlaks määratud (Ambler & Lines, 2020; Khmelevsky et al., 2017) neid siiski ei järgita ehk pigem kasutatakse hübriidlahendusi mitte Scrumi.

Küsimustike alusel antud vastused olid aluseks poolstruktureeritud intervjuudele, mille autor viis läbi seekord eranditult elektroonilisi kanaleid pidi. Intervjuudes esitas autor järgmised küsimused:

- Millised tehnoloogilised piirangud takistavad sinu hinnangul agiilsete meetodikate kasutuselevõttu?
- Millised organisatsioonilised piirangud takistavad sinu hinnangul agiilsete meetodikate kasutuselevõttu?
- Milliseid riske näed seoses agiilsetele printsiipidele üleminekuga?

Peale intervjuude kokkuvõtete kodeerimist joonistus välja, et tehnoloogiliste takistustena nähakse vaid kahte takistust – avalike pilveteenuste kasutamiskeeldu ning automatiseerituse väga madalat taset, kus vajalikud tööd teeb keegi manuaalselt. Käsitöö võtab põhjendatult aega ning töövälisel ajal ei ole mõningaid asju üldse võimalik saada. Arvestades arenduspartnerite muutunud töökorraldust, on arendusmeeskonnad tihti pillutatud üle maailma laiali, mis tähendab, et teatud tööde (näiteks andmebaas, nimekirje) tellimiseks oleks vaja iseteenindusportaali, et välistada olukord, kus arendus peab seisma terve nädalavahetuse.

Organisatoorseid piiranguid toodi tehnoloogilistest oluliselt rohkem välja. Autor grupeeris kodeeritud märksõnad / märgendid kokku ning moodustas neist teemad. Joonis 10 ilmestab kuidas märksõnad teemade vahel jagunesid.



**Joonis 10.** Märksõnade jagunemine teemade vahel. Autori koostatud

Nagu jooniselt Joonis 10 nähtub on suurima (28% märksõnadest) välja toodud organisatoorse takistusena personaliga seotud takistused. Tabel 3 sisaldab endas autori poolt teemadeks grupeeritud märksõnade loendit, kus iga teema juurde on märgitud olulisemad kõlanud märksõnad.

**Tabel 3.** Organisatsiooniliste piirangute jagunemine

| Teema                      | Märksõnad  |
|----------------------------|--|
| Vastutus                   | otsustuspunkt meeskonnast kõrgemal meeskonnal puudub õigus otsustada vastutuse jagamine alla puudulik  |
| Personal                   | tahtmatus muutuda lahenduste mitte välja pakkumine puuduvad inimesed ei saa - suhtumine ei hinnata inimeste individuaalset panust vanas kinni hoidmine puudub tahtmine otsustada           |
| Raha                       | hankimine rahastusmudel ei pruugi toetada meeskondade motivatsioonipakett tiimi eelarveid ei ole   |
| Töö takistamine            | puudub huvi/tugi taristu poolt kõik asjad võtavad aega kuna on käsitöö juurdepääsude saamine tülikas detailanalüüsis kinni olemine muudatuste pidev edasilükkamine puudub julgus otsustada |
| Organisatsioonipoolne tugi | motivatsiooniskeemid kokkuhoiu väärtustamine klientidele „müügitöö“ tegemine palka aeglaselt, lase lahti kiiresti ehk õigete inimeste palkamine  |

Allikas: autori koostatud

Tabeli 3 aluseks oleva informatsiooni on autor otsustanud kokkuvõtvalt ka lahti kirjutada, et oleks kergem teemat ja tabelis olevat märksõnaloendit kokku viia. Autor on tabelis toodud märksõnad järgnevatel lõikudes esitanud rõhutatud teksina.

**Vastutus** on teema, mis kõnetas kõiki intervjueeritavaid. Selle teema alla koondus nii probleeme kui ettepanekuid. Suuremateks probleemideks oli otsustusõiguse puudumine eelarvevahendite kasutamise osas ning vastutuse ja otsustuspunkti asumine kõrgemal kui meeskond ehk siis meeskond on kohati sundseisus. Sundseisus nähakse ennast nii tööde valimisel kui ka näiteks kasvõi otsustamisel, et millise teenusepakkuja käest mingit

teenust tellida. Ettepanekutena toodi enim välja, et meeskonnad peaksid saama suurema autonoomsuse ja vastutuse oma eelarve kasutamise osas ning suurema võimaluse valida sisse tulevaid töid, kui ei ole piisavalt kas eelarvelisi vahendeid või personali.

**Raha** teemas toodi esile eelneva vastutuse punktiga seotult eelarvete meeskondadele alla viimine, motivatsioonipaketi koostamine ning suure teemana hankimine. Klassikaline riigihange on paraku sobilikum toodete ostmiseks, aga mitte niivõrd teenuste, seda enam agiilse arendusteenuse, ostmiseks. Kuidas ja kas eelarve struktuur ja -poliitika meeskonnamudelit toetama hakkab, oli läbiv küsimus. Olukorda ei tee ka lihtsamaks fakt, et KEMITi rahakoti üle otsustab ministeerium ehk tegemist ei ole klassikalise isemajandava organisatsiooniga, kus finantside juurde saamiseks on võimalik kas loobuda mõningate teenuste pakkumisest või ennast ringi profileerida kasumlikumate teenuste pakkumise peale.

Teravalt puudutati **personali** teemat. Vastajad töid välja, et organisatsioonis on palju nn ei-suhtumist, kus öeldakse „ei“ ning ei pakuta välja lahendusi. Kas probleemi / väljakutset ei soovitagi lahendada või ei juleta vastu võtta olemasolevat olukorda muutvat otsust, on teadmata. Selline suhtumine tekitab olukorra, kus inimesed jäävad vanasse sisseharjunud rutiini kinni ja nendega kokku puutuvatel kinnistub arusaam, et neil puudub tahtmine muutuda – agiilse arenduse puhul on tegu tõelise probleemiga, et mitte öelda ületamatu takistusega. Lisaks toodi välja, et inimesi koheldakse küll võrdselt, kuid see tekitab mõningatel juhtudel tunde, et inimese reaalsel / individuaalset panust ei hinnata piisavalt. Lisaks eelnevale kerkis üles korduvalt ka inimeste puuduse probleem, leiti, et võibolla ei ole isegi mitte probleem selles, et inimesi on puudu, vaid mõne inimese roll ei ole talle sobilik. Kõik loodud tiimid ei kata ära kogu vajalikku oskusteavet ehk on alamehitatud – alamehitatud meeskonnad on kindlasti probleem, millega tuleb tegeleda.

**Töö takistamine** oli autorile mõnevõrra üllatuslikult üles kerkinud märksõnadest koostatud teema. Töö takistamise all ei mõeldud alati mitte seda, et keegi konkreetne inimene sihilikult takistab töö tegemist, vaid pigem seda, et kas kliendid on valmis arvestama sellega, et ka nende poolne panus meeskonnaliikmena peab suurenema. Lisaks nägid agiilseid metodikaid viljelevad projektijuhid suurt organisatsioonilist murekohta klientide seas, kes on pikka aega sama rakendust kasutanud ning uue rakenduse loomisel ei panustata piisavalt oma tööprotsessi muutmisesse ning agiilse arenduse eelduseks



olevasse pidevasse suhtlemisse. Kuna enim vastajaid oli tarkvaraarendusega kokku puutuvate inimeste hulgast, siis arvestades klassikalist eraldatud anumate printsiipi (IT-terminoloogias keeles silode maailma), siis oli ette arvata, et etteheited tuleb ka taristule – leitakse, et seoses käsitöö tegemisega on oht, et asjad venivad, sealt poolt ei ole piisavat huvi, et asju parendada ning pigem ollakse vanas kinni, mis taaskord tekitab riskiolukorra.

Viimasena, aga kindlasti mitte vähemolulisena, grupeeris autor märksõnad teemaks **organisatsioonipoolne tugi**. Enim leidsid organisatsioonipoolse toe osas märkimist klientidele KEMITi uute arengusuundade levitamise, klientide tippjuhtkonnast ei ole see alati otseselt projektidega seotud inimesteni jõudnud. Agiilse arenduse puhul oodatakse, et üle vaadatakse ja lepitakse kokku ka uued motivatsiooniskeemid, mis võtaksid ühtlasi arvesse inimeste panust ning meeskondade tulemusi rahalise kokkuhoiu saavutamisel. Ressursipuuduses olevale organisatsioonile kohaselt tuli mitu korda välja üleskutse, et palgata tuleks aeglaselt ja kindlasti just neid inimesi, keda on kõige rohkem vaja.

Lisaks tehnoloogilistele- ja organisatsioonilistele piirangutele uuris autor ka riske, mida nähakse agiilsele arendusele üleminekul. Läbiviidud uuringus kaardistatud riskid grupeeris autor nelja märksõna alla:

- personaliriskid,
- raha ehk finantsidega seotud riskid,
- mõõdikutega seotud riskid,
- klientidega seotud riskid.

**Raha** ehk finantsidega seotud riski puhul leiti, et eelarve struktuur ja -poliitika ei pruugi kattuda soovitava mudeliga, kus eelarve juhtimine oleks meeskondade kätte viidud. Riskiks on hankemenetluse käigus saadud partneri kompetentsid, tema soov ja võime agiilselt arendada. Kuna suur osa projekte on rahastatud eurorahast, siis toodi välja risk, et struktuurfondi (SF) vahenditest arenduste tellimine pärsib agiilse arenduse.

**Mõõdikute riski** all peetakse silmas, et agiilse arenduse puhul ei ole tulemuse mõõdikud üheselt paika pandud ja mõõdetavad. Selle riski välja toomine on autori arvates huvitav, kuna ka hetkel on põhimõtteliselt ainsaks mõõdikuks see, et kas sinu töö on tehtud vastavalt kokkuleppele ehk kas projekt on saanud valmis kokkulepitud ajaks, kokkulepitud eelarvega ja kokkulepitud funktsionaalsuses. Lisanduvalt toodi välja risk,

et ebaõigete mõõdikute puhul soodustakse mõõdikute järgi töötamist, eriti kui see ei ole üheselt seotud motivatsioonipaketiga.

**Personaliriskidena** toodi välja:

- senise töökorralduse muudatus tekitab inimestes segadust ja sellega seotult rahulolematust,
- kardetakse muudatusi ning seetõttu ei taha inimesed ennast muuta,
- meeskondades puuduvad vajalike oskustega inimesed,
- agiilsete meetodikatega end paremini tundvad inimesed on töökoormusega viidud murdumise äärele,
- meeskondade motivatsioonipakett võib soodustada „ei“ ütlemist lahenduste pakkumise asemel või tekitada olukorra, kus töö tegemise asemel hakatakse töö käima – keegi meeskonnast ikka ära teeb,
- tiimid on alakomplekteeritud ehk ei suuda täita neile pandud ülesandeid täies oodatud mahus,
- rahulolematuse tõttu uue töökorraldusega lahkuvate inimeste arvelt kasvab koormus,
- kliendi meeskonnast lahkuvad inimesed ning neid asendatakse ebapiisava oskusteabega spetsialistidega.

**Kliendiriskiks** on klientide võimalik mitte valmisolek uue ja nende jaoks harjumatu arendusprotsessiga kaasa tulla. Reaalseks hinnati vastajate poolt riski, et kliendi poolt ei ole võimalik panustada projektimeeskonda piisaval hulgal aega, mida agiilne arendus eeldab. On selge, et klientide esindajate koormus kasvab senisega võrreldes oluliselt ning ka nendelt oodatav panus sellega koos.

## **2.4. Järeldused ja ettepanekud**

Uuringu tulemustest on võimalik järeldada, et inimeste valmisolek agiilsete meetodikate kasutuselevõtuks on väga erinev. Need vastanud, kes juba kasutavad agiilseid meetodikaid tunnevad, et teised „lohisevad kaasa“ ning ei ole piisavalt motiveeritud enese muutmiseks. Need, kes veel ei kasuta, on samas piisavalt positiivselt meelestatud agiilse arenduse osas.

Huvitava leiuna oli agiilseid metoodikates ennast pigem tugevatena tundvatel projektijuhtidel korruga käimas olevate projektide arv oluliselt (keskmiselt isegi 3 korda) suurem, kui traditsiooniliste meetodite abil projekte juhtinud projektijuhtidel. Oma olulisemad tähelepanekud ja nende alusel tehtud järeldused ning ettepanekud on autor toonud välja ka tabeli kujul (Tabel 4) parema ülevaate saamiseks.

**Tabel 4.** Tähelepanekud, järeldused ja ettepanekud

| Tähelepanek  | Järeldus   | Ettepanek   |
|--|--|---|
| Ebäühtlane automaatpaigalduste kasutamise tase projektides | Parimate praktikate kasutuselevõtt ei ole olnud prioriteet   | Seada automaatpaigaldus normiks, tutvustada kõigile töötajatele selle kasutegurit ning võtta maha hirmud, et automaatika kaotab töökohti              |
| Mure töötajate võimekuse üle                               | Vastajad tunnevad muret organisatsiooni käekäigu pärast  | Korraldada ühisüritusi ning agiilsete metoodikate tutvustamist ning pakkuda agiilse juhendaja teenust näiteks esimese projekti juures                 |
| Soov võtta suurem vastutus                                 | Mingi osa organisatsioonist on valmis võtma omale suuremat vastutust kui seni, samas on piisavalt ka neid keda senine olukord rahuldab | Proovida võimalusel seda ühe meeskonna peal, agiilsed meeskonnad peaksidki idee poolest olema ise hakkama saavad                                      |
| Hankimine keeruline  | Riigihanked, sellega kaasnev juriidika ning bürokraatia on tehnilise taustaga inimestele keeruline ja võtab palju energiat ning aega   | Pakkuda hangete korraldamise teenust keskselt, et säästa tehniliste inimeste aega. Võimalusel võiks selline teenus olla üle haldusalade.              |
| Suhtumine „see on teise valdkonna töö“                     | Läheb veel aega kuni organisatsiooni sulandub üheks uue mudeli järgi töötavaks tervikuks   | Soodustada meeskondade ja valdkondadevahelisi projekte, tunnustada „silode“ lõhkumist. Korraldada ühiseid strateegianõupidamisi ja meeskonnakoolitusi |
| Arendusprotsessis ollakse kinni                            | Agiilne mõttelaad, tulemustele orienteeritus ning soov ja tahtmine teha asju ära on puudulik   | Koolituste korraldamine, arendus- ja tarneprotsessi muudatuste sisse viimine.   |
| Projektide arv 5 ja enam                                   | Agiilseid metoodikaid kasutavate projektijuhtide poolt juhitud projektide arv on ebamõislikult suur, et pakkuda piisavat kvaliteeti    | Võimalusel piirata ühe inimese projektide aru kahele.   |

Allikas: autori koostatud

Oluliseimad autori ettepanekud puudutavad tarkvara arendus- ja tarneprotsessi. Tarneprotsessi tuleb lisada automaatpaigaldus (CI), tööde vastuvõtukriteeriumiks peab saama tarne automaatne paigaldus ning tarkvara arendusprotsessi tuleb täiendada agiilse arenduse kasutamist soosivate osadega.

Tuginedes läbi viidud küsitlusele ning intervjuudele on autori seisukoht, et KEMIT organisatsioonina ei ole veel lõpuni valmis üleni agiilsete meetodikate kasutuselevõtuks. Eeldused on selleks küll olemas ning KEMITis töötavatel inimestel on selge huvi olemas. Suurimaks väljakutseks on personali teadlikkuse tõstmine. Intervjuude käigus ilmnes, et agiilsete arendusmeetodikate osas valitseb töötajate hulgas paljuski lisaks teadmatusele ka väärarusaamu ning põhjendamatuid hirme.

## KOKKUVÕTE

Agiilsed arendusmetoodikad on teema, millest me kuuleme pea iga tarkvaraarenduse projekti juures. Agiilsusest on saanud kahjuks peaaegu parasiitsõna nagu on selleks juba märksõnad Scrum ja DevOps – nende sõnade tähenduste kohta levib samapalju erinevaid tõlgendusi nagu on organisatsioonid, kes neid sõnu oma teenuste tutvustamisel kasutada tavatsevad.

Käesoleva töö eesmärk oli anda ülevaade agiilsetest tarkvaraarenduse protsessidest ja uurida Keskkonnaministeeriumi haldusala tarkvaraarendusprotsessides agiilsete metoodikate kasutuselevõtu võimalusi ning selgitada välja, milline agiilne metoodika oleks Keskkonnaministeeriumi infotehnoloogiakeskuses (KEMIT) kasutuselevõtuks sobivaim. Autori hinnangul said uurimisküsimused vastatud ning sellega seotud eesmärk täidetud.

Kirjandusest ülevaate koostamisel ilmnas, et agiilsete metoodikate kasutuselevõtuks tuleb organisatsioonidel ületada nii tehnoloogilised kui ka organisatoorsed takistused. Täpselt samale tulemusele jõudis autor viies läbi ankeetküsimustikule järgnenud poolstruktureeritud intervjuusid. Intervjuudest nähtus, et mida pädevamaks pidasid ennast intervjuueeritavad agiilsete printsiipide osas, seda madalamaks nad hindasid organisatsiooni kui terviku valmisolekut. Intervjuude vastustest tuli välja, et mitmed vastajad oleks valmis võtma senisest oluliselt suuremat vastutust.

Autor on esitanud kokkuvõtva informatsiooni ning kaardistanud ja välja toonud tähelepanekute alusel riskid, millega arvestada agiilsete metoodikate juurutamisel. Esitatud on autoripoolsed ettepanekud tarkvara arendus- ja tarneprotsessi täiendamiseks. Töö tulemusena valminud kaardistus ning riskide ja võimaluste loetelu on väärtuslik materjal, mida KEMIT saab kasutada agiilsema tarkvaraarenduse protsessi poole liikumisel. Autori hinnangul ei ole organisatsioon tervikuna veel valmis ainult agiilsele tarkvaraarendusmetoodikale üleminekuks. Küll aga leiab autor, et kui juhtkonnapoolne

otsus tarkvaraarenduses agiilsete metoodikate poole liikuma hakata püsib, siis distsiplineeritud DevOps võiks olla see raamistik, mille alusel edasi liikuda.

Töö maht ei võimaldanud tegevusplaanide väljatöötamist ning sellega seoses pakub autor järeltegevusena välja, et teemaga võiks minna edasi. Järeltegevusena tuleks kaardistada ja analüüsida KEMITi töötajate agiilse arendusmetoodika jaoks vajalikud tööalased oskused ja teadmised, et teha selle alusel juba organisatsioonile konkreetne plaan, kuidas ja millises mahus hakata juurutama distsiplineeritud DevOps-i. Nagu Desmond Tutu on öelnud, siis „ainus viis elevandi söömiseks on seda süüa suutäie kaupa“. Nii soovitab ka autor liikuda agiilsuse poole agiilselt ehk väikeste sammudena.

## VIIDATUD ALLIKAD

- Adamson, A. (2020). *Infosüsteemi arendamise põhimõtted muutuvate nõuete korral Eesti avalikus sektoris* [Magistritöö]. Tartu Ülikool.
- Agile Business Consortium. (2014). *The DSDM Agile Project Framework*.  
<https://www.agilebusiness.org/page/TheDSDMAgileProjectFramework>
- Alsanad, A. A., Chikh, A., & Mirza, A. (2019). A Domain Ontology for Software Requirements Change Management in Global Software Development Environment. *IEEE Access*, 7, 49352–49361. <https://doi.org/10.1109/ACCESS.2019.2909839>
- Ambler, S. W., & Lines, M. (2019). *Choose your WoW! a disciplined agile delivery handbook for optimizing your way of working*. Disciplined Agile Consortium.
- Ambler, S. W., & Lines, M. (2020). *Introduction to Disciplined Agile Delivery Second Edition: A Small Agile Team's Journey from Scrum to Disciplined DevOps*. Project Management Institute.
- Amoroso, E. (2018). Recent Progress in Software Security. *IEEE Software*, 35(2), 11–13. <https://doi.org/10.1109/MS.2018.1661316>
- Arula, M. (2020). *IT korralduse raamistik KEM valitsemisalas*. Keskkonnaministeeriumi Infotehnoloogiakeskus.
- AS PricewaterhouseCoopers. (2018). *SMIT IKT teenuste finantseerimise jätkusuutlikkuse analüüs*. AS PricewaterhouseCoopers.  
<https://www.smit.ee/files/lopparuande-esitlus-17.08.pdf>
- Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70–77. <https://doi.org/10.1109/2.796139>
- Beck, K., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D., & Fowler, M. (2001). *Manifesto for Agile Software Development*. Manifesto for Agile Software Development. <https://agilemanifesto.org/>

- Bell, J. T. (2017). *Extreme programming*. CS 442, Department of Computer Science University of Illinois, Chicago.
- Bhuiyan, N., & Baghel, A. (2005). An overview of continuous improvement: From the past to the present. *Management Decision*, 43(5), 761–771.  
<https://doi.org/10.1108/00251740510597761>
- Boutin, C. (2020, april 3). *Beginner's Guide to the Waterfall Model for Project Managers*. Unito. <https://unito.io/blog/waterfall-model-project-management-methodology/>
- Butler, C. W., Vijayarathy, L. R., & Roberts, N. (2019). Managing Software Development Projects for Success: Aligning Plan- and Agility-Based Approaches to Project Complexity and Project Dynamism: *Project Management Journal*.  
<https://doi.org/10.1177/8756972819848251>
- Cohn, M., Sim, S., & Lee, C. (2009). What Counts as Software Process? Negotiating the Boundary of Software Work Through Artifacts and Conversation. *Computer Supported Cooperative Work*, 18, 401–443. <https://doi.org/10.1007/s10606-009-9100-4>
- Cope, R. (2020). Strong security starts with software development. *Network Security*, 2020(7), 6–9. [https://doi.org/10.1016/S1353-4858\(20\)30078-7](https://doi.org/10.1016/S1353-4858(20)30078-7)
- Cybernetica AS. (2020). *AKIT - Andmekaitse ja infoturbe leksikon*. <https://akit.cyber.ee/>
- Dalla, L. O. F. B. (2020). Lean Software Development practices. *International Journal of Engineering and Modern Technology*, 6(1).
- Digital.ai Software Inc. (2020). *14th Annual State of Agile Report*. Digital.ai Software Incorporated. <https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494>
- Du, W. (Derek), Wu, J., Liu, S., & Hackney, R. A. (2019). Effective organizational improvisation in information systems development: Insights from the Tencent messaging system development. *Information & Management*, 56(4), 614–624.  
<https://doi.org/10.1016/j.im.2018.10.003>
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9), 833–859.  
<https://doi.org/10.1016/j.infsof.2008.01.006>



- Fritsch, J., Bogner, J., Zimmermann, A., & Wagner, S. (2019). From Monolith to Microservices: A Classification of Refactoring Approaches. J.-M. Bruel, M. Mazzara, & B. Meyer (Toim), *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment* (Kd 11350, lk 128–141). Springer International Publishing. [https://doi.org/10.1007/978-3-030-06019-0\\_10](https://doi.org/10.1007/978-3-030-06019-0_10)
- Gemino, A., Horner Reich, B., & Serrador, P. M. (2021). Agile, Traditional, and Hybrid Approaches to Project Success: Is Hybrid a Poor Second Choice? *Project Management Journal*, 52(2), 161–175. <https://doi.org/10.1177/8756972820973082>
- Gil, D. G., & Díaz-Heredero, R. A. (2018). A microservices experience in the banking industry. *Proceedings of the 12th European Conference on Software Architecture Companion Proceedings - ECSA '18*, 1–2. <https://doi.org/10.1145/3241403.3241418>
- Gren, L., Torkar, R., & Feldt, R. (2015). The prospects of a quantitative measurement of agility: A validation study on an agile maturity model. *Journal of Systems and Software*, 107, 38–49. <https://doi.org/10.1016/j.jss.2015.05.008>
- Griffin, L. (2021). Implementing Lean Principles in Scrum to Adapt to Remote Work in a Covid-19 Impacted Software Team. A. Przybyłek, J. Miler, A. Poth, & A. Riel (Toim), *Lean and Agile Software Development* (lk 177–184). Springer International Publishing.
- Hamunen, J. (2016). *Challenges in adopting a Devops approach to software development and operations* [Master's Thesis, Aalto University]. <https://aaltodoc.aalto.fi/handle/123456789/20766>
- Himmat, M., & Osman, A. S. A. (2020). Agile software development methods and current trends. *Journal of Software Engineering & Intelligent Systems*, 5, 6.
- Jevtic, G. (2019, mai 15). What is SDLC? How the Software Development Life Cycle Works. *PhoenixNAP Blog*. <https://phoenixnap.com/blog/software-development-life-cycle>
- Jonnalagadda, G., Shafey, S., Lynah, W., Massetti, M., Maerten, P., Wiczorek, S., Landzaat, S., Probst, M., Schuster, A., Oxborough, C., Bains, R., & Bonser, M. (2017). *Agile Project Delivery Confidence. Mitigate project risks and deliver value to your business*. PricewaterhouseCoopers International Limited.

- KeMIT. (2021). *Keskkonnaministeeriumi Infotehnoloogiakeskus*.  
 Keskkonnaministeeriumi Infotehnoloogiakeskus. <https://www.kemit.ee/>  
*Keskkonnaministeeriumi Infotehnoloogiakeskuse põhimäärus*. (2020).  
 Keskkonnaminister. <https://www.riigiteataja.ee/akt/124072020004?leiaKehtiv>
- Khmelevsky, Y., Li, X., & Madnick, S. (2017). Software development using agile and scrum in distributed teams. *2017 Annual IEEE International Systems Conference (SysCon)*, 1–4. <https://doi.org/10.1109/SYSCON.2017.7934766>
- Kolp, M., & Mylopoulos, J. (s.a.). *Software Architectures as Organizational Structures*. 6.
- Kuura, A., & Sandoval, I. (2019). Improvisation in Project Management: Lessons from Jazz. *Project Management Development – Practice and Perspectives*, 15–28.
- Lemberg, V. (2019). *Agiilse lähenemise rakendusvõimaluste uurimine Keskkonnaministeeriumi haldusala tarkvaraprotsessis* [Magistritöö]. Tallinna Tehnikaülikool.
- Mazlami, G., Cito, J., & Leitner, P. (2017). Extraction of Microservices from Monolithic Software Architectures. *2017 IEEE International Conference on Web Services (ICWS)*, 524–531. <https://doi.org/10.1109/ICWS.2017.61>
- Morales, J. A., Yasar, H., & Volkman, A. (2018). Implementing DevOps practices in highly regulated environments. *Proceedings of the 19th International Conference on Agile Software Development Companion - XP '18*, 1–9.  
<https://doi.org/10.1145/3234152.3234188>
- Niever, M., Trinh, H. J., Kerres, R., & Hahn, C. (2020). Integration of Agile Approaches in SME's Product Development: Demand Analysis and Concept Development. *Athens: ATINER'S Conference Paper Series, No: SME2020-2717*, 19.
- Nuottila, J., Aaltonen, K., & Kujala, J. (2016). Challenges of adopting agile methods in a public organization. *IJISPM - International Journal of Information Systems and Project Management*, 3, 65–85. <https://doi.org/10.12821/ijispm040304>
- Nurdiani, I., Börstler, J., Fricker, S., Petersen, K., & Chatzipetrou, P. (2019). Understanding the order of agile practice introduction: Comparing agile maturity models and practitioners' experience. *Journal of Systems and Software*, 156, 1–20.  
<https://doi.org/10.1016/j.jss.2019.05.035>

- Nutalapati, S. B., & Amaradri, A. S. (2016). *Continuous Integration, Deployment and Testing in DevOps Environment* [Master's Thesis, Blekinge Institute of Technology]. <http://urn.kb.se/resolve?urn=urn:nbn:se:bth-13334>
- Olagunju, A. O. (2018). Revamping the IT Curriculum with Agile and DevOps Methodology. *Proceedings of the 19th Annual SIG Conference on Information Technology Education - SIGITE '18*, 86–86. <https://doi.org/10.1145/3241815.3241818>
- Paasivaara, M. (2021). Teaching the Scrum Master Role using Professional Agile Coaches and Communities of Practice. *IEEE/ACM 43rd International Conference on Software Engineering: Joint Track on Software Engineering Education and Training (ICSE-JSEET)*, 10.
- Paule, C., Dullmann, T. F., & Van Hoorn, A. (2019). Vulnerabilities in Continuous Delivery Pipelines? A Case Study. *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 102–108. <https://doi.org/10.1109/ICSA-C.2019.00026>
- Porras, A. (2021, april). *DevOps For Business*. 4Geeks Blog. <https://blog.4geeks.io/devops-for-business/>
- Project Management Institute. (s.a.). *Foundation for Business Agility | Disciplined Agile*. Salvestatud 25. veebruar 2021, <https://www.pmi.org/disciplined-agile>
- Project Management Institute. (2017a). *A guide to the project management body of knowledge (PMBOK guide)* (Sixth edition.). Project Management Institute.
- Project Management Institute. (2017b). *The Agile practice guide*. The Project Management Institute.
- Project Management Institute. (2019). *PMI Pulse of the Profession 2019*. <https://www.pmi.org/learning/thought-leadership/pulse/pulse-of-the-profession-2019>
- Project Management Institute. (2020). *PMI Pulse of the Profession 2020*. <https://www.pmi.org/learning/thought-leadership/pulse/pulse-of-the-profession-2019>
- Rajkumar, M., Pole, A. K., Adige, V. S., & Mahanta, P. (2016). DevOps culture and its impact on cloud delivery and software development. *2016 International Conference*

- on *Advances in Computing, Communication, & Automation (ICACCA) (Spring)*, 1–6. <https://doi.org/10.1109/ICACCA.2016.7578902>
- Rush, L. (2016). *DevOps printsiipide rakendamise analüüs TransferWise'i näitel*. [Tallinna Tehnikaülikool]. <https://digi.lib.ttu.ee/i/?5126>
- Saltz, J., Anderson, E., & Sutherland, A. (2020). *Agile and Lean: Organizations, Products, and Development*. Hawaii International Conference on System Sciences.
- Serrador, P., & Pinto, J. K. (2015). Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051. <https://doi.org/10.1016/j.ijproman.2015.01.006>
- Shastri, Y., Hoda, R., & Amor, R. (2021a). The role of the project manager in agile software development projects. *Journal of Systems and Software*, 173, 110871. <https://doi.org/10.1016/j.jss.2020.110871>
- Shastri, Y., Hoda, R., & Amor, R. (2021b). *The role of the project manager in agile software development projects. Pre-interview survey questionnaire*. <https://doi.org/10.7910/DVN/QEASRH>
- Sohaib, O., Solanki, H., Dhaliwa, N., Hussain, W., & Asif, M. (2019). Integrating design thinking into extreme programming. *Journal of Ambient Intelligence and Humanized Computing*, 8. <https://doi.org/10.1007/s12652-018-0932-y>
- Sommerville, I. (2016). *Software engineering* (Tenth edition). Pearson.
- Špundak, M. (2014). Mixed Agile/Traditional Project Management Methodology – Reality or Illusion? *Procedia - Social and Behavioral Sciences*, 119, 939–948. <https://doi.org/10.1016/j.sbspro.2014.03.105>
- Tallinna Tehnikaülikooli Küberneetika Instituut. (2018). *Veebipõhine transkriptsioon*. <http://bark.phon.ioc.ee/webtrans/>
- Thompson, K., & Williams, N. (2021, jaanuar). *2021 is International Year of Responsible Project Management*. <https://www.ipma.world/2021-is-international-year-of-responsible-project-management/>
- Totah, Z. (2021, jaanuar 6). *Project Management Trends 2021 | Future Of Project Management*. <https://www.selecthub.com/project-management/project-management-trends/>
- Triebert, S. (2005). *SWARM: A Concluding Paper, Software Architecture Vs. Building Architecture*. 7.

- Vene, S. (2017). *DEVOPS juurutamine suuretevõttes* [Magistritöö]. Tallinna Tehnikaülikool.
- Vestues, K., Mikalsen, M., & Monteiro, E. (2021). *Using digital platforms to promote a service-oriented logic in public sector organizations: A case study*. Hawaii International Conference on System Sciences.  
<https://doi.org/10.24251/HICSS.2021.269>
- Viik, K. (2018). *KEMIT tarkvara arendusprotsess*. Keskkonnaministeeriumi Infotehnoloogiakeskus.
- Viira, T., Lääne, A., Saksing, M., Lazartšuk, J., & Nõuakas, A. (2019). *Avaliku sektori tarkvaraarenduse projektide juhtimine*. (lk 63) [Riigikontrolli aruanne Riigikogule]. Riigikontroll.
- What Is DevOps? (s.a.). *The Agile Admin*. <https://theagileadmin.com/what-is-devops/>
- Wiedemann, A., Forsgren, N., Wiesche, M., Gewalt, H., & Krömer, H. (2019). Research for practice: The DevOps phenomenon. *Communications of the ACM*, 62(8), 44–49. <https://doi.org/10.1145/3331138>
- Yuan, E. (2019). Architecture Interoperability and Repeatability with Microservices: An Industry Perspective. *2019 IEEE/ACM 2nd International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE)*, 26–33. <https://doi.org/10.1109/ECASE.2019.00013>

## Lisa 1. Küsimustik

### Osa A - üldinformatsioon

1. Sinu nimi \*

2. Millises vahemikus on sinu vanus \*

- 20-29
- 30-39
- 40-49
- 50-60
- üle 60

3. Sinu roll / ametinimetus \*

4. Kui pikk on sinu töökogemus/kokkupuude tarkvaraarendusega? (aastad) \*

- 0-5
- 6-10
- 11-20
- >20

5. Kui pikk on sinu töökogemus praeguse tööandja juures? \*

- 0-2
- 3-4
- 5-7
- >7

6. Mitmes waterfall (traditsioonilises projektis) oled seni osalenud? \*

- 0
- 1-5
- 6-10
- 11-20
- >20

7. Milline oli waterfall (traditsiooniliste projektide) keskmine kestvus? \*

- 6
- 12
- 24
- 36

8. Mitmes agiilse arenduse projektis oled osalenud? \*

- 0
- 5
- 10
- 20
- 25

9. Milline oli agiilse arendusprojekti keskmine kestvus? \*

- 6
- 12
- 24
- 36

10. Umbes kui kaua (täisaastates) oled sa tegelenud agiilsete projektidega? \*

Väärtus peab olema arv

11. Palun märgi milliseid agiilseid metoodikaid sa oled kasutanud \*

- Scrum
- XP
- Feature Driven Development
- Adaptive Software Development
- Dynamic Systems Development Method (DSDM)
- Rational Unified Process (RUP)
- Crystal
- Kanban
- Mitte ühtegi neist
-



## Osa B - projektipõhine info

Palun meenuta viimast/käimasolevat projekti kus kasutati Scrumi või Scrumilaadset ehk agiilset meetodikat

12. Mis sektori projektiga oli tegu? \*

- Avaliku sektori projekt
- Avaliku sektori projekt eurorahadega
- Erasektori projekt
- Mittetulunduse projekt

13. Mis rolli sa täitsid projektis? \*

14. Kas projektis kasutati agiilset või waterfall (traditsioonilist) arendusmeetodikat? \*

- Agiilset
- Waterfall

15. Kas meeskonnas oli ka projektijuht? \*

- Jah
- Ei

16. Milline oli meeskonnaliikmete arv projektis? \*

- 0-5
- 6-10
- 11-15
- 16-20
- 21-25
- >25

17. Milline oli põhiline agiilne meetod mida projektis kasutati? \*

- Scrum
- XP
- Feature Driven Development
- Adaptive Software Development
- Dynamic Systems Development Method (DSDM)
- Rational Unified Process (RUP)
- Crystal
- Kanban
- Ei oska öelda
- Mitte ükski neist, oli waterfall
- Muu

## Osa C - Praktikad/tavad ja kaasamine

Palun vasta sama projekti näitel mida kasutasid eelmise osa vastamiseks.

### 18. Palun hinda järgnevate praktikate kasutamist \*

|  | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| User stories                                       | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product backlog                                    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Sprint backlog                                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Edenemismõõdikud (n burndown chart)                | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Daily standup                                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum of scrums koosolek                           | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Sprindid / lühikesed iteratsioonid                 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Retrospektiivid                                    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Reliisi planeerimine                               | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskonna poolne ajamahu hindamine                 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Sprint review                                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Sprindi planeerimine                               | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Info visualiseerimine (taskboard, postit kleepsud) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Kanban board                                       | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Agiilsed mängud                                    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Definition of done                                 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

19. Märgi kes osales ajahinnangute andmisel ja kui oluline / kaasatud ta oli. \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

20. Märgi kes osales "Product Backlog" haldamisel ja kui oluline / kaasatud ta oli. \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

21. Märgi kes osales "Sprint Backlog" haldamisel ja kui oluline / kaasatud ta oli. \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

22. Märgi kes mõõtis projekti kiirust (Project velocity) kui oluline / kaasatud ta oli. \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

23. Märgi kes oli vastutav vastuvõtukriteeriumite (Definition of DONE) määratlemise eest ja kui oluline / kaasatud ta oli. \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

24. Märgi kes viis läbi Sprindi planeerimiskoosolekuid ja kui oluline / kaasatud ta oli. \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

25. Märki kes vastutas Sprindi review koosolekute läbi viimise eest ja kui oluline / kaasatud ta oli \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

26. Märki kes vastutas standupide (daily scrum) läbi viimise eest ja kui oluline / kaasatud ta oli \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

27. Märki kes vastutas Sprindi retrospektiivide korraldamise ja läbiviimise eest ja kui oluline / kaasatud ta oli \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

28. Märgi kes tegelesid toote backlogi viimistlusega (Backlog Refinement) ja kui oluline / kaasatud ta oli \*

|                       | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Arendaja projektijuht | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| KeMIT projektijuht    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Analüütik             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum Master          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product Owner         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Meeskond (arendajad)  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

## Osa D - meetodite kasutamine ja hinnang

29. Palun hindade kui tihti kasutati teie projektis järgmisi tegevusi \*

|   | Ei kasutanud          | Haruharva             | Tihti                 | Pidevalt              | Ei tea                |
|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Kogu meeskonna otsused                                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| CI - continuous integration                             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Testipõhine arendus (TDD)                               | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Refaktoormine   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Paarisprogrammeerimine                                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Planeerimismängud                                       | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Väikesed väljalasked (release)                          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| System Methaphor  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Lihne koodi ülesehitus                                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Koodi ühisomand   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Klient kaasatud arendaja otsuste juurde                 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Koodi standardid  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Igapäevane 15 minutit standup (arendajad ja tootemanik) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Arendaja otsesuhtlus äripoollega (vahendajata)          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Kokkulepitud tarneajad                                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |



30. Sinu hinnang KeMITi (muus asutuses töötamise puhul sinu asutuse) valmisolekule agiilsete meetodikate kasutuselevõtuks (1 - ei ole valmis, 10 - täiesti valmis) \*

|                       |                       |                       |                       |                       |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

31. Sinu hinnang agiilsete meetodikate toimimise kohta (1 - ei toimi, 10 - toimivad suurepäraselt) \*

|                       |                       |                       |                       |                       |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

32. Kui hästi sa ennast tunnend agiilse lähenemise osas? (1- halvasti, 10 - väga hästi) \*

|                       |                       |                       |                       |                       |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**Lisa 2. Originaalküsimustik (Shastri et al., 2021b)**

# Participant Preinterview Information Form

Part A- General Information

**\*Required**

1. 1.Please type your full name. \*

---

2. 2.What is your current role/job title ? \*

---

3. 3.Please select your age bracket :

*Mark only one oval.*

- 20-29  
 30-39  
 40-49  
 50-60  
 Above 60

4. 4.What is your total work experience in the software industry? (in years) \*

*Mark only one oval.*

- 0-5  
 6-10  
 11-20  
 >20

5. 5. What is the number of waterfall/traditional projects you have participated in till date? \*

*Mark only one oval.*

- 0  
 1-5  
 6-10  
 11-20  
 >21

6. 6. What was the average duration of the waterfall/traditional projects? \*

*Mark only one oval.*

- Not applicable  
 6 months or less  
 6-12 months  
 12-24 months  
 >24 months

7. 7. What is the number of Agile projects you have participated in till date? \*

*Mark only one oval.*

- 0  
 1-5  
 6-10  
 11-20  
 >20

8. 8.What was the average duration of the Agile projects? \*

Mark only one oval.

- Not Applicable
- 6 months or less
- 6-12 months
- 12-24 months
- >24 months

9. 9.What would be the approximate number of years you have worked on agile projects?

Please enter a value.If you have not worked on an Agile project ,enter "0".

\_\_\_\_\_

10. 10.Please tick all the relevant agile methodologies you have used \*

Tick all that apply.

- Scrum
- XP
- Feature Driven Development (FDD)
- Adaptive Software Development
- Dynamic Systems Development Method (DSDM)
- Rational Unified Process (RUP)
- Crystal
- Kanban
- None

Other:  \_\_\_\_\_

Part B:  
Project  
Information

IMPORTANT: Please recall ONE current/past project or product where you used Scrum or an adaptation of Scrum, and which stands out in your experience and use that project/product to answer the following.This project/product will be the basis of our interview conversation.

11. 1.What sector or domain was the project/product in? \*
- For example: Healthcare, Petrochemicals, Manufacturing, Finance & Banking etc.

\_\_\_\_\_

12. 2.What was your job title on the project/product? \*

\_\_\_\_\_

13. 3.Did the project use Agile or was it Waterfall/traditional? \*

*Mark only one oval.*

- Agile
- Waterfall/Traditional
- Other: \_\_\_\_\_

14. 4.Was there a Project Manager in your project/product team? \*

*Mark only one oval.*

- Yes
- No

15. 5. What was the number of team members on your project/product? \*

*Mark only one oval.*

- 0-5
- 6-10
- 11-15
- 16-20
- 21-25
- >25

16. 6.What was the main Agile method used in your project/product? \*

Mark only one oval.

- Not Applicable *Skip to question 17*
- Scrum *Skip to question 17*
- XP *Skip to question 17*
- Feature Driven Development (FDD) *Skip to question 17*
- Adaptive Software Development *Skip to question 17*
- Dynamic Systems Development Method (DSDM) *Skip to question 17*
- Rational Unified Process (RUP) *Skip to question 17*
- Crystal *Skip to question 17*
- Kanban *Skip to question 17*
- Other: \_\_\_\_\_

**Part C- Practices and  
Involvement (Scrum)**

Using the same project whose details you filled out in Section B,  
please complete this section.

17. 7.Please rate the frequency of using the practices listed below \*

The rating works as follows: 1- The practice is not used at all, 2-Occasionally used, 3- Frequently used, 4- Very frequently or Regularly used; and 0-Not applicable.

Mark only one oval per row.

|   | 1 (Not used)          | 2 (Occasionally)      | 3 (Frequently)        | 4 (Regularly)         | 0 (Not Applicable)    |
|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| User Stories                                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Product backlog                                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Sprint backlog                                    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Project Velocity measurement (eg burndown charts) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Daily Scrums/Daily standup                        | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Scrum of Scrums meeting                           | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Short Iterations/Sprints                          | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Retrospectives                                    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Release planning                                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Team based estimation                             | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Iteration /Sprint reviews                         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Iteration/Sprint Planning                         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Information visualization (e.g. taskboard)        | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Kanban board                                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Agile games                                       | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Definition of done                                | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

18. 8.Please indicate who performed "Estimation" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1-No involvement , 2-Somewhat involved, 3-High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision-maker) | 0 (Not Applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |

19. 9.Please indicate who maintained the "Product Backlog" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1- No involvement , 2-Somewhat involved, 3-High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision maker) | 0 (Not applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |



20. 10. Please indicate who maintained the "Sprint Backlog" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1- No involvement , 2-Somewhat involved, 3-High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision maker) | 0 (Not applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |

21. 11. Please indicate who measured the "Project velocity" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1- No involvement , 2-Somewhat involved, 3-High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision maker) | 0 (Not applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |

22. 12. Please indicate who was responsible for "Definition of Done" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1- No involvement, 2- Somewhat involved, 3- High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision maker) | 0 (Not applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |

23. 13. Please indicate who conducted the "Sprint planning meeting" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1- No involvement, 2- Somewhat involved, 3- High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision maker) | 0 (Not applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |

24. 14. Please indicate who was responsible for "Sprint review meeting" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1- No involvement , 2-Somewhat involved, 3-High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision maker) | 0 (Not applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |

25. 15. Please indicate who was responsible for organizing "Daily Scrum" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1- No involvement , 2-Somewhat involved, 3-High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision maker) | 0 (Not applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |

26. 16. Please indicate who was responsible for organizing "Sprint retrospective" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1- No involvement, 2- Somewhat involved, 3- High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision maker) | 0 (Not applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |

27. 17. Please indicate who performed "Product Backlog Refinement" and rate the frequency on the scale \*

Please rate the level of involvement on a scale of 0 to 4. The rating works as follows: 0- Not Applicable, 1- No involvement, 2- Somewhat involved, 3- High involvement, 4- Key decision maker.

Mark only one oval per row.

|                                   | 1 (No involvement)    | 2 (Somewhat involved) | 3 (High involvement)  | 4 (Key decision maker) | 0 (Not applicable)    |
|-----------------------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|
| Project Manager                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Team Members (Developers/Testers) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Business Analyst                  | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Scrum Master                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |
| Product Owner                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/> |

Part C- Practices and involvement (XP)

28. 7. Please rate the frequency of using the practices listed below \*

Please rate each practice on a scale of 0 to 4. The rating works as follows: 0-Not applicable, 1- The practice is not used at all, 2-Occasionally used, 3- Frequently used, 4- Very frequently used.

Mark only one oval per row.

|                               | 1                     | 2                     | 3                     | 4                     | 0                     |
|-------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Whole Team                    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Continuous Integration        | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Test-driven development (TDD) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Refactoring                   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Pair programming              | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Planning game                 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Small releases                | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| System Metaphor               | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Simple design                 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Collective code ownership     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| On-site customer              | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Coding standards              | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**Lisa 3. Tulemuste korrelatsioonitabel**

|   | Kogu meeskonna otsused | CI kasutamine | Testipõhine arendus (TDD) | Refaktoormine | Paarisprogrammeerimine | Planeerimismängud | Väiksed väljalasked (release) | System Methaphor | Lihntne koodi ülesehitus | Koodi ühisomand | Klient kaasatud arendaja otsuste juurde | Koodi standardid | Arendaja otsesuhtlus äripoollega (vahendajata) | Kokkulepitud tarneajad | Kemiti valmisolek | Sinu hinnang agilsete metoodikate toimimise kohta | Hinnang endale |
|---|------------------------|---------------|---------------------------|---------------|------------------------|-------------------|-------------------------------|------------------|--------------------------|-----------------|---|------------------|--|------------------------|-------------------|---|----------------|
| Kogu meeskonna otsused                  | 1,000                  |               |                           |               |                        |                   |                               |                  |                          |                 |   |                  |  |                        |                   |   |                |
| CI kasutamine                           | 0,653                  | 1,000         |                           |               |                        |                   |                               |                  |                          |                 |   |                  |  |                        |                   |   |                |
| Testipõhine arendus (TDD)               | -                      | -             | 1,000                     |               |                        |                   |                               |                  |                          |                 |   |                  |  |                        |                   |   |                |
| Refaktoormine                           | 0,006                  | 0,222         | 0,400                     | 1,000         |                        |                   |                               |                  |                          |                 |   |                  |  |                        |                   |   |                |
| Paarisprogrammeerimine                  | 0,441                  | 0,251         | 0,219                     | 0,292         | 1,000                  |                   |                               |                  |                          |                 |   |                  |  |                        |                   |   |                |
| Planeerimismängud                       | 0,230                  | 0,173         | 0,097                     | 0,295         | 0,505                  | 1,000             |                               |                  |                          |                 |   |                  |  |                        |                   |   |                |
| Väiksed väljalasked (release)           | 0,148                  | 0,087         | 0,248                     | 0,269         | 0,442                  | 0,111             | 1,000                         |                  |                          |                 |   |                  |  |                        |                   |   |                |
| System Methaphor                        | 0,079                  | 0,059         | 0,033                     | 0,101         | 0,194                  | 0,326             | 0,285                         | 1,000            |                          |                 |   |                  |  |                        |                   |   |                |
| Lihntne koodi ülesehitus                | 0,077                  | 0,179         | 0,308                     | 0,167         | 0,278                  | 0,318             | 0,085                         | 0,163            | 1,000                    |                 |   |                  |  |                        |                   |   |                |
| Koodi ühisomand                         | 0,492                  | 0,664         | 0,008                     | 0,312         | 0,062                  | 0,218             | 0,078                         | 0,127            | 0,202                    | 1,000           |   |                  |  |                        |                   |   |                |
| Klient kaasatud arendaja otsuste juurde | 0,156                  | 0,389         | 0,411                     | 0,091         | 0,235                  | 0,108             | 0,433                         | 0,055            | 0,097                    | 0,068           | 1,000                                   |                  |  |                        |                   |   |                |
| Koodi standardid                        | 0,492                  | 0,664         | 0,059                     | 0,312         | 0,321                  | 0,405             | 0,307                         | 0,207            | 0,059                    | 0,808           | 0,042                                   | 1,000            |  |                        |                   |   |                |

|  | Kogu meeskonna otsused | CI kasutamine | Testipõhine arendus (TDD) | Refaktoormine | Paarisprogrammeerimine | Planeerimismängud | Väiksed väljalasked (release) | System Methaphor | Lihtne koodi ülesehitus | Koodi ühisomand | Klient kaasatud arendaja otsuste juurde | Koodi standardid | Arendaja otsesuhtlus äripoolega (vahendajata) | Kokkulepitud tarneajad | Kemiti valmisolek | Sinu hinnang agiilsete metoodikate toimimise kohta | Hinnang endale |
|--|------------------------|---------------|---------------------------|---------------|------------------------|-------------------|-------------------------------|------------------|-------------------------|-----------------|---|------------------|---|------------------------|-------------------|--|----------------|
| Arendaja otsesuhtlus äripoolega (vahendajata)      | 0,730                  | 0,832         | 0,222                     | 0,187         | 0,384                  | 0,208             | 0,028                         | 0,124            | 0,000                   | 0,592           | 0,368                                   | 0,592            | 1,000   |                        |                   |  |                |
| Kokkulepitud tarneajad                             | 0,728                  | 0,830         | 0,009                     | 0,295         | 0,178                  | 0,000             | 0,091                         | 0,046            | 0,191                   | 0,690           | 0,475                                   | 0,690            | 0,887   | 1,000                  |                   |  |                |
| Kemiti valmisolek                                  | 0,673                  | 0,705         | 0,172                     | 0,181         | 0,410                  | 0,090             | 0,255                         | 0,300            | 0,149                   | 0,399           | 0,525                                   | 0,445            | 0,736   | 0,761                  | 1,000             |  |                |
| Sinu hinnang agiilsete metoodikate toimimise kohta | 0,822                  | 0,586         | 0,202                     | 0,199         | 0,417                  | 0,000             | 0,399                         | 0,098            | 0,263                   | 0,367           | 0,449                                   | 0,416            | 0,582   | 0,656                  | 0,780             | 1,000  |                |
| Hinnang endale                                     | 0,780                  | 0,730         | 0,019                     | 0,058         | 0,427                  | 0,166             | 0,121                         | 0,096            | 0,154                   | 0,428           | 0,547                                   | 0,471            | 0,884   | 0,785                  | 0,824             | 0,798  | 1,000          |

## **SUMMARY**

### **IMPLEMENTATION OF THE AGILE PRINCIPLES IN MANAGEMENT OF PROJECTS IN THE ADMINISTRATIVE AREA OF THE MINISTRY OF THE ENVIRONMENT**

Sten Viljus

Agile development methods are a popular topic in software development. Agile development, Scrum and DevOps – the meanings of these words are widely interpreted by different organizations that use these words to introduce their services.

The aim of this work was to provide an overview of the agile software development processes and explore the possibilities for the introduction of agile methods in the administrative area of the Ministry of the Environment. Also to find out which agile methodology would be most suitable for the implementation in the Information Technology Centre of the Ministry of the Environment (KEMIT). According to the author, the research questions were answered and the related objective was met.

When compiling an overview of literature, it appeared that the organizations must overcome both technological and organizational obstacles for adopting agile methods. Exactly the same result was achieved by the author when carrying out semi-structured interviews following the pre-interview questionnaire. In the interviews, the more competent the respondents felt themselves about the agile principles, the lower they rated the readiness of the organization as a whole. The answers from the interviews revealed that several respondents would be prepared to take much greater responsibility than they have been before.

The author has submitted summarizing information and mapped the risks to be taken into account when starting the implementation of agile methods based on observations. The



author's proposals are presented to supplement the software development and software delivery process. The mapping prepared as a result of the work and the list of risks and opportunities are valuable material that KEMIT can use in the process of moving towards a more vibrant software development process. According to the author, the organization as a whole is not yet ready for a aggressive transition to agile software development methodology. However, the author finds that if the management's decision to move towards agile methodologies in software development remains, the disciplined DevOps could be the framework on which to move forward.

The volume of the work did not allow the development of action plans and in this connection, the author suggests that the topic could go further. As a follow-up, the work skills and knowledge needed for the agile development methodology of KEMIT employees should be mapped and analyzed in order to make a specific plan on how and to what extent the disciplined DevOps will be implemented for the organization. As Desmond Tutu has said, “The only way to eat an elephant is to eat it by mouthful”. Thus, the author also recommends moving towards agileness in a gentle way, with small steps.

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Sten Viljus,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

AGIILSETE PRINTSIIPIDE RAKENDAMINE  
KESKKONNAMINISTEERIUMI HALDUSALA PROJEKTIDE JUHTIMISEL,

mille juhendaja on Arvi Kuura,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Sten Viljus

19.05.2021