



Carmen Aasamets

EESTI RIIGI INFOTEHNOLOOGIA ASUTUSTE ÜLEMINEK DEVOPS PÕHIMÕTETELE

LÕPUTÖÖ

Teenusmajanduse instituut
Majandusinfosüsteemide korraldamise õppekava
Juhendaja: Heli Freienthal, *MBA*

Mõdriku 2024

Autori deklaratsioon ja lihtlitsents

Mina, Carmen Aasamets, tõendan, et lõputöö on minu kirjutatud. Töö koostamisel kasutatud teiste autorite, sh juhendaja teostele on viidatud õiguspäraselt.

Kõik isiklikud ja varalised autoriõigused käesoleva lõputöö osas kuuluvad autorile ainuisikuliselt ning need on kaitstud autoriõiguse seadusega.

Juhendaja Heli Freienthal */allkirjastatud digitaalselt/*

Lõputöö on kaitsmisele lubatud teenusmajanduse instituudi direktori korraldusega nr 1-14/24/2.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Carmen Aasamets sünnikuupäev: 05.07.1993 annan Tallinna Tehnikakõrgkoolile (edaspidi kõrgkool) tasuta loa (lihtlitsentsi) enda loodud teose „Eesti riigi infotehnoloogia asutuste üleminek DevOps põhimõtetele“.

1. reprodutseerimiseks paberkandjal kõrgkooli raamatukogus avaldamise ja säilitamise eesmärgil;
2. elektroonseks avaldamiseks kõrgkooli repositooriumi kaudu;
3. kui lõputöö avaldamisele on instituudi direktori korraldusega kehtestatud tähtajaline piirang, lõputöö avaldada pärast piirangu lõppemist.

Olen teadlik, et nimetatud õigused jäävad alles ka autorile ja kinnitan, et:

1. lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid ega muid õigusi;
2. PDF-failina esitatud töö vastab täielikult kirjalikult esitatud tööle.

Mõdrikul, 08.01.2024 */allkirjastatud digitaalselt/*

SISUKORD

SISSEJUHATUS.....	4
1 DEVOPS OLEMUS JA JUURUTAMINE	6
1.1 DevOps olemus, kasutamise eesmärk ja areng.....	7
1.2 DevOps põhimõtete võrdlus teiste tarkvaraarenduse meetoodikatega.....	8
1.2.1 DevOps ja agiilsete meetoodikate elutsükli faaside võrdlus	13
1.2.2 DevOps eelised võrreldes teiste arendusmeetoodikatega	16
1.2.3 DevOps töökultuurile üleminekut piiravad tegurid	20
1.3 DevOps meetoodika juurutamise protsessi põhimõtted	22
1.4 DevOps meetoodika täiustamine DevSecOps põhimõtetega	24
2 EMPIIRILISE UURINGU METOODIKA	26
3 EESTI RIIGI INFOTEHNOLOOGIA ASUTUSTE DEVOPS PÕHIMÕTETELE ÜLEMINEKU UURINGU TULEMUSED	29
3.1 DevOps meetoodika määratlemine ja arendusmeetoodika muutmist tinginud tegurid riigi infotehnoloogia asutustes	29
3.2 DevOps põhimõtete võrdlus ning eelised teiste tarkvaraarenduse meetoodikatega riigi infotehnoloogia asutustes	31
3.3 DevOps mõju asutustele ning meeskondade tagasiside	36
3.4 DevOps põhimõtetele üleminek ja juurutamine riigi infotehnoloogia asutustes.....	41
3.5 Ettepanekud DevOps põhimõtetele üleminekuks.....	45
KOKKUVÕTE.....	49
SUMMARY	51
VIIDATUD ALLIKAD.....	53
LISAD	58
LISA 1 INTERVJU KÜSIMUSED.....	59

SISSEJUHATUS

Üks täpsemaid definitsioone DevOps'i osas on välja toodud autorite Jabbari, Ali, Petersen, Tanveer uurimistöös „Mis on DevOps?: Määratluste ja tavade süstemaatilise kaardistamise uuring“. Uurimistöös välja toodud definitsiooni kohaselt on DevOps arendusmetoodika, mille eesmärk on tarkvaraarenduse ja -halduse vahel rõhutada suhtlemist, koostööd, pidevat integreerimist ning samal ajal kvaliteedi tagamist ja automatiseeritud tarkvara tarnimist (Jabbari, Ali, Petersen, Tanveer, 2016).

Lõputöö teema on akadeemilises vaates uudne, kuna uurimistöid antud teemal on Eestis vähe ja temaatikal puudub kindel raamistik ning ühtne definitsioon. DevOps on saanud akadeemiline määratlus kolmelt arvutiteaduse uurijalt: Bass, Weber ja Zhu, kes on DevOps'i nimetanud kogumiks, mille eesmärk on vähendada aega süsteemi muutmise ja tootekeskonda viimise vahel, samal ajal tagades kõrge kvaliteet (Filip, 2020). DevOps kirjandus käsitleb peamiselt arendusmetoodika põhimõtteid ja parimate praktikate jagamist. Paljud organisatsioonid muudavad oma arendusprotsesse, eesmärgiga muuta need agiilsemaks. Agiilne arendusmetoodika on oluline, kuna võimaldab arendustiimidel kiiresti reageerida äri vajaduste muutumisele. Agiilne metoodika keskendub peamiselt tellija ja arendajate vahelisele suhtlusele ning rõhutab kiiret tarkvaraarendust. (Brush, Silverthorne, 2019)

Küll aga võib välja tuua, et vähem pööratakse tähelepanu tarkvaraarendajate ja haldajate omavahelisele suhtlusele ning protsesside parendamisele. Arendused saavad kiiresti valmis, kuid kasutusele võtmine on probleemne ja ajakulukas. Antud teema uurimine on vajalik, et infotehnoloogia asutused tegutseksid lisaks agiilse poole arendamisele ka tarneprotsessi arendamise ja automatiseerimisega.

Vastavalt Rahandusministeeriumi suunistele on Eesti riigi infotehnoloogia asutuste suund liikuda lõimitud tiimide praktika ja agiilsemate arendusmetoodikate suunas, kus on oluline, et arenduste tulemused luuakse kiiresti ning arenduste kasutuselevõtt oleks agiilsem (IKT arenduste toimemudel, 2023). Uurimisprobleem tuleneb Eesti riigi infotehnoloogia asutuste vajadusest töömeetodeid uuendada ja arendusprotsessi automatiseerida rakendades DevOps põhimõtteid.

Lõputöö eesmärk on kaardistada nelja Eesti riigi infotehnoloogia asutuse näitel DevOps põhimõtetele üleminekut ja arendusmetoodika muutmist tinginud tegureid.

Eesmärgi täitmiseks püstitatud ülesanded:

- tuua välja DevOps olemus ja kasutamise eesmärk;
- võrrelda DevOps olemust ja põhimõtteid teiste tarkvaraarenduse metoodikatega;
- tuua välja DevOps juurutamise protsessi põhimõtted ning DevOps metoodika täiustamine DevSecOps põhimõtetega;
- lähtuvalt valitud metoodikast luua uurimisinstrument empiirilise uuringu läbiviimiseks;
- kaardistada asutuste DevOps põhimõtetele üleminekut, arendusmetoodika muutmist tinginud tegureid ja uue metoodika eeliseid võrreldes asutuses eelnevalt kasutusel olnud arendusmetoodikaga;
- saadud tulemuste põhjal teha järeldusi ja ettepanekuid, mis võiksid olla abiks teistele infotehnoloogia asutustele DevOps põhimõtetele üleminekul.

Töö teoreetilises osas käsitletakse DevOps olemust, kasutamise eesmärki ning võrreldakse DevOps printsiipe teiste tarkvaraarenduse metoodikatega, tuuakse välja DevOps metoodika eeliseid ning ühtlasi ka piiranguid DevOps töökultuurile üleminekul. Lisaks tuuakse välja DevOps juurutamise põhimõtted ning metoodika täiustamise võimalused DevSecOps metoodikaga.

Lõputöös ei tooda täpselt välja võimalikke tehnoloogiaid, rakendusi või teenuseid, mis aitavad rakendada DevOps põhimõtteid ning ei käsitleta põhjalikult erinevaid tarkvaraarendusmetoodikaid. Lõputöö keskendub peamiselt arenduse ja halduse lõimimise põhimõtetele, DevOps ülemineku ja juurutamise uurimisele.

1 DEVOPS OLEMUS JA JUURUTAMINE

DevOps (*Development and Operations*) on tarkvaraarenduse metoodikate kogum, mis on loodud tarkvaraarenduse ja tarkvara toimingute integreerimiseks. Seda nähakse sageli ka liikumise või kultuurina (Loukides, 2012). Akronüüm tuleneb ingliskeelsest sõnast *Development and Operations* ehk arendus ja käitus (AKIT, s.a). DevOps võeti terminina kasutusele 2009. aastal, kui Debois korraldas Belgias konverentsi, mille nimi oli *DevOps Days*. Konverentsi järel toimus *Twitter*’is jätkuv arutelu, ja *Twitter*’i 140-tähemärgise piirangu tõttu hakati kasutama lühendit *#devops*, mis asendas algse viite *#devopsdays* (Edwards, 2019).

Aastate jooksul on DevOps kujunenud välja kui arenduse ja halduse koostöömudel, mis aitab ületada lõhe IT arenduse ja halduse vahel, võimaldades rakendada järjepidevat arendustsükli ja sagedast tarkvara tarnimist. DevOps metoodika soodustab arenduse ja halduse ühiseid ja tõhusamaid tööprotsesse. (Erich, 2014) DevOps mõiste on sündinud kahest omavahel seotud trendist. Esimest nimetatakse „paindlikuks süsteemihalduseks“, mis arenes välja paindlike arendusmetoodikate rakendamisel süsteemihalduses. Teine trend on „väärtuse tunnetamine“, mis tuleneb arendus- ja haldusmeeskondade koostööst kõigis arendustsükli etappides (Mueller, 2020). DevOps on seotud arendus- ja haldustiimidega, kes teevad koostööd tootlikkuse parandamiseks, töövoogude automatiseerimiseks ning rakenduste jõudluse pidevaks mõõtmiseks, et pakkuda lühema aja jooksul rohkem tarkvaralahendusi (Filip, 2020).

DevOps koondab tarkvaraarenduse metoodikad ja põhimõtted, mis võivad erinevates käsitlustes kohati erineda. DevOps’i põhimõtete ja kasutuselevõtu raamatu autor Sharma, kirjeldab DevOps’i kui tarkvaraarenduse metoodikat, mis rõhutab tarkvaraarendajate ja süsteemiadministraatorite vahelise suhtluse, koostöö ja integratsiooni tähtsust (Sharma, 2017). DevOps’i juhtumiuuringu autorid Senpathi, Buchan ja Osman defineerivad DevOps’i kui silda toote arendamise ja tarnimise vahel, mis saavutatakse avatud suhtluse, usalduse, austuse, ning motivatsiooniga. Eesmärk on võimaldada järjepidevat ja kiiret toote arendust ja tarnimist ning pakkuda samal ajal teenuseid, mis aitavad toetada arendamise elutsükli (Senapathi, Buchan, Osman, 2018).

Eelkõige tähendab DevOps tihedat koostööd tarkvaraarenduse spetsialistide ja teiste IT-spetsialistide vahel, kes ei ole tarkvaraarendusega otseselt seotud, sealhulgas erinevate valdkondade administraatorid (süsteemid, võrgu- või andmebaasiadministraatorid) ja IT-toodete

omanikud/projektijuhid, tugi- ja tarkvara testijad. Laiem lähenemine DevOps'ile ei hõlma ainult IT-arenduse ja -halduse koostööd, vaid ka lõppkasutajate vahelist koostööd.

1.1 DevOps olemus, kasutamise eesmärk ja areng

DevOps meeskonnad töötavad koos, et automatiseerida infrastruktuuri, töövooge ja mõõta koos rakenduste jõudlust. DevOps'i eesmärk on parandada kogu arendustsükli üldist tootlikkust ja efektiivsust. Antud arendusmetoodikat rakendavad tiimid keskenduvad automatiseerimisele – koodi testimisest töövoog ja üldise toimimiseni. (Filip, 2020) Puppet Labs'i tehtud uurimused on ilmestanud, et ettevõtted, mis kasutavad DevOps'i, toimivad infotehnoloogia valdkonnas enamasti tõhusamalt. Puppet Labsi iga-aastases DevOps'i ülevaates on rõhutatud, et organisatsioonide efektiivsus suureneb, mida pikemalt nad on DevOps'i meetodit rakendanud (Labs, State of DevOps Report, 2014). Kui võrrelda agiilset arendusmetoodikat, mis põhineb tarkvara iteratiivse arendamise põhimõttel ehk kiht-kihi haaval arendusel, siis DevOps arenduse puhul luuakse ja juurutatakse väikesed koodijupid samal või järgmisel päeval (Filip, 2020).

Kui soovitakse laiendada oma turgu üha kiiremini arenevas, keerulises ja ebamäärases maailmas, kus konkureerimine põhineb digitaalsete kanalite ja mobiilseadmete pideval evolutsioonil, on ettevõtetal jätkusuutlik strateegia DevOps metoodika juurutamine. Lisaks tõhusale koostööle arendus- ja haldusmeeskondade vahel vastutab DevOps süsteemide arendamise ja automatiseerimise eest, et pakkuda võimalikult sujuvat, kiiret ja usaldusväärset tarkvara (Ravichandran, Taylor, Waterhouse, 2016). DevOps metoodika on tihedalt seotud pideva arenduse integreerimise ja tarnimise põhimõtetega, mis aitavad organisatsioonidel arendustegevuse kaudu kiiremini turutingimustele reageerida. Suures pildis on antud töökultuuri põhimõte - liikuda lühemate arendustsüklite, sagedasemate juurutamiste ning töökindlamate tarkvaraversioonide avalikustamise poole. (Loukides, 2012) Sealjuures on oluline, et struktuuris oleksid arendus- ja haldus koos ehk tekiks autonoomne tiim, mitte eraldi meeskondades, ehk asutuses tuleks üle vaadata struktuur. Arendus ja haldus ei tohiks olla struktureeritud töötama eraldiseisvate üksustena omaenda visioonide, eesmärkide ja vastutustega. Kui arendus ja haldus ei tööta koostöös, siis töövoog takerdub kiiresti ja tootlikkus kannatab. (Mazyar, 2018)

DevOps mõtteviisil pannakse suurt rõhku töökultuurile. See sisendab meeskondades ühtset visiooni, mis on otseselt seotud ettevõtte ja selle klientidega. DevOps töökultuuris ei süüdistata vigades ühte indiviidi vaid vastutab kogu meeskond (Ravichandran, Taylor, Waterhouse, 2016). DevOps'i peetakse peamiselt IT-organisatsiooni kultuuriliseks töökorralduse muudatuseks, kuigi seda võib

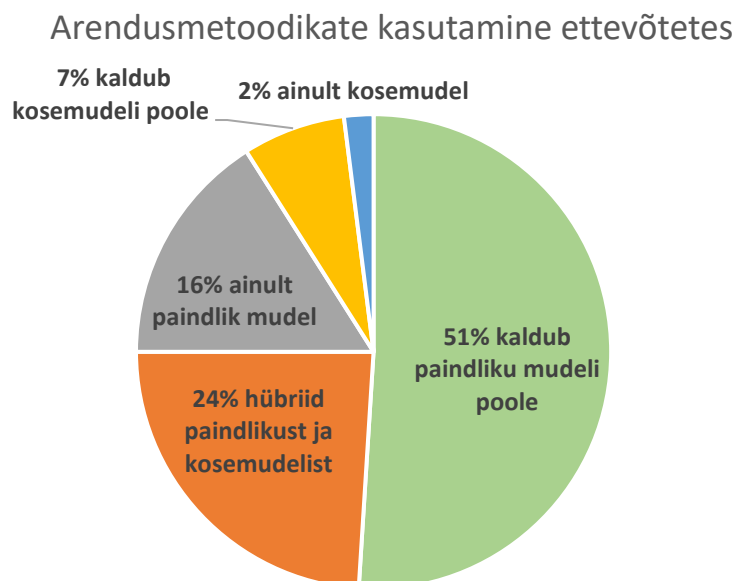
käsitleda ka protsesside ja tööriistade muutusena. See on vaadeldav töökultuuri muutusena, mille tulemuseks on arendus- ja haldustegevuste koostöö, mis tähendab erinevate tarkvara loomisega seotud osapoolte eesmärkide ühtlustumist. Antud töökorralduse edukust mõõdetakse eeskätt kultuuri, automatiseerimise, mõõtmise ja teadmiste jagamise kaudu (Sanchez-Gordon, Colomo-Palacios, 2018).

Kokkuvõtvalt võib välja tuua, et DevOps on töökorralduse muudatus, mille rakendamise eesmärk hõlmab kõiki tarkvara loomisega seotud etappe – sagedased juurutamised, lühem aeg tarkvaraversioonide avalikustamisel, väiksem vigade võimalus uuel avalikustatud tarkvaraversioonil ning lühem aeg vigade parandamisel. Struktuuris arendus- ja haldus kokku, et saaks tekkida multifunktsionaalsed autonoomsed tiimid, kus DevOps tiimil on otsast lõpuni vastutus teenuse ja toote eest ning kliendikeskne tegutsemine. DevOps metoodikat rakendatakse, et operatsiooni protsesside tõhusus, turvalisus ja hooldus oleks efektiivsemad.

1.2 DevOps põhimõtete võrdlus teiste tarkvaraarenduse metoodikatega

Tarkvaraarenduse tavad on aastate jooksul oluliselt muutunud –klassikalisest kosemudelist on liigutud iteratiivsema, paindlikuma ning agiilsema lähenemiseni, et lõppkasutajale tarkvara kiiremini tarnida. Peamiseks motivatsiooniks paindlikuma tarkvaraarenduse kasutamisel on veendumus, et antud arenduskäsitlus on kasutajakeskem ja parandab arendusmeeskondade vahelist koostööd (Jeremiah, Survey: Is agile the new norm?, 2016). Paindlikuma tarkvaraarendusega planeeritakse, arendatakse ja tarnitakse tarkvara võimalikult väikestes äriselt kasulikes osades, mille tulemuseks on kiiremad ja sujuvamad lahendused lõppkasutajatele (Ries, 2009). Lisaks saavad arendajad kohest tagasisidet oma arenduse kohta ning muudatusi ja täiendusi tehakse kogu tarkvaraarenduse elutsükli vältel.

2016. aastal viidi TechBaconi poolt läbi veebiküsitlus tarkvaraarendajate seas. Uuringus osales 600 tarkvaraarendajat.



Joonis 1. Arendusmeetodid ettevõtetes 2016. aasta TechBacon uuringu kohaselt (Jeremiah, Survey: Is agile the new norm?, 2016)

Uuringust selgus, et paindliku tarkvara printsiibid olid juba 2016. aastal kasutuses umbes 2/3 uuringus osalenud ettevõtetes. Hübriidne lähenemine on kasutusel 24% vastanutest, mis tähendab, et neil on kasutusel mõned agiilsed põhimõtted. Kokku 9% kirjeldab end „ainult kosemudeli“ või „kalduva kosemudeli poole“.

Agiilne metoodika on suuresti seotud mõtteviisi muutusega. See on iteratiivne lähenemine projektijuhtimisele ja tarkvaraarendusele, mis keskendub koostööle, klientide tagasisidele ja kiirele arenduste väljalasketele (Hall, s.a). Agiilsete põhimõtete rakendamine nõuab teisi töövahendeid kui näiteks klassikaline kosemudeli metoodika. Agiilsete metoodikate rakendamisel on oluline, et arendusmeeskond saaks võimalikult varajases faasis aru:

- kas loodav lahendus liigub õiges suunas;
- kas töö vastab tellija poolt esitatud soovidele;
- kas arenduse tellija ja arendusmeeskond on saanud üksteisest õigesti aru.

Ühtlasi aitab antud põhimõtete rakendamine nii tellija kui ka arendusmeeskonna jaoks maandada riske ja arvestada kulusid, sest kui tulevad välja vead või töö käigus selgub, et arenduse tellija soov on muutunud, siis ei ole teostatud kulud nii kõrged kui need oleksid suurema infosüsteemi loomisel.

Kui tuua välja kose metoodikal arendus, kus tööd teostatakse järjestikku ehk faasidena ning töö tulemusi näeb liiga hilises faasis, siis võrdluseks võib tuua, et agiilne metoodika annab tagasiside varajasemas faasis ning võimaluse tehtud vigu finantsiliselt soodsamalt parandada. (Morris, 2022)

DevOps ühendab arenduse ja halduse ning muudab kiiremaks arenduste väljastamise efektiivsuse, kiiruse ja turvalisuse võrreldes traditsiooniliste protsessidega (GitLab, s.a). Paindlikum tarkvaraarenduse elutsüklil annab ettevõtetele ja nende klientidele konkurentsieelise. DevOps metoodika optimeerib ja kiirendab protsesside toimimist. Võib ka välja tuua, et DevOps metoodika täiendab agiilset arendust, juhtides tähelepanu IT- haldusosakonna rollile. DevOps tõstab esile agiilse infrastruktuuri ehk koos tellija vajaduste agiilse rakendamisega tarkvarakoodis peab toimima ka infrastruktuuri paindlik haldus. (Debois, 2008)

DevOps meetod tarkvaraarenduses võimaldab DevOps tiimidel tarkvara kiiremini ja kindlamalt arendada, testida ning turule tuua, rakendades paindlikke reegleid ja tavasid. Arendus, testimine ja turule toomine leiavad aset nii agiilse kui ka DevOps lähenemise raames. DevOps põhimõtete kohaselt töötavad arendajad ja haldusmeeskonnad ühiselt läbi kogu rakenduste loomise, juurutamise ja haldamise protsessi tiimina (Hall, s.a). IBM (*International Business Machines Corporation*, rahvusvaheline arvutite, tehnoloogia, ärinõustamise ja infotehnoloogia konsultatsiooni ettevõtte) soovitus järgi peaks DevOps meeskonna optimaalne suurus olema kuni 10 liiget, mis annab võimaluse, et kõik liikmed saavad osaleda meeskonna peamiste funktsioonide juhtimises (IBM, s.a).

DevOps on üles ehitatud agiilse tarkvaraarendusega sarnastele põhimõtetele, "*The Agile Manifesto*":

- protsesside ja töövahendite asemel eelistatakse inimesi ja kommunikatsiooni;
- põhjaliku dokumenteerimise asemel on oluline toimiv tarkvara;
- eelistatud on kliendisuhetus lepinguliste läbirääkimiste asemel;
- ei tehta kindlat plaani vaid oodatakse muudatustele kiiret reageerimist (Thomas, 2001).

Vastavalt eelnevas loetelus välja toodud põhimõtetele võib öelda, et kuigi agiilne tarkvaraarendus ja DevOps tuginevad sarnaselt printsiipidel, on neil teatavad erinevused. Kui agiilne tarkvaraarendus keskendub peamiselt tarkvara arendamisele ehk kui tarkvara on arendatud ning tarnitud, siis arendusmeeskond sellega enam ei tegele (Hamilton, 2023). DevOps järgib, et tarkvara oleks kõlblik tarnimiseks, et seal oleks võimalikult vähe vigu ning oleks kasutuselevõtuks valmis. DevOps täidab osaliselt ka edaspidi kasutajatoe rolli, aidates klientidel oma ootusi edasi arendada ja äriprotsesse üha enam automatiseerida (Franklin, 2016).

Järgnevalt on toodud DASA (*DevOps Agile Skills Association*, DevOps agiilsete oskuste assotsiatsioon) poolt kirjeldatud DevOps kuus põhiprintsiipi:

- kliendikeskselt tegutsemine (*Customer-Centric Action*);
- lõppeesmärgi peale mõtlemine (*Create with the End in Mind*);
- vastutus algusest lõpuni (*End-to-End Responsibility*);
- multifunktsionaalsed autonoomsed tiimid (*Cross-Functional Autonomous Teams*);
- pidev parendamine (*Continuous Improvement*);
- kõige automatiseerimine, mis võimalik (*Automate Everything You Can*) (DASA DevOps Certification, s.a).

Lisaks eelnevalt toodud loetelule on DevOps töökultuuris oluline tiimi loomine ja koostöö, pidev parendamine ja probleemilahendamine, julgus ja eksperimenteerimine ning eestvedamine ja tagasiside (DASA DevOps Certification, s.a). Scrum meetodi kohaselt suudavad meeskonnaliikmed projekti jooksul kõik vajalikud tööd ära teha. Scrum põhineb sprintidel, mille kestus varieerub tavaliselt ühest nädalast kuni ühe kuuni (Franklin, 2016). Scrum ja DevOps on kaks üksteist täiendavat metoodikat, mida saab tarkvaraarenduse täiustamiseks koos kasutada. Integreerides DevOps praktikad Scrum protsessi, saavad Scrum meeskonnad DevOps meeskondadega töötada, et pakkuda kvaliteetseid tarkvaratooteid kiiremini ja tõhusamalt (Dash Technologies Inc, 2022).

Agiilne arendusmetoodika ei eelda toimimiseks kindlaid töövahendeid. Kogu kommunikatsioon võib toimida ka nt e-kirjade teel. DevOps sõltub suurel määral töövahendite ning tööloikude automatiseeritusest (Franklin, 2016). DevOps metoodikal on suur roll ka süsteemiadministraatoritel, kes välja arendatud toote elukäiku jälgivad, hooldavad ja tagasisidet annavad. Kogu protsess peaks olema võimalikult automatiseeritud. DevOps põhimõtete puhul on oluline välja tuua, et protsessi jälgitakse pidevalt, näiteks kui kaua arendustöö aega võtab, et vigadest õppida ja selle põhjal muudatused sisse viia, et protsessi veelgi tõhusamaks ja efektiivsemaks muuta. (Hall, s.a)

DevOps metoodikat võib vaadelda kui agiilsete tavade evolutsiooni või agiilses metoodikas puuduvat osa. Võttes kasutusele agiilse lähenemise uuendused ja rakendades neid tegevusprotsessides. Samas tuuakse välja, et see on puudulik agiilse metoodika osa, sest teatud agiilse metoodika põhimõtted realiseeruvad alles DevOps põhimõtete rakendamisel. Näiteks on agiilset metoodikat kirjeldavates dokumentides viiteid tarkvara pidevale tarnimisele, kuid kuna tärned hõlmavad ka erinevaid toiminguid, peetakse pidevat tarnimist tavaliselt DevOps lähenemiseks. (Hall, s.a)

Agiilse mudeli üks sagedamini nimetatud eeliseid kose mudeliga võrreldes on kuluefektiivsem vigade parandamine, mis on kõigi osapoolte tiheda suhtluse tulemus ning mis ühtlasi muudab võimalike probleemide ennetamise ja parandamise lihtsamaks juba arengu varajases faasis (Hoogenraad, 2018). Agiilse lähenemise korral tehakse osa planeerimisest ja projekteerimisest ette, kuid arendus toimub väikeste osadena ja hõlmab tihedat koostööd. Muudatused viiakse sisse pidevalt ja toote kasutatav versioon avaldatakse kiiremini, võrreldes kose mudelil välja töötatud toodetega (Franklin, 2016). See pakub palju eeliseid, millest kõige olulisem on, et kui tarkvara ei vasta kliendi vajadustele või ootustele, saab seda reaalajas parandada. Klassikalise kose mudeli rakendamisel läheb arendajal ja tellijal üksteisest valesti arusaamine väga kalliks. Näiteks, kui tellijale esitatakse arendusmeeskonna poolt nende mõistes valmis lahendus, mis ei vasta arenduse tellija soovile, tekivad kulukad vaidlused ning arendusega peab suures osas otsast peale alustama. (Hall, s.a)

Agiilne arendus ja DevOps on loodud väga sarnastel printsiipidel. Kui klient tellib arendusmeeskonnalt IT-lahenduse, siis projektijuht kirjeldab arenduse tarkvaraarendajale ning arenduse lõppedes esitab arendaja lahenduse projektijuhile, kes tutvustab arendust kliendile (BrainStation, s.a). Agiilsel mudelil põhinevas arenduses tutvustavad arendajad tihemini pooleliolevat tööd projektijuhile ja tellijale ning küsivad tagasisidet, et lahendus vastaks äripoole soovidele ning oleks selge, et arendaja liigub õiges suunas (Brush, Silverthorne, 2019). DevOps puhul on ametinimetusega projektijuhi asemel tooteomanik, kes vastutab toote haldamise eest kuni IT-lahenduse elutsükli lõpuni, mil kasutajatel seda enam vaja ei ole. DevOps arendusprotsessi üheski etapis ei öelda, et projekt ja tellimus on valmis, vaid arendusmeeskond täidab osaliselt ka edaspidi kasutajatoe rolli, aidates klientidel oma ootusi edasi arendada ja äriprotsesse üha enam automatiseerida. (BrainStation, s.a)

Tabel 1. Agiilse arendusmetoodika ja DevOps metoodika võrdlus (Hall, s.a)

Agiilne arendusmetoodika	DevOps arendusmetoodika
koostöö arendus- ja haldusmeeskondade vahel	hõlmab lisaks arendus- ja haldusmeeskondade koostööle ka operatsioonide/hoolduse meeskonda
koondab arenduse ideest kuni koodi lõpetamiseni	koondab arenduse ideed, sh keskendub tarnimisele ja ka hilisemale hooldusele ning vajadusel edasiarendusele
rõhub iteratiivsele arendusele ja väikestele tarnetele	keskendub lisaks väikestele tarnetele ka testimise ja tarne automatiseerimisele

Eelnevalt kirjeldatud tabel toob välja põhilised erinevused agiilses ja DevOps metoodikas. Kui agiilne metoodika keskendub arendus- ja haldusmeeskondade koostööle, siis DevOps kaasab lisaks eeltoodule ka operatsioonide meeskonna. Agiilses metoodikas on esikohal üksikisikud, suhtlus, töötav tarkvara, klientide koostöö ja muutustele reageerimine (Linchpinseo, 2023). DevOps põhimõtted on selles osas samad agiilse mõtteviisiga, kuid DevOps metoodikas laienevad arendusprotsessile lisaks süsteemide ja töötavate rakenduste hooldamine ning DevOps metoodikal on oluline rõhk meeskonnal, mitte üksikisikul. Lisaks sisaldab kaksteist *Agile Software*'i põhimõtet viiteid DevOps põhimõtetele. Näiteks rõhuasetus pideval integreerimisel, tarnimisel, väikeste tarnetega töötamine, sh sagedased väljalasked ja automatiseerimise kasutamine on kõik viidatud *Agile Software*'i kaheteistkümnes põhimõttes. (Hall, s.a)

Kokkuvõtvalt võib välja tuua, et agiilse ja DevOps'i eesmärgid on samad: parandada tarkvaraarenduse kiirust ja kvaliteeti. Paljud meeskonnad on leidnud, et agiilsed metoodikad aitavad neid tohutult, samas kui teised on näinud vaeva, et agiilse lähenemisviisi eeliseid realiseerida. Sellel võib olla mitu põhjust, sealhulgas meeskonnad, kes ei mõista agiilseid tavasid või ei rakenda neid õigesti. DevOps lähenemisviisi kaasamine võib aidata täita lüngad organisatsioonidel, kellel on probleeme agiilse metoodika rakendamisega.

1.2.1 DevOps ja agiilsete metoodikate elutsükli faaside võrdlus

Tarkvaraarenduse elutsükkel on uue tarkvara loomise või vana tarkvara muutmise protsess ning süsteemi arendamiseks kasutatavad mudelid ja meetodid. Tarkvara, kui toode on süsteemi arendusväljund. Süsteemi arendusprotsess hõlmab nii toote disainimist kui ka toote valmistamist. Tarkvaraarenduse raames toodetakse kvaliteetne tarkvara, mis vastab kasutajate soovidele ja ootustele, on valmis kokkulepitud aja ja kuluga, töötab tõhusalt ja tulemuslikult ning ei ole kulukas hooldada. Põhiliste meetodite tundmine, sobivate arendusmetoodikate ja tehnikate kasutamine muudab arendustöö efektiivsemaks, süsteemsemaks ja kuluefektiivsemaks. (EUCIP, s.a)

Kose meetod eraldab tarkvaraarenduse erinevad etapid ja tarneetapid (nt analüüs, disain, arendus ja testimine) ning viib iga etapi läbi kindlas järjestuses. Selle tulemusel saab koodi loomine alata alles siis, kui projekt on juba käimas. Olulised testimise ja kvaliteedi tagamise etapid võivad olla lühenenud või vahele jäetud, kui eelnevates etappides esineb viivitusi. Kui testimise käigus ilmnevad probleemid, tuleb tarkvara salvestada ja minna tagasi arendusprotsessi algusesse. (Yigal, s.a) Agiilse arendusprotsessi vältel tekkivate tulemite kohta tehakse otsuseid kogu protsessi vältel (nt nõuded, kavandamine, realiseerimine ja testimine toimuvad vaheldumisi). Samuti viiakse testimisi läbi

varajases faasis ning arendajad saavad probleeme lahendada ja kohandusi teha samal ajal, kui koodi luuakse. (Brush, Silverthorne, 2019)

DevOps töökultuuri rakendamisel kasutatakse pidevat integreerimist ja tarne protsessi, mis võimaldab koodi testida ning tarnete väljalasked toimuvad automatiseeritult. See annab DevOps meeskondadele pidevat tagasisidet, mis aitab vähendada vigade arvu ning uuendused saavad tellija jaoks kiiremini tehtud. (Digitark, 2023)

DevOps elutsükli kirjeldamiseks kasutatakse joonisena tavaliselt lõpmatuse märki, et näidata kuidas DevOps elutsükli faasid on seotud. Vaatamata sellele, et silmus näitab voolavat järjestikku, sümboliseerib see pideva koostöö ja täiustamise vajadust kogu elutsükli vältel. DevOps elutsükkel koosneb faasidest, mis esindavad arendamiseks vajalikke protsesse, võimalusi, tööriistu ja toiminguid. (GitLab, s.a) Näiteks võimaldab *Git* kui versioonihaldussüsteem koodi haldamist lihtsalt ja annab meeskonnaliikmetele selge ülevaate tehtud muudatustest (Digitark, 2023). Igas faasis teevad meeskonnad koostööd ja suhtlevad, et säilitada ühtlust, kiirust ja kvaliteeti (Hall, s.a).

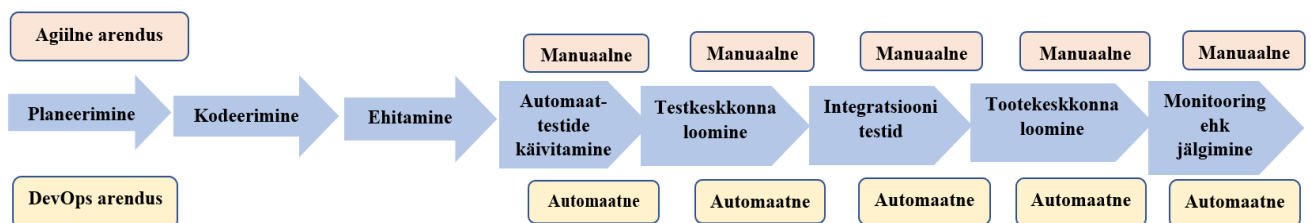
DevOps elutsükli faasid võib välja tuua järgmiselt:

- Planeerimine ja pidev areng – korraldada vajalik töö, anda sellele prioriteet ning jälgida lõpuleviimist (GitLab, s.a). Antud arenguetapis töötatakse välja eesmärgid, mida vastava toote/teenuse arendamisel rakendatakse. Kui eesmärgid on lõpule viidud, siis algab lähtekoodi loomine (Hall, s.a).
- Ehitamine/loomine – lähtekoodi loomise protsess (Hall, s.a) ehk luuakse, kujundatakse ja hallatakse koodi ja projektandmeid oma meeskonnaga (GitLab, s.a).
- Pidev integreerimine ja tarnimine – pärast arendust toimub integreerimine, kus kavandatakse mitmeid teste, mis tuleb läbi viia. Antud faas aitab välja selgitada, kas väljatöötatud rakendus vastab soovitud nõuetele (Hall, s.a). Testimine võiks toimuda automatiseeritud testimise abil (GitLab, s.a).
- Monitooring ja hoiatamine – kiire probleemide tuvastamine tootel, mis võib mõjutada toote tööaega, kiirust ja funktsionaalsust (Hall, s.a). Meeskond peab olema teadlik kõigist automaatsetest muudatustest, kõrge riskiga toimingutest või tõrgetest, jälgides mõõdikuid ja vigu (GitLab, s.a).
- Haldamine – DevOps meeskonnad haldavad kogu toote/teenuse täielikku tarnimist klientidele, mis hõlmab kogu organisatsiooni teenuseid toetava infrastruktuuri kavandamise, juurutamise,

konfigureerimise ja hooldusega seotud tavaid (Hall, s.a). Sealhulgas haldama turvariske ja vastavust organisatsioonis (GitLab, s.a).

- Pidev tagasiside - DevOps meeskonnad hindavad iga väljalaset ja koostavad aruandeid tulevaste versioonide täiustamiseks. Pidevat tagasisidet kogudes saavad meeskonnad oma protsesse täiustada ja klientide tagasisidet järgmise versiooni täiustamiseks kaasata. (Hall, s.a)

Võrdluseks eelnevale võib välja tuua, et agiilne arendusmetoodika lähtub põhimõttest - tarkvara arendatakse kihtide haaval. Eelkõige pannakse rõhku äripoolle ja arendusmeeskonna vahelisele kommunikatsioonile ning iga kokku lepitud perioodi järel näidatakse äripoolle selleks ajaks valminud tarkvaralahendust ning planeeritakse koos äripoollega järgmised arendused. (Plutora, 2021)



Joonis 2. Agiilne arendusprotsess võrreldes DevOps protsessiga (Põllumäe, 2019)

Vastavalt eeltoodud joonisele võib välja tuua, et agiilne arenduse ja DevOps arenduse metoodika elutsükli faasid on väga sarnaselt üles ehitatud. Mõlemad on seotud tarkvara arendusprotsessi täiustamisega. Agiilsed metoodikad keskenduvad protsessi esimestele sammudele ning klientide ja arendajate vahelisele suhtlusele (Hoogenraad, 2018). Seevastu DevOps metoodika protsessi viimastele sammudele ning arendajate ja administraatorite vahelisele suhtlusele. Lisaks on DevOps arendusmetoodika puhul oluline, et suur osa arenduse protsessist on automatiseeritud (Red Hat, s.a). Automaattestid käivitatakse kohe pärast kodeerimist ja edukal lõpetamisel jõuavad muudatused automaatselt testkeskkonda, kus käivitatakse integratsioonitestid. Lisaks sellele on tootmiskeskonna seadistamine ja jälgimine automatiseeritud (Plutora, 2021).

Antud alapeatükis keskenduti agiilse ja DevOps raamistike elutsükli faaside võrdlusele. Siinkohal võib välja tuua, et mõlemad metoodikad tegelevad tarkvara arenduse protsessi parendamisega ning elutsükli faasid on sarnase ülesehitusega. Erisuseks on oluline DevOps'i puhul märkida, et enamus protsessist on automatiseeritud (sh alates automaattestide käivitamisest kuni monitooringuni), mis näiteks agiilse arenduse puhul on manuaalsed.

1.2.2 DevOps eelised võrreldes teiste arendusmetoodikatega

DevOps põhimõtete uurimine on oluline, kuna järjest enam hakatud paindlikku arendust täiustama DevOps põhimõtetega. DevOps põhimõtete rakendamine organisatsioonides tagab sageli töö tõhususe suurendamise ning lühema aja jooksul saab rohkem tarkvaralahendusi.

Tabel 2. DevOps metoodika eelised (GitLab, s.a)

DevOps metoodika eelised	
meeskonna koostöö	DevOps metoodika kasutuselevõtt loob arenduse ja operatsioonimeeskondade vahel ühtekuuluvuse. Kõik on kaasatud ühiste eesmärkide saavutamisele
kiire reageerimine	meeskonna koostöö viib suurema tõhususeni. Muudatusi ja täiustusi saab kiiremini ellu viia
lühemad tsüklid arenduste tarnimiseks	suurenenud tõhusus ja tihe suhtlus meeskondade vahel lühendavad tsüklit. Uue tarne saab kiiremini välja anda, säilitades samal ajal kvaliteet ja turvalisus.

Tarkvara arendamiseks on mitmeid metoodikaid. Näiteks kose metoodikal arendus, pidev integreerimine ning iteratiivsed arendusprotsessid. Lisaks võib välja tuua, et nende hulka kuulub ka DevOps metoodika. DevOps metoodika väärtus ja kultuuri eelised seisnevad tarkvara kiires ja pidevas täiustamises (GitLab, s.a). Eelnevalt toodule on lisaks veel teisigi metoodikaid ja neil kõigil on omad eelised ja puudujäägid ehk igal metoodikal on vastav koht rakendamiseks. Selleks, et olla edukas organisatsioon, peab leidma viisi, kuidas teha võimalikult kvaliteetset tööd, olles seejuures ka võimalikult tõhus. Seda on näiteks DevOps töökultuuri abiga saavutanud Amazon, Netflix või NASA (Automate, 2018). Ühe näitena võib välja tuua, et Netflix kasutab DevOps töökultuuri, mis võimaldab neil teha päevas kuni 4000 koodimuudatust. DevOps meetodi kasutamine võimaldab video- ja muusika voogedastusteenustel pakkuda oma platvormide kasutajatele järjepidevaid uuendusi ilma pikki arendustsükleid läbimata ning samal ajal võtta arvesse ka kasutajate soove (Digitark, 2023).

Lisaks võib välja tuua Soome ülikoolides läbi viidud uuringu, mille tulemusena toodi välja DevOps praktikate juurutamisega kaasnevad tegurid:

- Rohkem juurutatud protsess ja sagedasemad tarned ehk automatiseeritud protsessid võimaldavad organisatsioonil funktsionaalsust luua. Lisaks aitab protsesside automatiseerimine vähendada tarnetele kuluvat töömahtu, võimaldades tarnida nii sageli kui tarvis.
- Kvaliteedi paranemine – arendusprotsessi automatiseerimine aitab tagada, et kõiki muudatusi testitakse enne juurutamist. Muudatusi kontrollitakse igas arendusetapis, mis tähendab, et arendatavas tarkvaras on vähem vigu ning vigade parandamine tuleb soodsam.
- Meeskonnas on paranenud koostöö ning kommunikatsioon – DevOps metoodika kasutamine hõlbustab suhtlust arendus- ja haldusmeeskondade vahel ning suurenenud on suhtlus ja koostöö, mis hõlbustab kogemuste ja teadmiste edasiandmist meeskondade vahel. Kui oskusi tuleb juurde, saavad multifunktsionaalsed autonoomsed meeskonnad oma võimalusi paremini ära kasutada.
- Vähem stressi meeskonnas – regulaarsed tarned vähendavad meeskonna stressi, erinevalt harvadest ja mahukatest tarnetest, kuna uute funktsioonide kasutuselevõtul esinevate võimalike vigade arv on väiksem.
- Äripoolle kiiremini juurutatavate arenduste kõrgem kvaliteet – arendusi on võimalik äripoolle kiiremini rakendada. Arendustiim saab koheselt tagasisidet, mis aitab toodet/teenust täiustada ning kiiresti reageerida turvanõuetele, samuti võimaldab see erinevate ideede kiiret katsetamist (Riungu-Kalliosaari, 2019).

Arvestades eeltoodut, võib järeldada, et DevOps mudeli kasutuselevõtt organisatsioonijuhtimises toob kaasa reaalselt mõõdetavaid tulemusi. Lisaks on uuringud näidanud, et DevOps põhimõtteid rakendavate organisatsioonide töö on enamasti efektiivsem. Puppet Labs koostas 2014. aastal uuringu „*2014 State of DevOps Report*“, mille raames analüüsiti üle 9200 tehnilise professionaali vastuse DevOps metoodika osas. Kokkuvõttest selgus, et mida kauem organisatsioon on rakendanud DevOps põhimõtteid, seda produktiivsemaks muutub arendusmeeskonnas tehtav töö. Lisaks paranevad äritulemused ja üldine infotehnoloogiline pool. (Labs, State of DevOps Report, 2014)

Lisaks võib välja tuua Puppet Labs'i korraldatud 2015. aasta uuringu: „*The 2015 State of DevOps Report*“, millest selgus, et suure jõudlusega ja hästi juhitud organisatsioonid, kes rakendavad DevOps põhimõtteid, tarnivad arendusi 30 korda sagedamini ja 200 korda kiiremini. Riketest taastumine toimub 168 korda kiiremini ning rikkeid on üleüldiselt 60% vähem. (Labs, State of DevOps Report, 2015) Statistilised andmed kinnitavad, et organisatsioonid, mis kasutavad DevOps metoodikat, genereerivad uusi koodi 46 korda sagedamini ning nende rakenduste uuendamisel esineb 7 korda vähem ebaõnnestumisi võrreldes traditsioonilisi meetodeid kasutavate arendustiimidega. Tööprotsesside automatiseerimine on vähendanud 71% DevOps metoodikat kasutavate

organisatsioonide manuaalse töö mahtu (Digitark, 2023). See lükkab ümber müüdi, et infotehnoloogia peab valima kiiruse ja töökindluse vahel. Kuna tõrked ja vead on vältimatud, on esiplaanile tõusnud vastavate rikete kõrvaldamise kiirus. See omakorda võib määrata, kas organisatsioon on turul edukas.

SMIT (Siseministeeriumi Infotehnoloogia- ja arenduskeskus) on välja toonud „*The State of DevOps Report*“ põhjal DevOps metoodika efektiivsusmõõdikud. Põhilised eesmärgid, mida DevOps'il mõõdetakse on kiirus, kvaliteet ning rakenduste jõudlus.

Tabel 3. Kõige sagedamini kasutatavad ja olulisemad mõõdikud (Siseministeeriumi haldusala IKT teenuse arendamise ja haldamise lõpparuanne, 2018)

Enamkasutatavad efektiivsuse näitajad	Näitaja	Kõrge efektiivsusega asutused	Keskmise ja madala efektiivsusega asutused	Väga efektiivsete organisatsioonide näitajad
koodi toodangusse rakendamine	kui sageli viib organisatsioon läbi koodi rakendamist?	viivad koodi tootmisesse vastavalt vajadusele, kuni mitmeid kordi päevas	ajavahemikus kord nädalas kuni kord kuus	tõstsid aastatel 2016-2017 koodi juurutamise sagedust umbes 46 korda
koodimuudatuste rakendamise aeg	kui palju aega kulub koodi salvestamisest kuni selle rakendamiseni tootmiskeskkonnas?	rakendamine toimub alla ühe tunni	ajavahemik nädalast kuni ühe kuuni	vähendasid muudatuste rakendamise aega umbes 440 korda
taastumisaeg pärast intsidente	kui kaua kulub teenuse taastamiseks pärast intsidenti?	taastuvad alla tunni	keskmise tõhususega asutustel vähem kui päev, madalate puhul ühest päevast kuni ühe nädalani	parandasid taastumiskiirust intsidentide järel ligikaudu 96 korda
muudatuste ebaõnnestumise protsent	kui suur protsent muudatustest tekitab rakenduse	kõige tõhusamate asutuste	keskmise tõhususega asutustel	langetasid ebaõnnestunud muudatuste

	ja/või teenuse kvaliteedi langust või nõudis parandamist?	muutuste ebaõnnestumise protsent kuni 15%	muutuste ebaõnnestumise protsent kuni 15%, 31-45% madala tõhususega asutuste puhul	protsenti ligi 5 korda
arendustöötundide ja vigade parandamise ajaline võrdlus	kui palju lisaaega tekitavad arendustöötunnid vigade parandamisele?	efektiivsusmõõdik näitab meeskonna võimet teostada kvaliteetset tööd, võimaldades meeskondade vahelist võrdlust ja tervislikku konkurentsi.		

Vastavalt eelnevalt toodud tabelile on „*The State of DevOps*“ raporti alusel 2016-2017. aastate trendi hinnates jõutud arvamusele, et madala efektiivsusega meeskonnad, kes keskenduvad üleliia kiirusele, võivad kannatada kvaliteedi all, suurendades kriitiliste intsidentide ja tõrgete riski. See omakorda nõuab märkimisväärseid jõupingutusi nende lahendamiseks ja teenuste taastamiseks. Kõrge efektiivsusega organisatsioonid mõistavad, et stabiilsuse ohverdamine kiiruse kasuks või vastupidi ei ole jätkusuutlik ja keskenduvad kvaliteetse arenduse tagamisele (Siseministeeriumi haldusala IKT teenuse arendamise ja haldamise lõpparuanne, 2018).

DevOps rakendamine tarkvaraarenduses on kasvav trend, tänu millele on võimalik kiiremini reageerida tellija vajadustele. Paljud ettevõtted püüavad paremini ära kasutada DevOps metoodikaga kaasnevat kasu. Tarkvaraarendajate jaoks on oluline mõista DevOps põhimõtteid ja tööriistu ning omaks võtta koostööle orienteeritud kultuur. DevOps põhimõtete tundmine ja sellega töötamise oskus annavad eelise nii kliendile kui ka arendajale, kuna mõlemad saavad kasu arenduse kiirusest ja kvaliteedist. (Singleton, 2023)

Eelnevaid punkte kokku võttes võib välja tuua, et DevOps'i rakendades on organisatsioonidel võimalik paremini kasutada oma arendusmeeskondade ajalisi ressursse, vältides suuri vigu arenduse lõppfaasis ning tõsta tehtud töö ajalisi väärtusi. Lisaks võib pidev testimine ja tagasiside parandada suhtlemist klientidega ja seeläbi parandada teenindust. Selleks tuleks sisse seada automatiseeritud testid ning kõigi antud teenust või toodet kasutavate osapoolte vahel peaks olema pidev ja avatud suhtlus.

1.2.3 DevOps töökultuurile üleminekut piiravad tegurid

Kui asutuses on toimumas suuremad muutused, mis mõjutavad töötajate igapäevatööd, siis võib see olla asutuste jaoks keeruline. Muudatustega võivad kaasneda eelised, mis tõstavad töö efektiivsust ning muudavad üldise töökeskkonna meeldivamaks, samas ka puudused/piirangud, mis omavad vastupidist mõju. Asutuses suuremate muudatuste tegemisel peaks arvestama töötajate kohanemisega muutuva olukorra ja töökultuuriga. Kohanemine on protsess, mille käigus inimene õpib tundma ümbritsevat. Üleminekuperiood, mille jooksul tuleb reageerida muutustele, teha otsuseid uute olukordadega toimetulekuks ja harjumiseks. Tihti toimuvad muutused iseendas, inimeste ja kolleegide vahel ning inimeste ja nende keskkonna vahel (Källo, 2017). Kohanemisvõimet on mõistetud ka kui pädevust efektiivselt muutustele vastata (Tucker, Pleban, Gunther, 2010). Kohanemine on iga integreerumisprogrammi kõige põhilisem ja hädavajalik osa ning uue arendusmetoodika rakendamisel peaks asutus sellega arvestama ja ülemineku perioodil töötajaid kohanemisel toetama.

Järgnevalt on toodud näited DevOps põhimõtetele üleminekuga kaasnevatest puudustest/piirangutest, mis põhinevad Põhja-Eesti Regionaalhaigla näitel läbi viidud bakalaureusetöö „DevOps ja selle rakendatavus Põhja-Eesti regionaalhaiglas“ tulemustel.

- Töökultuur ja arendus - töötajate kohanemine uue arenduskultuuriga ning uute meetodite kasutamine (Ojasalu, 2021).

Algselt võib töötajatel töö efektiivsus langeda, sest metoodika on uus ning vajab harjumist ja kohandamist. Kui organisatsioon otsustab DevOps`ile üle minna, ei tohiks lasta esialgsel negatiivsetel tulemustel motivatsiooni ja tahet mõjutada, sest uuringud näitavad, et DevOps`i kasutavad organisatsioonid võivad olla kordades tõhusamad ja edukamad. (Labs, State of DevOps Report, 2015) DevOps metoodika rakendamine võib osutuda efektiivseks nii üldisele töökorraldusele kui ka töötajatele individuaalselt.

- Käitus – uute meetodite õppimine ja nendega kohanemine. Lisaks võib olla asutuses vaja muuta ka rolle ja kohustusi (Ojasalu, 2021).

Uue õppimine suurendab esialgu töötajate koormust ning võib välja tuua, et kui käitus DevOps metoodika tavade järgi rohkem automatiseerida, siis see kiirendab toodete tarnimist ja juurutamist ning vähendab inimlike eksimuste võimalust. (Hall, s.a)

- Töövahendid – uute töövahenditega tutvumiseks on vaja töötajatel koolitust. Töökoormus suureneb esialgu õppimise ning uute ja tundmatute vahenditega harjumise tõttu (Ojasalu, 2021).

DevOps metoodikaga kaasneb uusi töövahendeid, mis aitavad automatiseerida arendusprotsessis olevaid ülesandeid ja protsesse, ühtlasi muutes töö lihtsamaks ja vahetumaks (Digitark, 2023). Algne töökoormus töötajatel suureneb, kuna kasutusel on uued töövahendid ja põhimõtted, millega tuleb harjuda. Samas tuleks siinkohal kindlasti näha tuleviku mõistes ressursi kokkuhoidu, näiteks toote või teenuse juurutuse aega ja sagedust, mis kulub muudatuste tegemiseks. Samuti keskmine aeg, mis kulub muudatustest taastumiseks. Need tegurid määravad suutlikkuse ja stabiilsuse. IT jõudluse parandamiseks peavad asutused keskenduma arendusele ning suurendama vastavalt suutlikkust ja stabiilsust, mida on võimalik saavutada DevOps arendusmeetodite rakendamisel. (Labs, State of DevOps Report, 2014)

- Finantseerimine - algselt on DevOps metoodikale üleminek asutuste jaoks suur investeering (Ojasalu, 2021).

DevOps põhimõtetele üleminekul võib olla on üks suurimaid probleeme finantseering. Uued töövahendid, koolitused töötajatele, uute töötajate värbamine – ja võimalikud lisakulud, millega tuleks ülemineku protsessis arvestada. Uute töövahendite hankimine tähendab, et asutus peab ostma vastava tarkvara kasutamiseks ka vajalikud litsentsid ja load. DevOps'iga kaasneb erinevaid vahendeid, mis on asutuse jaoks väljaminek, millega tuleks eraldi arvestada. Lisaks eelnevalt toodud kuludele lisandub koolitus ja juhendamine, et töötajad saaksid uute meetodite ja tööriistadega kiiremini tuttavaks. Algne rahaline kulu võib tunduda üüratu ja ebameeldiv, kuid arvestada tuleks edaspidist täiendavat ressurside kokkuhoidu. DevOps'i üks eesmärke on muuta töö tõhusaks ja kuluefektiivseks. Sellise tõhususe saavutamine kõrvaldab varem tuvastatud ressurside raiskamise probleemid. Näiteks, kui esialgne investeering tundub suur, tuleb arvestada kogu ressursi ja rahaga, mida DevOps'i rakendamise pealt edaspidi edukalt kokku hoitakse. (Ojasalu, 2021)

Kokkuvõtvalt võib eelnevalt toodu põhjal välja tuua, et uuele arengukäsitlusele üleminekuks vajavad asutused tahet, planeerimist ja rahalisi võimalusi. Eelnevalt toodud uuringu tulemused näitavad, et DevOps lähenemine muudab asutuse arendusprotsessi tõhusamaks. Näiteks saab paljusid töökultuuri samme automatiseerida, mis suurendab esialgu töötajate töökoormust. Hiljem on oluliselt lihtsam, kui paljud aeganõudvad tegevused automatiseeritakse. Arendus muutub tõhusamaks tänu sellele, et arendus jõuab klientideni kiiremini ja vähemate vigadega. Eksimused nt arenduses või halduses ning muus tekitavad töökeskkonnas lisastressi ja -pingeid. DevOps põhimõtetel töökultuur võib tagada paremat ja rahulikumat töökeskkonda.

1.3 DevOps metoodika juurutamise protsessi põhimõtted

DevOps tiimide loomisel on esimeseks suureks sammuks organisatsiooni juhtkonna mõtteviisi muutus, DevOps töökultuuri loomine, juhtide ja töötajate koolitamine. DevOps kasutab arendus- ja haldusmeeskondade vahelise koostöö hõlbustamiseks, protsesside integreerimise automatiseerimiseks ja süsteemi töökindluse tagamiseks erinevaid tehnilisi töövahendeid ja tehnikaid (Virmani, 2015). DevOps kultuur nõuab koostööd soosiva keskkonna loomist, kus arendus- ja haldusmeeskonnad saavad ühiselt väärtust luua. Selle saavutamiseks on vajalik juhtkonna toetus ja asutuse valmidus pidevaks arenguks. Efektiivsus ja kvaliteet on võtmesõnad DevOps põhimõtetele üleminekul. (Digitark, 2023)

Organisatsioonides on sageli probleemiks arendajate ja halduse töövahendite ühisosa puudumine. Tiimid kasutavad eraldiseisvaid töövahendeid- ja keskkondi, mis ei ühildu. Probleemid tekivad tavaliselt tarkvara tarnimisel, mille käigus annab arendusmeeskond arendustöö haldusmeeskonnale üle (Loukides, 2012). Veaotsing on asutuse jaoks kulukas, sest töövahendina kasutatavaid töökeskkondi ei tunta ja puudub vastav kogemus. Tekib olukord, kus iga tarne tekitab kogu meeskonnale stressi ja probleemid tarnetega on pigem reegel kui erand. (Virmani, 2015)

DevOps metoodika keskendub sellele, et valminud tarkvara jõuaks võimalikult kiiresti ja kvaliteetselt tootekeskonda ja oleks nähtav lõppkasutajale (Filip, 2020). Kui lähtekood on loodud, siis käivitatakse automatiseeritud testid ning peale testide edukat läbimist jõuavad muudatused automaatselt testkeskkonda. Lisaks on tootekeskonna seadistamine ja jälgimine automatiseeritud (Automate, 2018). Eelkirjeldatud protsessi automatiseerimiseks on organisatsioonis vaja muuta seni asutuses kehtinud põhimõtteid, kuidas tarkvara luua, testida ja hallata (Virmani, 2015).

Tabel 4. Esmased põhimõtted, millest peaks DevOps metoodika juurutamisel tehnilises vaates alustama (Education Wiki, s.a)

põhimõtted DevOps metoodika juurutamiseks	rakendamine
pidev tarkvara integreerimine	arendajad panevad iga päev koodid jagatud koodihoidlasse, mis on hea arendustava
pidev tarne	tarkvara peab olema võimalik kogu aeg tarnida
pidev juurutamine	kõik muudatused peaksid läbima alati automatiseeritud testid

töövoo kaardistamine	asutusel peaks kasutusel olema tööriist, mis aitab kuvada kogu teavet toimimise kohta, sh kvaliteet ja aeg
piirangute teooria	meeskond peab selgitama välja kõige piiravamad tegurid, et neid siis järk-järgult parandada ja korrigeerida kuniks need ei ole enam piiravad tegurid, et töökvaliteeti järjest parandada

Vastavalt eelnevalt toodud loetelule eeldab DevOps põhimõtetele üleminek, et kasutusele on võetud ka sobiv automatiseeritud töövoog, mida kasutatakse peale iga muudatuse tegemist ja mille eesmärk on liigutada muudatus arendaja keskkonnast test- ja tootekeskonda (Põllumäe, 2019). Pideva integreerimise korral tuleks kõik erinevate arendajate tehtud koodimuudatused mitu korda päevas liita kesksesse või jagatud koodihoidlasse. Koodimuudatusi tuleks lisada järk-järgult ja testida kohe automaatsetestidega. See võimaldab avastada ja parandada vigu varajases arengujärgus. Pidev tarnimine on järgmine samm pideva integreerimise protsessis, olles ühtlasi üks DevOps'i keskseid põhimõtteid. (Rush, 2016) Pideva juurutamise eelduseks on, et arendatav toode läbib automaatse testimise, kontrollimaks, kas toode on ootuspärane ja piisavalt turvaline. Automaattestid on DevOps meetodi puhul väga olulised, kuna arvutite võimekus kiiresti ja ulatuslikult teste läbi viia võimaldab avastada arendusprotsessis esinevad probleemid kiiremini, mille tulemusel saab arendustiim kohe reageerida, tuvastatud probleemidele tähelepanu pöörata ning need kiiresti lahendada. (Hall, s.a)

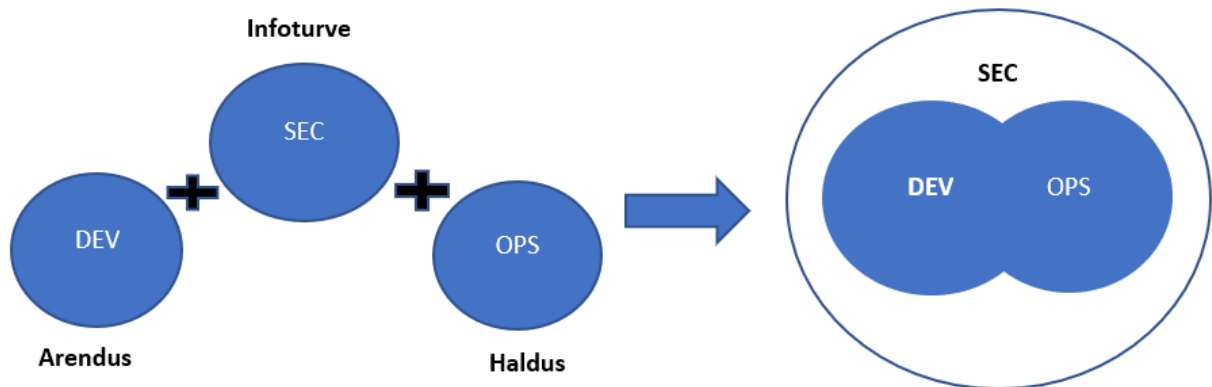
Töövooge on võimalik teostada erinevalt, kuid seni on kõige edukamad olnud näiteks Docker konteinerite lahendused, kus kasutatakse ka mikroteenusarhitektuuri (Põllumäe, 2019). Docker on virtualiseerimistehnoloogia, mis võimaldab lisada rakenduse ja selle sõltuvused konteinerisse, mida saab hõlpsasti juurutada erinevatele platvormidele, alates arendajate arvutitest kuni organisatsiooni serverite ja pilveteenuse pakkujateni (Docker Inc., 2019). Väga paljud süsteemid on loodud selliselt, et üks rakendus täidab paljusid funktsioone, samas on süsteemil vähe arendajaid, siinkohal on üks DevOps kultuuri osast mikroteenusarhitektuur. Mikroteenused on sõltumatud, autonoomsed, kiiresti arendatavad rakendused, mis täidavad ühte kasulikku funktsiooni (Parmakson, 2018). Sõltumatus ja autonoomia tähendab, et seda ei ole vaja arendada samaaegselt ülejäänud süsteemiga ning see toetub võimalikult vähe muudele mikroteenustele, tehnoloogiatele ja raamistikele. Kiire areng tähendab mitte rohkem kui ühte nädalat teenuse muutmiseks või uuendamiseks. See funktsioon on oluline, kuna tagab sõltumatuse programmeerimiskeeltest ja juurutustest ning olemasolevaid komponente saab vajadusel eemaldada ja uuesti luua. (Parmakson, 2018)

Kokkuvõtvalt võib välja tuua, et DevOps metoodika juurutamiseks on olulised organisatsiooni juhtkonna esmased sammud töökultuuri ja mõtteviisi loomise osas. Tarkvaraarenduse ja IT-operatsioonide meeskondade ühendamine eesmärgiga parandada koostööd ja varustada neid asjakohaste tööriistadega on DevOps'i lähenemise tuum. See võimaldab luua kiirema ja kvaliteetsema arendusprotsessi. DevOps'i põhimõtetele üleminekul on kesksel kohal tõhusus ja kvaliteet. Alustades töökultuuri loomisest, töötajate koolitamisest, tööprotsesside loomisest ning vastavalt tööprotsessidele ka õigete töövahendite tagamine.

1.4 DevOps metoodika täiustamine DevSecOps põhimõtetega

Kui organisatsioonid võtavad kasutusele DevOps põhimõtted, siis jääb sageli turvalisuse aspekt tagaplaanile. Uuringud näitavad, et infoturbe integreerimine toimub ainult 20% juhtudel, kui organisatsioonid liiguvad üle DevOps printsiipidele. Tihti peetakse infoturvet agiilsuse ja arendustempo aeglustavaks teguriks (Myrbakken, Colomo-Palacios, 2017). Samal ajal on DevOps'i rakendavates organisatsioonides tehtud uuringute põhjal üle poole juhtidest veendunud, et infoturbe on arendusprotsessi lõimitud, kuid arendajate seas jagab seda arvamust vaid veidi rohkem kui kolmandik (Koskinen, 2020).

DevOps ja DevSecOps (*Development, Security and Operations*) on mõlemad tarkvaraarenduse metoodikad, mis keskenduvad koostööle ja automatiseerimisele tarkvaraarenduse ja tarnimise protsessis, kuid nende peamine erinevus seisneb turvalisuse kaasamises ja rõhuasetustes (Alvarenga, 2022). DevSecOps on turvalisuse integreerimine kiirelt arenevasse IT ja DevOps arenduse kultuuri, tehes seda võimalikult sujuvalt ja läbipaistvalt. Siinkohal tuleks arvestada, et arendajate töötamise kiirus ei väheneks ja arenduse töös ei tekiks ebamõistlikke katkestusi (Gartner, 2019). IT-sektori konkurentsivõimes püsimiseks peavad infotehnoloogia asutused olema võimelised kiiresti arenguga kaasas käima, sealhulgas arendama DevOps metoodikat DevSecOps suunas, kus turvalisus on integreeritud juba arendusprotsessi algfaasides ja see on läbivalt keskseks elemendiks kogu tarkvaraarenduse elutsükli vältel (Uptime, s.a).



Joonis 3. DevSecOps töökorraldus (autori koostatud)

Joonisel 3 on välja toodud DevSecOps olemus. DevSecOps metoodika on DevOps metoodika laiendus, mille eesmärk on integreerida infoturbe DevOps tarkvaraarenduse tsükklisse, edendades koostööd infoturbe, arenduse ja halduse vahel (Department of Defence, 2021).

DevSecOps metoodika alusel tuuakse turvalisuse teemad arendusprotsessi juba selle algfaasis. Kui eesmärk on integreerida infoturbe eesmärgid võimalikult varajases etapis, peab see olema võimalikult valutu. Turvatiimide ja eesmärkide integreerimine ei tohiks olla arendajate kohustus. Organisatsioonid, kellel on ühine vastutus arenduse ja operatsioonide eest, suudavad iteratsioone teha kiiremini ja seetõttu on nad edukamad. DevSecOps peaks olema mõeldud kui loomulik jätk DevOps`ile, mitte eraldi idee või kontseptsioonina. (Snyk, s.a) Kui DevOps põhimõttele liikumine on toonud arendajad, halduse ja äritiimi kokku, hõlbustades agiilset arendust, pidevat testimist ja oluliselt tõhusamat protsessi, siis turvalisuse aspektid jäid sageli tähelepanuta. Tihti käsitleti neid alles testimisfaasis või siis, kui arendusetapp oli valmis klientidele üle andmiseks. DevSecOps lähenemine aitab seda olukorda muuta, pakkudes nii metoodikat kui ka laiemat mõtteviisi ja arenduspõhimõtet. (Uptime, s.a). Tiimid, kes on edukad DevOps tavade rakendamisel, peaksid mõtlema DevSecOps`i kui evolutsioonilist sammu (Snyk, s.a).

Kokkuvõtvalt võib välja tuua et DevSecOps lisab turvalisuse aspekti DevOps mudelisse, muutes see turvalisemaks ja vähendades turvariskide võimalusi tarkvaraarenduse ja tarnimise protsessis. DevSecOps on töökorraldus, mis põhineb agiilsetel printsiipidel ja oma olemuselt on DevOps töökorralduse edasiarendus. DevSecOps eesmärk on ühendada DevOps infoturbega, läbi selle arendada äritulemusi ning maandada infoturbest tulenevaid riske.

2 EMPIIRILISE UURINGU METOODIKA

Vastavalt Rahandusministeeriumi suunistele on riigi infotehnoloogia asutuste suund liikuda lõimitud tiimide praktika ja agiilsemate arendusmetoodikate suunas, kus on oluline, et arenduste tulemused luuakse kiiresti ning arenduste kasutuselevõtt oleks agiilsem (IKT arenduste toimemudel, 2023). Uurimisprobleem tuleneb Eesti riigi infotehnoloogia asutuste vajadusest töömeetodeid uuendada ja arendusprotsessi automatiseerida rakendades DevOps põhimõtteid.

Empiirilise uuringuga soovitakse kaardistada nelja (4) Eesti riigi infotehnoloogia asutuse näitel DevOps põhimõtetele üleminekut ja arendusmetoodika muutmist tinginud tegureid. Eesmärgi täitmiseks on püstitatud järgmised ülesanded:

- valida sobiv metoodika ja luua uurimisinstrument empiirilise uuringu läbiviimiseks;
- kaardistada asutuste DevOps'ile üleminekut, muutmist tinginud tegureid ja uue metoodika efektiivsust võrreldes varem kasutatud arendusmetoodikaga;
- saadud tulemuste põhjal teha järeldusi ja ettepanekuid, mis võiksid olla abiks teistele infotehnoloogia asutustele DevOps põhimõtetele üleminekul.

Lõputöö ülesehitus lähtub deduktiivsest lähenemisviisist, kus toetutakse varasemalt töös käsitletud teooriale. Andmeid kogutakse kvalitatiivsel andmete kogumise meetodil. Kvalitatiivse andmete kogumise meetodi valik tuleneb soovist uurida loomulikes tingimustes, sh keskkonnaga manipuleerimata (Õunapuu, 2014). Kvalitatiivse meetodi eeliseks on võimalus keskenduda harva esinevatele ja unikaalsetele nähtustele, kuid samas puudub tihti võimalus andmeid millegagi võrrelda. See teeb üldistamise keeruliseks – sellisel juhul piirduakse mõne osa esiletõstmisega (Kalmus, 2015).

Uurimisstrateegiaks on kaardistusuuring (*survey*). Eelmainitud uurimisstrateegia oma olemuselt on uurimise viis, mille abil selgitatakse tegelikke tingimusi ja asjaolusid (Hirsjärvi, Huttunen, 2005). Uurimisinstrumendina kasutati uuringu läbiviimisel poolstruktureeritud intervjuud. Intervjuu on autori koostatud (vaata Lisa 1). Tulenevalt empiirilise uuringu eesmärgist on kõige sobilikum andmete kogumise meetodina rakendada poolstruktureeritud intervjuud. Intervjuu kasuks antud uuringu läbiviimisel osutusid intervjuu eelised: vahetu kontakt küsitletavaga ning vajadusel võib intervjuueerija jagada arusaamatuse puhul lisaselgitusi (Kuusik, Virk, Aarna, 2010). Intervjuu küsimused on koostatud teemade kaupa lähtudes lõputöö eesmärgist: DevOps metoodika määratlemine ja arendusmetoodika muutmist tinginud tegurid, DevOps põhimõtete võrdlus ja eelised teiste tarkvaraarenduse metoodikatega, DevOps mõju asutustele ning meeskondade tagasiside, DevOps põhimõtetele üleminek ja juurutamine.

Lõputöö uurimisobjektideks on neli (4) Eesti riigi infotehnoloogia asutuste DevOps põhimõtetele ülemineku ja/või juurutamise protsessi eest vastutav töötaja või töötajad, kes on asutuses DevOps põhimõtetele üleminekuga kõige rohkem kursis. Uuring viidi läbi 2023. aasta oktoobris. Valimina kasutati eesmärgistatud valimi põhimõtet (*purposive sampling*) (Õunapuu, 2014), sest see võimaldab koostada valimit vastavalt eesmärgile, st uuritavad valitakse valimisse mingi kriteeriumi alusel, millele kõik uuritavad vastavad. Eesti riigi infotehnoloogia asutuste hulgast valiti välja viis (5) asutust – Riigi Infosüsteemide Amet (RIA), Tervise ja Heaolu Infosüsteemide Keskus (TEHIK), Registrate ja Infosüsteemide Keskus (RIK), Rahandusministeeriumi Infotehnoloogiakeskus (RMIT), Siseministeeriumi Infotehnoloogia- ja arenduskeskus (SMIT). Antud asutustega kontakteeruti e-posti teel, sooviga viia läbi intervjuu asutuse DevOps põhimõtetele ülemineku osas. Esialgselt valimist anti tagasisidet viie (5) asutuse poolt, kuid lõplikuks valimiks jäid neli (4) asutust, keda esindasid asutuse poolt kas *Agile Coach* (välearenduse ekspert), arhitekt või mõnel muul ametikohal olevad töötajad, kes on seotud DevOps põhimõtetele ülemineku ja/või juurutamise protsessiga asutuses.

Uuringu läbiviimiseks võeti ühendust asutuste poolt suunatud isikutega e-posti teel, et leppida kokku intervjuu läbiviimise aeg. Intervjueeritavatele tutvustati intervjuu teemat, intervjuule kuluvat orienteeruvat aega ning saadeti ka küsimused eelnevalt tutvumiseks ja vastuste ettevalmistamiseks. Intervjuud viidi läbi *Microsoft Teams* keskkonnas.

Järgnevalt koostati tabel läbiviidud intervjuude kuupäevade ja kestvusega. Tabelis on toodud tähed, mida kasutatakse edaspidi asutuse tähisena.

Tabel 5. Läbiviidud intervjuude andmed

asutuse tähis	intervjueerimise kuupäev	intervjuu kestvus
A	02.10.2023	32min 56sek
B	06.10.2023	1h 8min 31sek
C	12.10.2023	40min 27sek
D	13.10.2023	35min 8sek

Kvalitatiivse sisuanalüüsi käigus analüüsiti nelja (4) poolstruktureeritud intervjuud. Läbiviidud intervjuude salvestusele anti kirjalik kuju, ehk transkribeeriti. Transkribeerimiseks kasutati veebipõhist kõnetuvastusprogrammi. Saadud tekst kontrolliti üle ning korrigeeriti, et saadud

kirjalikus tekstis ei oleks informatsiooni kaduma läinud. Transkribeeritud tekstide kogumaht oli 40 lehekülge. Transkriptsiooni teel saadud andmed loeti mitmeid kordi üle ning märgistati uurimisülesannetega kooskõlas olevad tekstilõigud. Analüüsi illustreeriti intervjuude tsitaatide ja tabelitega.

Uuringu läbiviimisel lähtuti uurimiseetikast, et kõik uuringus osalenud asutused osalesid uuringus vabal tahtel. Konfidentsiaalsuse huvides ei avaldata analüüsis uuritavate asutuste ega asutusi esindavate intervjuueeritavate nimesid ega vihjata asutustele, millest võiks teha järeldusi, soovides kaitsta asutuste mainet, huve ning poliitikat.

3 EESTI RIIGI INFOTEHNOLOOGIA ASUTUSTE DEVOPS PÕHIMÕTETE ÜLEMINEKU UURINGU TULEMUSED

Andmeanalüüsi tulemused esitati intervjuu küsimuste teemaplokkide kaupa. Tulemuste kinnitamiseks kasutati intervjuudest pärinevaid tsitaate, analüüsiti saadud informatsiooni ning tehti järeldusi. Vastajate tsitaadid esitati kursiivkirjas. Välja toodud laused ning sõnad esitati järgnevalt: „/.../“. Kui sama mõtet kirjeldas mitu intervjuueeritavat, siis võeti see kokku ühe tsitaadina ning märgiti sulgudesse nimetamiste arv.

3.1 DevOps metoodika määratlemine ja arendusmetoodika muutmist tinginud tegurid riigi infotehnoloogia asutustes

Lähtudes teoreetilise osa alapeatükist 1.1 DevOps metoodika olemusest ja eesmärgist, uuris lõputöö autor asutuste definitsiooni mõistele „DevOps“. Intervjuueeritavate vastustest selgus, et DevOps olemust ja eesmärki sõnastatakse eelkõige kui arenduse ja halduse koostöös toimuvat sujuvat ja võimalikult automatiseeritud töövoogu „/.../DevOps`i puhul on oluline, et töövoog oleks väga hästi paigas (1), kogu protsess oleks kiirendatud (4) ehk arendusest kuni toodanguni läheksid arendused kiiresti, sujuvalt ja automatiseeritult (4)/.../“. Erisusena võib välja tuua kaks seisukohta, mis lisaks eeltodule välja toodi. Esimeses toodi välja ka vastutuse olulisus DevOps tiimides „/.../Põhimärksõna on DevOps`il ka vastutus, ehk siis minu tiimi teenused on need, mille eest mina naha ja karvadega vastutust tunnen/.../“. Teiseks toodi DevOps definitsiooni ka Scrum mõiste „/.../DevOps on üks samm edasi Scrum lähenemisest/.../“. Alapeatükis 1.2 on toodud, et Scrum ja DevOps on kaks üksteist täiendavat metoodikat, mida saab tarkvaraarenduse täiustamiseks koos kasutada. Integreerides DevOps praktikad Scrum protsessi, saavad Scrum meeskonnad DevOps meeskondadega tõhusamalt töötada.

Vastavalt alapeatükile 1.1 koondab DevOps tarkvaraarenduse metoodikad ja põhimõtted, mis võivad erinevates käsitlustes kohati erineda. DevOps (*Development and Operations*) on tarkvaraarenduse metoodikate kogum, mis on loodud tarkvaraarenduse ja tarkvara toimingute integreerimiseks. DevOps rakendamise eesmärk hõlmab kõiki tarkvara loomisega seotud etappe – sagedased juurutamised, lühem aeg tarkvaraversioonide avalikustamisel, väiksem vigade võimalus uuel avalikustatud tarkvaraversioonil ning lühem aeg vigade parandamisel.

Intervjuu käigus uuriti asutuste peamisi tegureid, mis tingisid arendusmetoodika muutmise. Vastustest lähtuvalt on koostatud tabel, kus on välja toodud uuringu käigus selgunud peamised arendusmetoodika muutmist tinginud tegurid ning asutuste arv, kes antud tegurit välja on toonud.

Tabel 6. Uuritavate asutuste arendusmetoodika muutmist tinginud tegurid

arendusmetoodika muutmist tinginud tegurid	asutuste arv
arenduse, halduse ja äri koostöö edendamine	4
„silotornide“ lõhkumine struktuuris	3
teenuste toimivus	2
väline surve (kiiremad ja kvaliteetsemad teenused)	2
kliendikeskus	2
vastutuste määratlemine tiimides	1

Alapeatükis 1.1 toodi välja, et eelkõige tähendab DevOps tihedat koostööd tarkvaraarenduse spetsialistide ja teiste IT-spetsialistide vahel. Tabelis 6 on näha, et 4 vastajat 4-st on toonud arendusmetoodika muutmist tinginud teguriks arenduse, halduse ja äri koostöö edendamise „/.../Kõik mida asutus oma teenusena pakub peab olema võimalikult kiire, efektiivne ja võimalikult suure väärtusega /.../“. Vastavalt alapeatükile 1.1 pannakse DevOps mõtteviisil suurt rõhku töökultuurile, mille tulemusel paraneb arenduse ja halduse koostöö, et muuta töökultuur kiiremaks ja efektiivsemaks. Teoreetilises uuringus selgus, et on oluline teha asutuse struktuuris muudatus, et arendus ja haldus oleksid ühes meeskonnas, ühiste eesmärkide ja vastutusega.

Tabelis 6 on näha, et 3 vastajat 4-st tõid välja „silotornide“ lõhkumise struktuuris „/.../Suures pildis oli tegemist silotornide lõhkumisega, mille mõte oli haldus ja äri kokku panna, et kiirendada kogu protsessi/.../“. Vähem toodi välja teenuste toimivust (2), välist survet (2), kliendikesksust (2) ja vastutuste määratlemist tiimides (1) „/.../Ühelt poolt on klient nõudlikumaks muutunud ning teiseks on väline surve asju teistmoodi teha/.../“. Alapeatükis 1.1 põhjal selgus, et DevOps töökultuuris on vastutuste määratlemine DevOps tiimides oluline, sest DevOps tiimides ei vastuta üksik indiviid vaid kogu meeskond, kes on ühise eesmärgi nimel töötanud.

Intervjueeritavatelt uuriti, et mis arengufaasis on asutus, mida nad esindavad (ülemineku, juurutamise või rakendamise faasis). Vastustest lähtuvalt on koostatud tabel, kus on välja toodud uuringu käigus selgunud asutuste arengufaasid DevOps metoodika osas ning asutuste arv ja tähis, kes on antud faasis.

Tabel 7. Uuritavate asutuste DevOps metoodika arengufaasid

asutuse tähis	asutuse arengufaas DevOps metoodika osas	asutuste arv
A, C	ülemineku faas	2
B	juurutamise faas	1
D	rakendamise faas	1

Tabel 7 alusel võib välja tuua, et 50% asutustest (A ja C) on ülemineku faasis, 25% juurutamise (D) ning 25% (B) rakendamise faasis. Antud vastustest lähtuvalt saab ülevaate asutuste arengufaaside osas, mis annavad sisulisema ülevaate edasise analüüsi tõlgendamiseks.

Esimeses intervjuu teemaplokis, mis käsitles uuritavate asutuste arusaama „DevOps“ olemusest, eesmärgist ja asutuse arendusmetoodika muutmist tinginud teguritest tõid 4 asutust 4-st välja DevOps metoodika mõistes olulised märksõnad: kiirus, efektiivsus ja automatiseeritud töövoog, mis vastavad teoreetilises osas toodud lähtekohtadele. Ühtlasi toodi asutuste poolt välja arendusmetoodika muutmist tinginud teguriteks teoreetilistest lähtekohtadest selgunud seisukohad, mida DevOps metoodikat rakendades on võimalik asutuses parendada, näiteks arenduse, halduse ja äri koostöö edendamine ning struktuurimuudatus, kus ühe valdkonna eest vastutavad töötajad on ühes meeskonnas, mis aitavad parendada ka teisi välja toodud tegureid nagu teenuste toimivus, kliendikesksus ning vastutuste määratlemine tiimides.

Järeldused seoses DevOps metoodika määratlemise ja arendusmetoodika muutmist tinginud teguritega:

1. Koondatud vastustest võib järeldada, et asutused teadvustavad „DevOps“ mõistet kui arenduse ja halduse koostöös toimuvat sujuvat ja võimalikult automatiseeritud töövoogu.
2. Ühtlasi võib järeldada asutuste poolt toodud arendusmetoodika muutmist tinginud teguritest, et arendusmetoodika muutmise ja DevOps metoodikale ülemineku võib tuua asutustele kasutajakesksema arenduskäsitluse, hõlbustada suhtlust arenduse, halduse ja äri vahel, mis lisaks hõlbustab kogemuste ning teadmiste edasiandmist meeskondade vahel.

3.2 DevOps põhimõtete võrdlus ning eelised teiste tarkvaraarenduse metoodikatega riigi infotehnoloogia asutustes

Lähtuvalt teoreetilisest alapeatükist 1.2 on tarkvaraarenduse tavad aastate jooksul oluliselt muutunud –klassikalisest kosemudelist on liigutud iteratiivsema, paindlikuma ning agiilsema lähenemise suunas. Sellest lähtuvalt uuris autor asutustes eelnevalt kasutusel olnud arendusmetoodika osas. Vastuste põhjal on koostatud tabel, kus on välja toodud uuringu käigus selgunud peamised tegurid, mis endise arendusmetoodika osas välja toodi ning asutuste arv, kes antud tegurit välja on toonud.

Tabel 8. Uuritavate asutuste senine arendusmetoodika

endine arendusmetoodika asutuses	asutuste arv
teenusepõhisuse puudumine ehk tiimil puudus ühtne vastutus kindla teenuse eest	3
suured tiimid või osakonnad, kus spetsialiseeruti kindlatele valdkondadele	2
tiimi siseselt vastutuste määratlemine	2

Tabelis 8 on näha, et endise lähenemise juures puudusid asutustes ühe kindla teenuse eest vastutavad tiimid – rollid, kes töötavad ühe projekti raames olid erinevates osakondades „/.../Asutuses puudus kindel kooslus, kes ühe kindla teenuse eest vastutuse võtaks ehk puudus teenusepõhisus (3), olid suuremad tiimid või isegi osakonnad, kus toimus spetsialiseerumine kindlatele valdkondadele (2), ja vastutuste määratlemine (2)/.../“. Alapeatükist 1.2 toodi välja, et DevOps põhimõtete järgi töötavad arendajad ja haldus kõrvuti kogu rakenduste arendamise, juurutamise ja haldamise protsessi vältel, mis tähendab olulist rollide ja vastutuste määratlemist struktuuris.

DevOps metoodika rakendamine on suuresti seotud asutuse mõtteviisi muutusega, et teha muudatused struktuuris, rollides ning töökorralduses, mis selgus ka intervjuu raames „/.../Teenusepõhiste DevOps tiimide loomine on kindlasti üks suur muudatus, mida enne ei olnud ehk arendajad, kes meil majas on, olid eelnevalt kõik ühes osakonnas - arendajate osakonnas, nüüd on arendajad erinevates teenusepõhistest tiimides, mis omakorda tagab, et kindlal tiimil on vastutus väiksema tüki eest/.../“.

Intervjuu käigus uuriti DevOps metoodika eeliseid võrreldes asutuses eelnevalt kasutusel olnud metoodikaga. Vastustest lähtuvalt on koostatud tabel, kus on välja toodud uuringu käigus selgunud peamised DevOps metoodika eelised võrreldes asutuses eelnevalt kasutusel olnud metoodikaga ning asutuste tähis ja arv, kui palju on antud eeliseid välja toodud.

Tabel 9. DevOps metoodika eelised võrreldes asutuses seni kasutusel olnud metoodikaga

asutuse tähis	DevOps metoodika eelised	asutuste arv
A, C	efektiivsem kommunikatsioon meeskonnas	2
B, C	automaatika täiustamine	2
B	taristu automatiseerimine	1
B	uuenduslikud töövahendid (DevOps tööriistade kasutuselevõtt)	1

Alapeatükist 1.2 toodi välja, et agiilne ja DevOps metoodika põhinevad väga sarnastel põhimõtetel. Ühtlasi võib välja tuua, et DevOps metoodika täiendab agiilset arendust, juhtides tähelepanu IT-

haldusosakonna rollile, mis selgus ka asutuse A intervjuust „*.../Kui teeme koostööd välise arenduspartneritega, siis on suhtlus oluliselt lihtsam, kui arenduspartner suhtleb tooteomaniku ja haldusega ühes tiimis. Kõik osapooled teavad, mis seisus arendused on ja haldusele ei tule üllatusena, kui tulevad mingid suuremad tärned/.../*“. DevOps tõstab esile agiilse infrastruktuuri ehk koos tellija vajaduste agiilse rakendamisega tarkvarakoodis peab toimima ka infrastruktuuri paindlik haldus.

Tabelist 9 selgub, et lisaks toodi asutuste B ja C poolt DevOps eelisena tööloikude automatiseerimine ja täiustamine ning uuenduslikumad töövahendid „*.../Automaatika täiustamine (2), taristu automatiseerimine, DevOps tööriistade kasutuselevõtt (Anisble) jne on muutnud arendusprotsessi oluliselt kiiremaks ja kvaliteetsemaks/.../*“. DevOps sõltub suurel määral töövahendite ning tööloikude automatiseeritusest. Vastavalt alapeatükile 1.2.1 on DevOps puhul enamus protsessist automatiseeritud (sh alates automaatsete käivitamisest kuni monitooringuni). DevOps arendusprotsessis automaatselt käivitatakse kohe pärast kodeerimist ja edukal lõpetamisel jõuavad muudatused automaatselt testkeskkonda, kus käivitatakse integratsioonitestid. Lisaks sellele on tootmiskeskonna seadistamine ja jälgimine automatiseeritud.

Tabel 9 põhjal selgub lisaks, et asutused A ja C töid olulise eelisena välja ka kommunikatsiooni, mis on asutuses paranenud „*.../Kommunikatsioon on DevOps'i puhul oluline märksõna, kommunikatsiooni paranemine on kindlasti üks positiivne pool, mis DevOps metoodikaga kaasneb/.../*“. Kui varem olid arendus ja haldus eraldi, võis see kaasa tuua probleeme suhtluses, segadust, ajakava ületamist ning muid tarkvara arendusprotsessi takistusi, mis tulenesid meeskondade vahelisest piiratud koostööst. Nagu ka alapeatükis 1.2.2 kirjeldati, siis DevOps metoodika abil ühendatakse arendus- ja haldustiimid ühte meeskonda, kus on ühine ja selge eesmärk: saavutada tõhus ja kvaliteetne tarkvara arendamine ning sujuv käivitamine.

Intervjueeritavatel uuriti DevOps töökultuurile üleminekul ja juurutamisel tekkinud õppetunde ja edusamme. Vastustest lähtuvalt on koostatud tabel, kus on välja toodud uuringu käigus selgunud peamised DevOps metoodikale ülemineku ja juurutamisega seotud õppetunnid ja edusammud.

Tabel 10. Asutuste DevOps töökultuurile ülemineku ja juurutamise õppetunnid ja edusammud

õppetunnid DevOps töökultuurile üleminekul/juurutamisel	edusammud DevOps töökultuurile üleminekul/juurutamisel
kogu organisatsioonis ühtse DevOps töökultuuri ja mõtteviisi tekitamine	arendusprotsessi väljatöötamine ja tarkvaratööriistade komplekti <i>Ansible</i> kasutuselevõtt asutuses
struktuurimuudatusega rollide lahti mõtestamine ja suure pildi loomine	näidiste ja juhendmaterjalide koostamine seoses DevOps tööprotsesside ja -vahendite kasutuselevõtiga
lisaks meeskonnasisesele kommunikatsioonile tuleks toimima saada ka meeskondade vaheline kommunikatsioon asutuses, et ei tekiks arenduste kordusi erinevates tiimides	meeskonnad tunnevad end kindlamalt ja koostöö on oluliselt paranenud
	töökindlamad ja stabiilsemad süsteemid

Lähtudes DevOps teoreetilistest lähtekohtadest, on DevOps tiimide loomisel esimeseks suureks sammuks organisatsiooni juhtkonna mõtteviisi muutus, millel on oluline roll kogu organisatsiooni töökultuuri muutumisel. Tabelist 10 selgub ühtse töökultuuri ja mõtteviisi tekitamise osas õppetund, mida toodi välja asutuse A intervjuu käigus „*.../Kogu organisatsiooni töökultuur väga hästi ei toeta vastutuse võtmist ja ei ole tekkinud ühtset tiimitunnet/mõtteviisi ning esineb endiselt silotorne, kus üksteisele antakse ülesandeid, mitte ei proovita ühtse tiimina lahendada/.../*“. Siinkohal võib välja tuua, et DevOps metoodikale ülemineku etapis võib töötajatel töö efektiivsus langeda, sest metoodika on uus ning vajab harjumist ja kohandamist.

Tabelist 10 selgub õppetund, mis toodi välja asutuse C poolt: struktuurimuudatuste planeerimisel ning selle mõtestatus ja suurema pildi loomine „*.../Struktuurimuudatusega seoses on osad rollid lõpuni lahti mõtestamata ning kohati on loodud liiga suured tiimid ehk ühe tiimijuhi all 30 inimest on DevOps tiimina palju/.../*“. Nagu alapeatükis 1.2 on välja toodud, et IBM soovitusel on DevOps meeskonna soovituslik maksimaalne suurus 10 inimest, mis annab võimaluse kõigil kaasa lüüa meeskonna põhifunktsioonide juhtimises.

Alapeatükist 1.2.3 kirjeldati, et kui organisatsioon otsustab DevOps`ile üle minna, ei tohiks lasta esialgsel negatiivsetel tulemustel motivatsiooni ja tahet mõjutada, sest uuringud näitavad, et DevOps metoodikat kasutavad organisatsioonid võivad olla kordades tõhusamad ja edukamad ning

DevOps rakendamine asutuses võib osutuda efektiivseks nii üldisele töökorraldusele kui ka töötajatele individuaalselt.

Lisaks on tabelis 10 näha, et kui asutuses on loodud DevOps tiimid ning paika on saadud ka tiimide sisesed vastutused ja töökorraldus, siis on tekkinud teistpidi probleemid, et töötatakse tiimide siseselt ning tiimide vahel on info jagamine jäänud tagaplaanile, mis selgus asutuse D intervjuust „*.../Meil on tekkinud majasiseselt arendustes kordused ehk on meil on erinevad DevOps meeskonnad, erineva kliendi jaoks, kes arendavad täpselt sama funktsionaalsust nullist peale, mis ei ole tuleviku mõistes jätkusuutlik ning selle teemaga tuleb tegeleda, näiteks kui üks meeskond hakkab arendama mingit uut asja, siis oleks kindel koht, kust saab vaadata, et kas selliseid asju on majas tehtud ja kas on mingi võimalus võtta see kood ja seda taaskasutada/.../*“. DevOps teoreetiliste lähtekohtades kirjeldatakse palju tiimide sisest koostööd, põhimõtteid ning töökorraldust aga vähem tuuakse välja organisatsiooni üldist töökorraldust ja põhimõtteid.

Asutuse edusammudena on tabelist 10 näha arendusprotsessi väljatöötamine, tarkvaratööriistade komplekti kasutuselevõtt ning näidiste ja juhendmaterjalide koostamise olulisus, mis selgusid asutuse B intervjuust „*.../Olulised sammud, mida kindlasti välja tooksin olid arendusprotsessi väljatöötamine ja Ansible kasutuselevõtt. Toon ka kõigile teistele välja, kes on DevOps metoodikale üle minemas ja seda protsessi juhtimas, siis üks olulisi asju on kõige kohta teha näidiseid ja juhendeid, mida suunata asutust kasutama/.../*“. Vastavalt alapeatükile 1.2.3 kulub algselt töötajatel palju ressursi uute juhendite ja töövahendite õppimisele. Uuele arengukäsitlusele üleminekuks vajavad asutused tahet ja planeerimist. Näiteks saab paljusid töökultuuri samme automatiseerida, mis suurendab esialgu töötajate töökoormust. Hiljem on lihtsam, kuna paljud aeganõudvad tegevused on automatiseeritud.

Tabelis 10 on toodud olulise edusammuna töötajate enesetunne seoses struktuurimuudatusega, mis selgus asutuse C intervjuust „*.../Tegime majasiseselt välkküsitluse, et uurida kuidas inimesed tunnevad ennast 1 aasta peale struktuurimuudatust ja need tulemused olid väga positiivsed - tiimid tunnevad, et nad toimivad ja koostöö on oluliselt paranenud/.../*“. Asutuse D intervjuust selgus, et DevOps metoodika abil on suudetud ära hoida kriitiliste intsidentide põhjustamisi üksiksüsteemi poolt „*.../Oleme suutnud viia oma infosüsteemid tasemele, kus nad on suhteliselt töökindlad, st tiimi vastutuses olev süsteem päriselt töötab ja kliendile ei põhjustata süsteemi maas olekut ühe süsteemipõhiselt/.../*“. DevOps metoodika üks põhieesmärke on tagada tarkvara stabiilsus ja järjepidevus arendusest tootmiseni. DevOps metoodikal saavutatakse stabiilsus automatiseerimise, meeskonna koostöö, monitooringu ja pideva parendamise põhimõtete rakendamisega. Antud

põhimõtete integreerimine arenduse ja halduse protsessidesse võimaldab organisatsioonidel luua ja säilitada stabiilseid, usaldusväärseid ja tõhusaid süsteeme.

Teises intervjuu teemaplokis, mis käsitles uuritavate asutuste eelnevalt kasutusel olnud arendusmetoodika ning DevOps eeliste osa, tõid asutused välja, et eelnevalt kasutusel olnud töökultuuris puudusid kindla teenuse eest vastutavad tiimid ehk rollid, kes töötavad ühe projekti raames olid erinevates osakondades. Puudus teenusepõhisus ja kindlad vastutused ei olnud määratletud. Ühtlasi toodi välja, et DevOps metoodika eelisteks on halduse kaasamine protsessi, tööloikude automatiseerimine, täiustamine, uuenduslikud töövahendid ning efektiivsem kommunikatsioon. Lähtudes DevOps teoreetilistest lähtekohtadest, muudab DevOps lähenemine asutuse arendusprotsessi tõhusamaks läbi töökultuuri muutuse ja meeskonna ühise koostöö. Arendus muutub tõhusamaks ja tänu sellele jõuab klientideni kiiremini ja vähemate vigadega. Eksimused nt arenduses või halduses ning muus tekitavad töökeskkonnas lisastressi ja -pingeid. DevOps põhimõtetel töökultuur võib tagada paremat ja rahulikumat töökeskkonda.

Järeldused DevOps põhimõtete võrdluse ning eeliste osas teiste tarkvaraarenduse metoodikatega:

1. Koondatud vastustest selgus, et olulised õppetunnid DevOps metoodikale üleminekul on juhtkonna mõtteviisi muutus ning kogu organisatsiooni kaasamine mõtteviisi muutusele (vastutuse võtmine, ühtne tiimitunnetus, koostöö). Lisaks võib välja tuua oluliste õppetundidena ka rollide lahti mõtestamise ja tiimide vahelise kommunikatsiooni (sh koodi jagamine/taaskasutamine asutusesiseselt).
2. Eelneva põhjal võib järeldada, et DevOps pakub mitmeid eeliseid tarkvaraarendajatele ja organisatsioonidele tervikuna. Ka tellijate jaoks on oluline, et oleks tagatud stabiilne ja selge arenduskeskkond ja antakse ülevaade, millises seisus on tarkvaraloome.
3. Uuringu tulemustest võib järeldada, et asutus, mis rakendab DevOps metoodikat on süsteemid saanud töökindlamaks ning kliendile ei põhjustata süsteemi maas olekut ühe süsteemipõhiselt. DevOps arendusmetoodikat rakendades on võimalik saavutada IT jõudlus ja stabiilsus.

3.3 DevOps mõju asutustele ning meeskondade tagasiside

Vastavalt alapeatükile 1.2.3 võib DevOps töökultuurile üleminekul esineda ka teatavaid piiranguid. Intervjueeritavatel uuriti DevOps metoodikale üleminekul võimalike kitsaskohtade osas. Muudatustega võivad kaasneda eelised, mis tõstavad töö efektiivsust ning muudavad üldise töökeskkonna meeldivamaks, samas ka puudused/piirangud, mis omavad vastupidist mõju.

Vastustest lähtuvalt on koostatud tabel, kus on välja toodud uuringu käigus selgunud peamised DevOps töökultuurile üleminekul ja juurutamisel tekkinud kitsaskohad ning asutuste tähis, kes antud kitsaskohta on välja toonud.

Tabel 11. Asutuste DevOps töökultuurile üleminekul tekkinud kitsaskohad

asutuse tähis	DevOps töökultuurile üleminekul tekkinud kitsaskohad
A	info liikumine tiimide vahel
D	teenusepõhiste tiimide koostamine võib tekitada olukorra, et kõik valdkonna inimesed liiguvad tiimidesse ning eraldi jääb üks arhitekt, kes kõik keerulised olukorrad erinevates meeskondades peab ära lahendama
B	juhendmaterjalid koostamiseks ei ole piisavalt ajalist ressursi muude oluliste tööülesannete kõrvalt

Tabelist 11 on näha, et asutuste A poolt toodi välja kitsaskohti teenusepõhiste DevOps tiimide ja sellest lähtuvalt tiimide vahelise info liikumise osas „*.../Kui me tekitame teenusepõhised tiimid, siis tekivad jälle teistpidi „silotornid“, ehk valdkonna inimesed - arendajad või siis platvormi inimesed, on üle meeskondade laiali, siis peavad olema kindlasti tasakaalustavad mehhanismid, et nendel omavahel kuidagi info liiguks ning parim praktika saaks jagatud, et iga tiim ei leiutataks jalgratast, kuna ei teata, et juba mõnes teises tiimis on midagi sarnast tehtud/.../*“.

Lisaks selgub tabelist 11 asutuse D poolt toodud oluline kitsaskoht teenusepõhiste tiimide koostamise osas, kus kõik valdkonna inimesed liiguvad tiimidesse ning eraldi jääb arhitekt, millele tuleks DevOps üleminekul ja struktuurimuudatuse korraldamisel mõelda „*.../Teenusepõhiste tiimide korral liiguvad valdkonna inimesed arendustiimidesse, mis võib tekitada olukorra, kus on kohati „pudelikaelad“, nii et jääbki üks väga hea platvormi arhitekt, kes kõik keerulised probleemid majas kõikide meeskondade jaoks peab ära lahendama/.../*“.

Tabelist 11 selgub asutuse B intervjuus toodud olulise kitsaskohana asutuses kasutusel olevate platvormide muutmise/uuendamine ning sellest lähtuvalt näidiste ja juhendmaterjalide koostamine „*.../Kui DevOps spetsialistid teevad muudatusi platvormidel, siis sellest äritiimide teavitamine ning juhendmaterjalide loomine ja uuendamine on nõrk koht seoses DevOps spetsialistide ajalise ressursiga/.../*“.

Teoreetilises uuringus toodi välja, et DevOps metoodika üks oluline põhimõte on vähendada käsitööd ning olla automatiseeritum kõigis etappides. Siinkohal on näidiste ja juhendmaterjalide olemasolu väga oluline, et hoida hiljem kokku ajalist ressursi juhendamiste osas. Nagu ka alapeatükis 1.2.3 välja toodi, siis algne töökoormus töötajatel suureneb, kuna kasutusel on uued töövahendid ja põhimõtted. Samas tuleks siinkohal kindlasti näha tuleviku mõistes ressursi

kokkuhoidu, mis kulub muudatuste tegemiseks ning töötajate juhendamiseks. DevOps põhimõtteid rakendades on organisatsioonidel võimalik paremini kasutada oma arendusmeeskondade ajalist ressursi, vältides suuri vigu arenduse lõppfaasis ning tõsta tehtud töö ajalist väärtust.

Intervjueeritavatel uuriti DevOps metoodikale ülemineku osas meeskondade tagasisidet ning üldise töötajate tagasiside küsimise kohta asutustes. Asutuste vastustest selgus, et otseselt DevOps põhimõtetele ülemineku osas töötajatelt tagasisidet ei ole küsitud, küll aga toodi kõigi asutuste poolt välja, et neil on iga-aastased töötajate rahuloluküsitlused, mille raames saavad töötajad anda tagasisidet ka asutuses toimunud muudatuste osas „*.../Asutuses ei ole eraldi küsitud töötajatelt tagasisidet DevOps põhimõtetele ülemineku osas, küll aga on iga-aastane töötaja rahulolu-uuring (4), kus käiakse läbi koostöö ja muud teemad, sh kõik muudatused, mis asutuses tehakse. Töötajapoolne põhiline tagasiside väljendub töötaja rahulolu-uuringus/.../*“.

Lähtudes teoreetilisest alapeatükist 1.2.3 tuleks üleminekuperioodil reageerida muutustele, teha otsuseid uute olukordadega toimetulekuks ja harjumiseks. Näiteks, kui asutus on üle minemas uuele töökultuurile ja asutuses on sellest lähtuvalt toimumas ka suuremad muudatused, siis on töötajatelt tagasiside küsimine ning vastavalt sellele töötaja toetamine uuega kohanemisel väga oluline. Kohanemine on iga integreerimisprogrammi kõige põhilisem ja hädavajalik osa ning uuele arendusmetoodikale üleminekul peaks asutus sellega arvestama ja üleminekul töötajaid kohanemisel toetama.

Teoreetilises alapeatükis 1.3 tuuakse välja DevOps tiimide loomisel töötajate koolitamise olulisust. Sellest lähtuvalt uuriti intervjueeritavatel DevOps põhimõtete rakendamiseks spetsiifiliste koolituste/sertifitseerimiste korraldamist meeskondades. Vastustest lähtuvalt on koostatud tabel, kus on välja toodud uuringu käigus selgunud töötajate koolitamine seoses DevOps töökultuurile üleminekuga, asutuste tähis ning arv, kui palju on antud koolitust välja toodud.

Tabel 12. Töötajate koolitamine seoses DevOps töökultuurile üleminekuga

asutuse tähis	DevOps koolitused töötajatele	asutuste arv
A, B, C	tooteomanikele rollipõhiste koolituste korraldamine	3
A	ITIL (<i>Information Technology Infrastructure Library</i> , infotehnoloogia infrastruktuuride loetelu) raamistiku põhjal koolitused	2
C	välise <i>Agile Coach</i> ’i kaasamine	1
A	üldine majasisene agiilse mõtteviisi kasvatamine	1
D	suuremate muudatuste korral on kõigil töötajatel võimalik läbida vastavasisulisi koolitusi	1

Asutused A, B ja C tõid välja tooteomaniku/tootejuhi koolitused, mida asutusse sisse osteti, kui asutustes DevOps põhimõtete üleminekuga alustati „/.../Tooteomanikele korraldati erinevaid rollipõhiseid koolitusi (3), mis aitavad rolli paremini sisse elada, toovad välja tooteomaniku ülesanded ja vastutused ning annavad teadmised ja oskused tooteomaniku rollis töötamiseks/.../“. Tooteomanike koolitamine on oluline, et tagada nende valmisolek ja aktiivne osalus DevOps põhimõtetele üleminekul ning aidata neil saavutada paremaid tulemusi nii tootearenduse kiiruse kui ka stabiilsuse osas. Lisaks toodi asutuste poolt välja „ITIL raamistiku põhjal koolitused asutusele“, „välise Agile Coach'i kaasamine“ ning „üldine majasisene agiilse mõtteviisi arendamine“. Asutuse D poolt selgitati olulist seisukohta asutuses muudatuste läbiviimisel „/.../Kui asutuses tehakse suuremaid muudatusi, siis on kõikidel seotud töötajatel võimalik läbida vastavasisuline koolitus/.../“.

DevOps põhimõtete teoreetilisest osast lähtuvalt on oluline reageerida tellija ehk kliendi vajadustele agiilselt. Intervjueeritavatel uuriti, kuidas jälgitakse ja hinnatakse asutuses DevOps põhimõtete rakendamise mõju teenuste kvaliteedile ja kasutajakogemusele. Asutuste poolt toodi välja erinevaid võimalusi klientide tagasiside kogumisest.

Asutus A intervjuust selgus, et hetkel kindlat metoodikat selleks asutuse ei ole, kuid töö selles sunnas käib, et asutus saaks süsteemsemalt hakata kliendi tagasisidet hindama „/.../Üldiselt on iga-aastaselt klientidele rahuloluküsitlused (3), kus me hindame üldist rahulolu infosüsteemidega, rahulolu arenduste läbiviimisega ning üldist rahuolu asutusega. Tahaksime jõuda selleni, et tekiks arusaam, kuidas kliendi rahulolu vastu tiimi hinnata või kuidas tiim ise tunneb tervikuna, et kuidas neil läheb. Selle koha pealt töö käib, aga ei ole veel konkreetset metoodikat, mida kasutusse võtta/.../“.

Asutuse B poolt toodi välja, et klientide tagasisidet teenuse lõikes jälgitakse igapäevaselt „/.../Kliendid saavad anda tagasisidet igal ajal vastavasse tagasisidestamise kohta veebis, mis ühtlasi on seotud mõõdikutega. Tootejuhid näevad igapäevaselt klientide tagasisidet, sealtkaudu jälgime ka hinnanguid. Selle kohta on ka mitu head näidet, kuidas sellest on olnud suur abi. Näiteks tegime süsteemi tarne, järgmine päev hakkas tagasisidest järjest välja tulema, et mingisugune asi ei tööta või töötab harjumuspäraselt, mis oli suureks abiks, et olukord saaks kiiresti lahendatud/.../“.

Asutus D poolt toodi lisaks üldisele iga-aastasele klientide rahuloluküsitlusele välja ka transaktsioonipõhine küsitlus „/.../Kasutusel on kaks erinevat küsitlust, üks on transaktsioonipõhine, kus me küsime kliendi käest peale suurema muudatuse tegemist, et kuidas ollakse tulemusega rahul. Lisaks küsime klientide käest korra aastas üldist rahulolu, mis puudutab nii teenuseid kui ka teenuste arengut/.../“. Ühtlasi toodi välja ka alapeatükis 1.2.1 DevOps elutsükli faaside võrdluses, et üks DevOps faase on pidev tagasiside. Pidevat tagasisidet kogudes saavad DevOps tiimid oma protsesse

täiustada ja klientide tagasisidet järgmise versiooni täiustamiseks kaasata. DevOps tiim vastutab haldamise eest kuni IT-lahenduse elutsükli lõpuni, mil kasutajatel seda enam vaja ei ole. Arendusmeeskond täidab osaliselt ka edaspidi kasutajatoe rolli, aidates vajadusel klientidel oma ootusi edasi arendada.

Kolmandas intervjuu teemaplokis, mis käsitles DevOps metoodika kitsaskohti ning meeskondade tagasisidet, toodi oluliste kitsaskohtadena välja: info liikumist tiimide vahel, teenusepõhiste tiimide tekitamisel valdkonna inimeste ülekoormus ning sellest tulenevalt ka juhendmaterjalide ja näidiste koostamisel ajalise ressursi probleem. DevOps metoodika üks oluline põhimõte on vähendada käsitööd ning olla automatiseeritum kõigis etappides, et edaspidi oleks töömaht väiksem ning rohkem saaks tegeleda sisulisemate teemadega. Näidiste ja juhendmaterjalide olemasolu on oluline, et hoida hiljem kokku ajalist ressursi juhendamiste osas. Asutused peavad oluliseks töötajate koolitamist, mille raames toodi välja erinevaid koolitusi, mis seoses DevOps'ile üleminekuga korraldati (tooteomanike koolitus, ITIL raamistiku põhjal koolitus, välise *Agile Coach* i kaasamine ja üldine töötajate koolitamine suuremate muudatuste korral ning üldine agiilse mõtteviisi kasvatamine asutuses. Asutuste jaoks on oluline klientide tagasiside, mis ühtib ka DevOps põhimõtetega, kus pidev tagasiside küsimine on DevOps üks elutsükli faasidest.

Järeldused DevOps mõjust asutustele ning meeskondade tagasiside osas:

1. Koondatud vastustest võib järeldada, et teenusepõhiste tiimidel loomisel võib tekkida olukordi, kus info liigub tiimide siseselt, kuid tiimide vahel mitte nii hästi. Lisaks kasvab algselt töötajate töömaht seoses uute töövahendite kasutuselevõtuga, harjumisega ning juhendite koostamisega.
2. Uuringust selgus, et töötajatelt küsitakse asutuse muudatuste osas tagasisidet iga-aastaselt, mis on oluline, et töötaja saaks tuua välja tema jaoks olulised kohad ning asutus saaks töötajat vastavalt toetada.
3. Intervjueeritavate vastustest võib järeldada, et DevOps metoodikale üleminekuks on korraldatud asutustes erinevaid koolitusi, näiteks ITIL põhjal koolitused ja tooteomanike koolitused, mida enamuse intervjueeritavate poolt välja toodi. Lisaks peetakse oluliseks töötajate koolitamist erinevate muudatuste osas asutuses ning antakse võimalus töötajatel end vastavalt täiendada.
4. Tulemuste põhjal võib järeldada, et asutustes on klientidelt tagasiside küsimine oluline ning iga-aastaselt viiakse läbi klientide seas rahuloluküsitlusi, mille tulemusel mõõdetakse: kasutajate rahulolu infosüsteemidega, arendustega ning üldist rahuolu asutusega. Lisaks selgus ühest intervjuust, et klientidel on võimalik süsteemisiseselt tagasisidet anda igal hetkel ning lisaks küsitakse tagasisidet ka peale suuremate muudatuste tegemist.

3.4 DevOps põhimõtetele üleminek ja juurutamine riigi infotehnoloogia asutustes

Alapeatükis 1.3 toodi välja, et DevOps põhimõtetele üleminekul on esimeseks suureks sammuks organisatsiooni juhtkonna mõtteviisi muutus, millest lähtuvalt saab organisatsioon astuda esimesi samme töökultuuri muutmiseks. Intervjueeritavatel uuriti asutuste esimesi samme arendusmetoodika muutmiseks. Vastustest lähtuvalt on koostatud tabel, kus on välja toodud uuringu käigus selgunud asutuste tegevused DevOps metoodikale üleminekuks, asutuse tähis ning arv, kui palju antud tegevust välja toodi.

Tabel 13. Asutuste esimesed tegevused arendusmetoodika muutmiseks

asutuse tähis	esimesed tegevused arendusmetoodika muutmiseks	asutuste arv
A, B, C, D	struktuurimuudatus teenusepõhiste DevOps tiimide tekitamiseks	4
D	tiimides vastutuste tekitamine	1
A	pilveprojekti loomine tiimides	1
A	infoturbe aspekti integreerimine arendusprotsessi	1
B	kogu arendusprotsessi kaardistamine	1

Tabelist 13 selgub, et esmaseks oluliseks tegevuseks on struktuurimuudatus, mis teeb võimalikuks teenusepõhiste DevOps tiimide tekke, kuhu koondatakse nii Dev (arendus) poolt töötajaid kui ka Ops (haldus) poolt töötajaid „*.../Kõige suurem samm on olnud kõikide ettevalmistavate tööde raames struktuurimuudatus (4), mis teeb võimalikuks teenusepõhiste tiimide tekke ja sinna koondatakse nii Dev'i poolt kui ka Ops'i poolt töötajaid. Lisaks on üks oluline mõtteviisi muutus see, et tiim ei ole ainult termin „inimesed“, tiim on tervikuna see, kes on klient, see, kes oleme meie ja see, kes on ka arenduspartner. Selle mõtteviisi muutmise on hästi oluline, et teenusepõhised tiimid tööle saaksid hakata/.../*“.

Lähtudes alapeatükis 1.2 toodud DevOps põhimõtetest, on oluline, et struktuuris oleksid arendus- ja haldus koos ning tekiks teenusepõhised tiimid, kes vastutavas ühiselt antud teenuste eest.

Teise olulise sammuna selgub tabelis 13 asutus D poolt välja toodud tiimides vastutuste tekitamine „*.../Ühtpidi läksime baasteenustega teenusepõhiseks, et tekiks võimalus seda alusplatvormi teenusena kasutada. See oli üks esimesi samme ja teine sinna juurde oli, et hakata tiimidele rohkem vastutust tekitama/.../*“.

Nagu ka alapeatükis 1.3 on välja toodud, et DevOps mõtteviisil pannakse

suurt rõhku töökultuurile, mille üks eesmärkidest on, et tiimides oleks kindlad eesmärgid ning sellest lähtuvalt määratletud ka tiimide sisesed vastutused.

Lisaks selgus asutuse A poolt oluline määratlus pilveprojekti loomiseks tiimides „/.../Tiimid, kes tekitatakse, on üha rohkem sõltumatud teistest tiimidest, ehk siis tiimid saavad vastutada algusest lõpuni, neil on olemas töövahendid, neil on olemas oskused ning nad mõistavad oma rakendusi, oma süsteeme/.../“. Oluliste märksõnadena toodi asutuste poolt veel välja: infoturbe aspekti lahti mõtestamine asutuses ja kogu arendusprotsessi kaardistamine, et saada aru, kuidas on arendusprotsess seni toiminud ja mida peaks arendusprotsessis seoses arendusmetoodika muutumisega täpselt muutma. Teoreetilise osa alapeatükis 1.2.1 anti ülevaade, et agiilsed metoodikad keskenduvad peamiselt protsessi esimestele sammudele ning klientide ja arendajate vahelisele suhtlusele ning DevOps metoodika keskendub protsessi viimastele sammudele ning arendajate ja administraatorite vaheline suhtlusele. Lisaks on DevOps arendusmetoodika puhul suur erinevus, et suur osa arenduse protsessist on automatiseeritud. Alapeatükis 1.4 keskenduti infoturbe aspektidele, mille osas toodi välja, et kui organisatsioonid on DevOps põhimõtetele üleminekul ja juurutamisel, siis ei keskenduta piisavalt turvalisusele ning jäetakse tihti infoturbe aspekt katmata.

Tarkvara maailmas on DevOps muutumas üha enam standardiks ning antud metoodika rakendamisega püütakse saavutatud kasu paremini ära kasutada. Intervjueeritavatelt uuriti, et milliseid muutusi oodatakse DevOps metoodika juurutamisel seoses efektiivsuse, kiiruse ja kvaliteediga.

Asutus B vaates, kes on DevOps juurutamise faasis, selgus, et põhiline ootus on, et tarnete kiirus ja kvaliteet paraneks „/.../Täna on meil tärned nii-öelda üle nädala ehk kaks korda kuus. Aga soovime jõuda punkti, kus saame tärnet teha päevas mitu korda ja selliselt, et lõppkasutaja seda ei märka/.../“. Asutuse C vaates, kes on DevOps ülemineku faasis, selgus protsessi automatiseerimise vajadus „/.../Soovime protsessi automatiseerida/.../“.

Asutus D vaates, kes on DevOps rakendamise faasis, selgus, et DevOps metoodika juurutamisega on saadud süsteemide arendus efektiivsemalt tööle „/.../Arvan, et oleme saanud süsteemide arenduse palju efektiivsemalt tööle. Lisaks on meil IT-süsteemide osakaal haldusalas kasvanud päris arvestatavas koguses/.../. Vastavalt eelnevalt toodule, saab välja tuua, et kuna asutused on erinevas DevOps töökultuuri arengufaasis, siis on asutuste ootused erinevad. Samas võib eelnevalt toodu põhjal järeldada, et DevOps juurutamisega on asutuses D saadud arendusprotsess efektiivsemaks, mis on aidanud kasvatada ka haldusalas olevate süsteemide osakaalu.

Intervjueeritavalt uuriti, et mil määral tehakse koostööd teiste Eesti riigi infotehnoloogia asutustega, et jagada DevOps osas kogemust ja parimaid praktikaid. Intervjueeritavate vastustest selgus, et koostöö DevOps teemal on pigem vähene „.../DevOps osas on vähem ja sellel on väga praktiline põhjus sest kõik IT majad on erinevad ja see, kuidas ja mis töövahendeid kasutatakse jne/.../“ või on tehtud koostööd juhtkonna tasemel „.../Juhtkonna tasemel kindlasti, et enne kui seda muudatust tegema hakati, siis juhtkond ja osakonnajuhatajad suhtlesid teiste asutustega, et vaadata, millised on nende struktuurid, kuidas üldse midagi on tehtud, kuidas rollid on kirjeldatud jne/.../“. Lisaks toodi välja, et asutuse poolt, kes on DevOps metoodika rakendamise faasis, et nemad on käinud teistes asutustes tutvustamas „.../Oleme käinud erinevates IT majades tutvustamas, andnud ülevaate, kuidas on meie organisatsioon üles ehitatud, mis põhjustel niimoodi ehitatud jne/.../“.

Intervjueeritavatel uuriti, et millised on asutuste edasised plaanid seoses DevOps põhimõtete edasise juurutamise ja täiustamisega. Vastustest lähtuvalt on koostatud tabel, kus on välja toodud uuringu käigus selgunud DevOps põhimõtete edasine juurutamine ja täiustamine, asutuse tähis ning arv, kui palju antud tegevusi välja toodi.

Tabel 14. Asutuste tulevikuplaanid DevOps metoodika edasise juurutamise ja täiustamisega

asutuse tähis	DevOps metoodika edasine juurutamine ja täiustamine	asutuste arv
A, B, D	infoturbe integreerimine arendusprotsessi (DevSecOps)	3
B, D	töölõikude automatiseerimine	2
B	automaattestimise ülesehitamine	1

Tabelist 14 selgub, et 3 vastajat 4-st tõid välja, et DevOps metoodikat soovitakse täiustada infoturbe integreerimisega arendusprotsessi ehk DevSecOps metoodika juurutamisega „.../Kindlasti infoturbe (Sec) pool DevOps põhimõtetele integreerida (3) ehk iga arendusmeeskond vastutab selle eest, et tema teenused oleksid turvalised. Infoturbeosakond annab nõu ja toetab töövahenditega, et saaksime oma meeskonnale rohkem kätte nii turbe hindamise kui teostamise vahendeid. Suudaksime jõuda sinna, et infoturbeosakond oleks nõuandev organ. Tahame „Sec“ teadmise tekitada meeskondadesse/.../“. Alapeatükis 1.4 toodi välja, et DevSecOps lisab turvalisuse aspekti DevOps mudelisse, muutes see turvalisemaks ja vähendades turvariskide võimalusi tarkvaraarenduse ja tarnimise protsessis. DevSecOps on asutuse töökorraldus, mis põhineb ühtlasi agiilsetel printsiipidel ja olemuslikult on DevOps töökorralduse edasiarendus.

Tabelist 14 selgub, et asutuste B ja D poolt toodi välja ka automatiseerimise olulisus „.../Automatiseerimine (2), väga lihtsustatult öeldes tahame kõigi tiimidega jõuda sinnamaale, et

tooteomanik on võimeline kasutuselevõtu tegema/.../“. Vastavalt alapeatükile 1.2.2 võimaldavad automatiseeritud protsessid organisatsioonil rohkem funktsionaalsust luua. Protsesside automatiseerimine aitab vähendada tarnetele kuluvat töömahtu, võimaldades tarnida nii sageli kui tarvis.

Lisaks selgub, et asutus B toob välja ka automaattestide ülesehitamise olulisuse *„/.../Automaattestimise poolt ehitame järjest üles. Asutuses on nüüd olemas testijuht, kes kogu protsessi veab. Soovime protsessiga jõuda faasi, et arendajale tuleb näiteks hommikul pilet, ta arendab selle ära, testib, toimub automaattestimine ja kui kõik on korras, siis automaatika viib juba selle arenduse koheselt tootekeskonda/.../“.* Alapeatükis 1.3 toodi välja, et automaattestid on DevOps metoodika rakendamisel väga olulised, kuna arvutid suudavad teste rohkem ja kiiremini läbi viia ning probleemid leitakse kiiremini üles ehk arendusmeeskond saab võimalikult kiiresti reageerida ja probleemid kõrvaldada.

Viimases intervjuu teemaplokis, mis käsitles DevOps põhimõtetele ülemineku tegevusi ja juurutamist, tõid 4 asutust 4-st välja olulise tegevusena DevOps metoodikale üleminekul struktuurimuudatuse asutuses ning sellest lähtuvalt teenusepõhiste tiimide tekitamise. Siinkohal võib välja tuua, et need on olulised määratlused ka DevOps teoreetilistest lähtekohtadest, et struktuuris oleksid arendus- ja haldus koos, kes vastutavad ühiselt tiimis olevate teenuste eest. Lisaks toodi välja, et on oluline tiimides kindlate eesmärkide seadistamine ja vastutuste määratlemine, mis on välja toodud ka alapeatükis 1.3, et DevOps töökultuuris on üheks eesmärkidest, et tiimides oleks kindlad eesmärgid ning vastutused oleks määratletud. Ühtlasi toodi välja ka asutuste edasised plaanid DevOps juurutamise ja täiustamise osas, kus olulise märksõnana toodi välja infoturbe integreerimine (DevSecOps) arendusprotsessi. Teoreetilisest osast lähtudes võib välja tuua, et DevSecOps lisab turvalisuse aspekti DevOps mudelisse, muutes DevOps turvalisemaks ja vähendades turvariskide võimalusi tarkvaraarenduse ja tarnimise protsessis. Lisaks selgus tööloikude automatiseerimise olulisus ning automaattestide ülesehitamine. DevOps lähtub suuresti seisukohast, et kogu protsess peab olema võimalikult automatiseeritud.

Järeldused DevOps põhimõtetele ülemineku ja juurutamise osas:

1. Intervjueeritavate vastustest võib järeldada infoturbe aspekti olulisust DevOps metoodikale ülemineku faasis ning selle integreerimist võimalikult varajases etapis.
2. Intervjuude põhjal võib välja tuua, et infotehnoloogia asutustes võivad olla kasutusel erinevad rakendused, töövahendid ning -meetodid, kuid asutustel on võimalik üksteisele jagada kogemusi, näiteks kuidas alustada arendusmetoodika muutmise protsessi, millest alustada ning millele on

oluline kohe alguses mõelda. Lisaks on võimalik vahetada kogemusi, et kuidas asutuses arendusprotsess uute põhimõtete järgi üles ehitati, millest koosneb ning milliste reeglite järgi on korraldatud.

3. Uuringu tulemustest selgus, et asutus, kes rakendab DevOps metoodikat, on saanud arendusprotsessi efektiivsemaks, mis on ühtlasi aidanud kasvatada ka haldusalas olevate süsteemide osakaalu.
4. Uuringus osalenud asutuste jaoks on oluline DevOps metoodika edasiarendus DevSecOps metoodikale. DevSecOps lisab turvalisuse aspekti DevOps mudelisse.
5. Tulemustest võib järeldada, et oluliste täiustustena DevOps metoodika juurutamisel/rakendamisel on protsesside automatiseerimise olulisus ning automaatsete ülesehitamine. DevOps metoodika puhul on oluline, et suur osa arendusprotsessist oleks automatiseeritud.

3.5 Ettepanekud DevOps põhimõtetele üleminekuks

Uuringust selgus, et asutused teadvustavad „DevOps“ mõistet kui arenduse ja halduse koostöös toimuvat sujuvat ja võimalikult automatiseeritud töövoogu. Ühtlasi selgusid asutuste arendusmetoodika muutmist tinginud tegurid: arenduse, halduse ja äri koostöö edendamine, „silotornide“ lõhkumine struktuuris, teenuste toimivus, väline surve (kiiremad ja kvaliteetsemad teenused), kliendikeskus ning vastutuste määratlemine. Arendusmetoodika muutmine ja DevOps metoodikale üleminekuks võib tuua asutustele kasutajakesksema arenduskäsitluse, hõlbustada suhtlust arenduse, halduse ja äri vahel, mis lisaks hõlbustab kogemuste ja teadmiste edasiandmist meeskondade vahel.

Lähtuvalt analüüsi alajaotustes toodud järeldustest teeb autor Eesti riigi infotehnoloogia asutuste näitel teistele asutustele, kes alustavad DevOps metoodikale üleminekuks kümme (10) ettepanekut, mis võivad olla abiks DevOps põhimõtetele üleminekul.

Uuringust selgus, et olulisteks tegevusteks DevOps metoodikale üleminekul on asutuses teenusepõhiste DevOps tiimide tekitamine. DevOps põhimõtetest lähtuvalt on oluline teha asutuse struktuuris muudatus ja lõhkuda „silotornid“, et arendus ja haldus oleksid ühes meeskonnas ning oleksid väikesed, multifunktsionaalsed tiimid, kes töötavad autonoomselt ühiste eesmärkide ja vastutusega.

Ettepanek 1. Luua teenusepõhised DevOps tiimid, et hõlbustada koostööd arenduse, halduse ning äri vahel.

Uuringu tulemusel selgus ka *Agile Coach*’i roll asutuses, kes toetab asutust laiemalt, aitab protsessi koordineerida, teenusepõhiseid DevOps tiime toetada ning rollide osas nõustada ja juhendada.

Ettepanek 2. Luua *Agile Coach*’i ametikoht või kaasata väline *Agile Coach* protsessi koordineerimiseks ja asutuses agiilse mõtteviisi juurutamiseks.

Uuringu tulemusel selgus, et teenusepõhiste DevOps tiimide loomisel on oluline tiimides eesmärkide seadmine ning tiimide siseselt vastutuste määratlemine. Teenusepõhisel tiimil on ühised eesmärgid ning selleks on oluline jagada tiimi siseselt, et kes mille eest vastutab. DevOps tiimides ei vastuta lõpptulemuse eest üksik indiviid vaid kogu meeskond võtab ühise vastutuse selle eest, millisena lõppkasutajad tarkvara tajuvad.

Ettepanek 3. Meeskondliku vastutuse teadvustamiseks seada teenusepõhistes DevOps tiimides selged ühised eesmärgid ja määratleda iga rolli vastutus ühiste eesmärkide saavutamisel.

Tulemustest selgus, et üks olulisi aspekte DevOps töökultuurile üleminekul on uute põhimõtete ja töövahendite kasutuselevõtt ning nende rakendamiseks manuaalide ja näidiste koostamine, mille kasutamisele saab asutuse töötajaid suunata. DevOps üks põhimõte on vähendada käsitööd ning olla automatiseeritum kõigis etappides. Siinkohal on näidiste ja juhendmaterjalide olemasolu oluline, et hoida hiljem kokku ajalist ressursi juhendamiste osas.

Ettepanek 4. Kaaluda erinevate DevOps töömeetodite ja -vahendite kasutuselevõttu, näiteks tarkvaratööriistade komplekt *Ansible*, ning koostada vastavalt uute töömeetodite ja -vahendite kasutuselevõtiga juhendmaterjalid.

Uuringust selgus, et DevOps metoodikale üleminekuks on korraldatud asutustes erinevaid koolitusi, näiteks tooteomanikele rollipõhised koolitused ning üldine koolitus kõigile töötajatele ITIL raamistiku põhjal. Lisaks peetakse oluliseks töötajate koolitamist erinevate muudatuste osas asutuses ning antakse võimalus töötajatel end vastavalt täiendada. Koolituse ja juhendamise kulu võib asutusele algselt suure kuluna tunduda, kuid uuele töökultuurile üleminekul on oluline, et töötajad saaksid uute meetodite ja tööriistadega kiiremini tuttavaks. Algne rahaline kulu võib tunduda üüratu ja ebameeldiv, kuid arvestada tuleks edaspidist täiendavat ressurside kokkuhoidu. DevOps üks eesmärke on muuta töö tõhusaks ja kuluefektiivseks.

Ettepanek 5. Koolitada töötajaid ITIL raamistiku põhjal ning korraldada tooteomanikele rollipõhiseid koolitusi.

Uuringu tulemustest selgus, et asutuste jaoks on oluline DevOps metoodika edasiarendus DevSecOps metoodikale. DevSecOps lisab turvalisuse aspekti DevOps mudelisse, vähendades turvariske.

Selleks, et püsida konkurentsias peavad asutused suutma arengutega kaasa minna, sh DevOps metoodikat edasi arendada DevSecOps metoodika suunal, kus turvalisus on juba arendusprotsessi algfaasidest alates integreeritud ja see on kogu tarkvaraarenduse elutsükli jooksul kesksel kohal.

Ettepanek 6. Integreerida turvalisuse aspekt DevSecOps arendusprotsessi võimalikult varajases etapis, et tekitada turvalisuse teadmus DevOps tiimides.

Uuringust selgus protsesside automatiseerimise olulisus ning automaattestide ülesehitamine. DevOps arendusmetoodika puhul oluline, et suur osa arenduse protsessist oleks automatiseeritud. Protsesside automatiseerimine aitab vähendada tarnetele kuluvat töömahtu, võimaldades tarnida nii sageli kui tarvis. Automaattestid on DevOps meetodi puhul olulised, kuna automaattestidega suudetakse rohkem ja kiiremini läbi testida ning probleemid leitakse kiiremini üles ehk arendusmeeskond saab võimalikult kiiresti reageerida ning probleemid kõrvaldada.

Ettepanek 7. Automatiseerida arendusprotsessi tööloigud ning üles ehitada automaattestid.

Uuringu tulemusel selgus, et töötajatelt küsitakse asutuse muudatuste osas tagasisidet iga-aastaselt, mis on oluline, et töötaja saaks välja tuua olulise ning asutus saaks töötajat toetada uue töökultuuriga kohanemisel. Kui asutus on üle minemas uuele metoodikale ja sellest lähtuvalt on toimumas ka suuremad muudatused, siis on töötajatelt tagasiside küsimine ning vastavalt sellele töötaja toetamine uuega kohanemisel väga oluline, kuna kohanemine on iga sisseelamisprogrammi kõige olulisem ja vajalikum osa ning uuele arendusmetoodikale üleminekul peaks asutus sellega arvestama ja töötajaid kohanemisel toetama.

Ettepanek 8. Küsida vähemalt kord aastas töötajate tagasisidet seoses DevOps metoodika rakendamisega ja üldiste suuremate muudatustega asutuses.

Uuringust selgus, et asutuste jaoks on klientidelt tagasiside küsimine oluline, iga-aastaselt viiakse läbi kliendi rahuloluküsitlusi, mille tulemusel mõõdetakse: kasutajate rahulolu infosüsteemidega, arendustega ning üldist rahuolu asutusega. DevOps metoodikast lähtuvalt on tagasiside küsimine oluline, kuna see aitab parandada suhtlust klientidega ja seeläbi parandada teenindust. Kliendil võiks olla lisaks võimalus anda tagasisidet ka jooksvalt (süsteemisisene tagasiside andmise võimalus jne). Pidevat tagasisidet kogudes saavad meeskonnad oma protsesse täiustada ja klientide tagasisidet näiteks järgmise versiooni täiustamiseks kaasata.

Ettepanek 9. Juurutada klientidele agiilne süsteemisisene tagasiside andmise võimalus ning lisaks küsida klientidelt tagasisidet iga-aastaselt kliendi rahuloluküsitlustega.

Uuringu tulemustest selgus, et DevOps teemal on seni olnud koostöö teiste Eesti riigi infotehnoloogia asutustega pigem vähene või juhtkonna ja osakonnajuhatajate tasemel. Selleks, et DevOps töökultuur saada asutuse jaoks tööle, on soovituslik vahetada kogemusi ka teiste infotehnoloogia asutustega, kes antud põhimõtteid juba asutuses rakendavad.

Ettepanek 10. Vahetada kogemusi teiste infotehnoloogia asutustega DevOps metoodikale üleminekul ja rakendamisel.

KOKKUVÕTE

Lõputöö eesmärk oli kaardistada Eesti riigi infotehnoloogia asutuste näitel DevOps põhimõtetele üleminekut ja arendusmetoodika muutmist tinginud tegureid, et teha järeldusi ja ettepanekuid, mis võiksid olla abiks teistele infotehnoloogia teenust pakkuvatele asutustele DevOps põhimõtetele üleminekul ja juurutamisel.

Lõputöö eesmärk, kaardistada Eesti riigi infotehnoloogia asutuste näitel DevOps põhimõtetele üleminekut ja arendusmetoodika muutmist tinginud tegureid, autori hinnangul saavutati. Eesmärgi täitmiseks püstitati kuus (6) uurimisülesannet. Autori hinnangul kõik uurimisülesanded täideti.

Teoreetiliste uurimisülesannete täitmiseks, koostati autori poolt teoreetiline raamistik, kus toodi välja DevOps metoodika olemus, kasutamise eesmärk ja areng, DevOps põhimõtete võrdlus teiste tarkvaraarenduse metoodikatega sh DevOps ja agiilsete metoodikate elutsükli faaside võrdlus, DevOps eelised võrreldes teiste arendusmetoodikatega, DevOps töökultuurile üleminekut piiravad tegurid, DevOps metoodikate juurutamise põhimõtted ning DevOps metoodika täiustamine turvalisuse aspektiga ehk DevSecOps põhimõtetega.

Teoreetilistele alustele tuginedes koostas autor intervjuu küsimustiku. Intervjuule vastasid 2023. aasta oktoobris nelja (4) Eesti riigi infotehnoloogia asutuse DevOps põhimõtetele ülemineku ja/või juurutamise protsessi eest vastutav töötaja või töötajad, kes on antud asutuses DevOps põhimõtetele üleminekuga kõige rohkem kursis. Intervjuude vastused transkribeeriti ning kodeeriti, et leida sarnaseid vastuseid analüüsi läbiviimiseks, mis lõi aluse viienda uurimisülesande lahenduseks.

Viiendale uurimisülesandele, kaardistada asutuste DevOps põhimõtetele üleminekut, arendusmetoodika muutmist tinginud tegureid ja uue metoodika eeliseid võrreldes asutuses kasutatava arendusmetoodikaga, koguti intervjuude vastustest vastavasisulised märksõnad analüüsi käigus. Vastajad avaldasid põhilised arendusmetoodika muutmist tinginud tegurid: arenduse, halduse ja äri koostöö edendamine, „silotornide“ lõhkumine struktuuris ehk teenusepõhiste DevOps tiimide tekitamine, teenuste toimivus, väline surve (kiiremad ja kvaliteetsemad teenused), kliendikeskus ning vastutuste määratlemine. Arendusmetoodika muutmine ja DevOps metoodikale üleminek võib tuua asutustele kasutajakesksema arenduskäsitluse, hõlbustada suhtlust arenduse, halduse ja äri vahel, mis lisaks hõlbustab kogemuste ja teadmiste edasiandmist meeskondade vahel.

Uurimisülesandele, koostada ettepanekud, mis võiksid olla abiks teistele infotehnoloogia teenust pakkuvatele asutustele DevOps põhimõtetele üleminekul, seadis autor kümme (10) ettepanekut

DevOps põhimõtetele üleminekuks. Olulisemateks punktideks DevOps metoodikale üleminekuks osutusi: teenusepõhiste DevOps tiimide loomine, *Agile Coach*’i kaasamine, meeskondliku vastutuse teadvustamine ning iga rolli vastutuse määratlemine, DevOps töömeetodite ja -vahendite kasutuselevõtt ning juhendite koostamine, töötajate koolitamine (näiteks rollipõhised koolitused tooteomanikele, ITIL koolitus töötajatele), turvalisuse aspekti DevSecOps integreerimine arendusprotsessi, töövoogude ja -protsesside automatiseerimine ning automaatsete ülesehitamine, töötajatelt tagasiside küsimine seoses DevOps metoodika rakendamisega ja üldiste suuremate muudatuste korral, klientidele agiilse süsteemisese tagasiside andmise võimaluse loomine, kogemuste vahetamine teiste infotehnoloogia asutustega.

SUMMARY

The title of the thesis is *The Transition of Estonian State IT Institutions to the DevOps Principles*.

The thesis contains three chapters: a theoretical chapter on the nature and implementation of DevOps, the choice of research methodology, and the analysis and proposals of the research. The thesis consists of 58 pages, 3 figures, and 14 tables. The thesis is based on 60 sources.

The topic of the thesis has not been researched academically, however, the DevOps work culture is gaining popularity worldwide and is used by major corporations such as Amazon, Netflix, NASA (Automate, 2018). DevOps was first used in 2009 when P. Debois organised a conference in Belgium called DevOps Days (Edwards, 2019) Bass, Weber and Zhu defined DevOps as a set of practices aimed at reducing the time between system change and normal production while ensuring high quality (Filip, 2020). There is little research on the subject in Estonia and the topic does not have a solid framework or common definition.

The thesis aims to map the factors that led to the transition to the DevOps principles and the change in development methodology using the example of Estonian state IT institutions.

Tasks set to meet the objective:

- outline the nature and purpose of DevOps;
- compare the nature and principles of DevOps with other software development methodologies;
- outline the principles of the DevOps deployment process and the enhancement of the DevOps methodology with the principles of DevSecOps;
- develop a research tool for an empirical study based on the chosen methodology;
- map the factors that led to the transition of institutions to the DevOps principles, the factors that led to the change in development methodology, and the advantages of the new methodology compared to the development methodology used in the institution;
- draw conclusions and make suggestions based on the results that could help other IT organisations in their transition to DevOps.

The structure of the thesis follows a deductive approach. The research strategy is a survey. A qualitative data collection method was used to collect the data. The research subjects of the thesis are four (4) employees responsible for the DevOps transition and/or deployment process in Estonian state

IT institutions who are most familiar with the DevOps transition in the institution. Semi-structured interviews were used as the research method. Qualitative content analysis was carried out based on these four (4) semi-structured interviews. The author considers the aim of the thesis, which was to map the factors that led to the transition to DevOps principles and the change in development methodology using the example of Estonian state IT institutions, to be achieved. The main factors that led to the change in the development methodology were identified: promoting collaboration between development, administration, and business, breaking down the ‘silos’ in the structure, i.e. creating service-based teams, service performance, external pressure (faster and higher quality services), the customer-centric approach, and defining responsibilities. Changing the development methodology and transitioning to DevOps can bring a more user-centric approach to development for institutions and promote communication between development, administration and business, which also facilitates the transfer of experience and knowledge between teams. Six (6) research tasks were posed to meet the objective, and the author considers that the research tasks were completed. The author has drawn up ten (10) suggestions based on the results that could help other IT organisations in their transition to DevOps. The most important aspects in transitioning to and implementing the DevOps methodology were: the establishment of service-based DevOps teams, involvement of an agile coach, awareness of team responsibilities and definition of the responsibilities of each role, introduction of DevOps working methods and tools and the creation of guidelines and templates in this context, staff training (e.g. role-based training for product owners, ITIL training for staff), integrating the DevSecOps security aspect into the development process, automating workflows and processes and setting up automated tests, asking staff for feedback on the implementation of the DevOps methodology and on major changes in general, providing customers with the option of giving agile feedback within the system, and exchanging experiences with other IT organisations.

VIIDATUD ALLIKAD

Kuusik, Virk, Aarna. (2010). *Teadlik turundus*. Tartu Ülikooli Kirjastus.

AKIT. (s.a). Allikas: Andmekaitse ja infoturbe leksikon: <https://akit.cyber.ee/term/8926-devops>

Alvarenga, G. (2022). *DevOps vs. DevSecOps*. Allikas: CrowdStrike: <https://www.crowdstrike.com/cybersecurity-101/cloud-security/devops-vs-devsecops/>

Automate. (2018). *6 Companies That Are Doing DevOps Well*. Allikas: Automate: <https://www.fortra.com/blog/6-companies-are-doing-devops-well>

BrainStation. (s.a). *Product Manager vs Project Manager*. Allikas: BrainStation: <https://brainstation.io/career-guides/product-manager-vs-project-manager>

Brush, Silverthorne. (2019). *Agile software development*. Allikas: Techtarget: <https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development>

DASA DevOps Certification. (s.a). *IT Governance & Service*. Allikas: APMG International: <https://apmg-international.com/product/dasa-devops>

Dash Technologies Inc. (2022). *Agile Vs Scrum Vs DevOps: How They Are Different?* Allikas: Dash Technologies: <https://dashtechinc.com/agile-vs-scrum-vs-devops-how-they-are-different/>

Debois, P. (2008). *Agile Infrastructure & Operations*. Allikas: Jedi: <https://www.jedi.be/presentations/agile-infrastructure-agile-2008.pdf>

Department of Defence. (2021). *OSD. DoD Enterprise DevSecOps Strategy Guide*. Allikas: <https://dodcio.defense.gov/Portals/0/Documents/Library/DoDEnterpriseDevSecOpsStrategyGuide.pdf>

Digitark. (2023). *DevOps metoodika muudab tarkvaraarenduse tõhusamaks ja kiiremaks*. Allikas: Digitark Telia: <https://digitark.telia.ee/era/devops-metoodika-muudab-tarkvaraarenduse-tohusamaks-ja-kiiremaks/>

Docker Inc. (2019). *Docker overview*. Allikas: Docker docs: <https://docs.docker.com/get-started/overview/>

Education Wiki. (s.a). *DevOps põhimõtted - DevOps põhimõtete põhjalik juhend*. Allikas: Education Wiki: <https://et.education-wiki.com/3406646-devops-principles>

- Edwards, D. (2019). *The History of DevOps*. Allikas: IT revolutions: <https://itrevolution.com/articles/the-history-of-devops/>
- Erich, F. (2014). *Report: DevOps Literature Review*. Allikas: Researchgate: https://www.researchgate.net/publication/267330992_Report_DevOps_Literature_Review
- EUCIP. (s.a). *Elutsükli faasid*. Allikas: EUCIP: https://eopearhiiv.edu.ee/e-kursused/eucip/arendus/121_elutskli_faasid.html
- Filip, A. M. (2020). Allikas: TechAcute Empowered by information: <https://techacute.com/what-is-devops/>
- Franklin, C. (2016). *Agile vs. DevOps: 10 Ways They're Different*. Allikas: Information Week: <https://www.informationweek.com/software-services/agile-vs-devops-10-ways-they-re-different>
- Gartner. (2019). *12 Things to Get Right for Successful DevSecOps*. Allikas: Gartner: <https://www.gartner.com/en/documents/3978490>
- GitLab. (s.a). *What is DevOps?* Allikas: GitLab: <https://about.gitlab.com/topics/devops/>
- Hall, T. (s.a). *DevOps vs. Agile. What are the differences and similarities between agile and DevOps?* Allikas: Atlassian : <https://www.atlassian.com/devops/what-is-devops/agile-vs-devops>
- Hamilton, T. (2023). *Agile Vs. DevOps – Difference Between Them*. Allikas: Guru99: <https://www.guru99.com/agile-vs-devops.html>
- Hirsjärvi, Huttunen. (2005). *Sissejuhatus kasvatustadustesse*.
- Hoogenraad, W. (2018). *Agile ja Waterfall metoodika olulised erinevused*. Allikas: ITpedia: <https://www.itpedia.nl/et/2018/09/19/10-belangrijke-verschillen-tussen-agile-en-waterfall-methodology/>
- IBM. (s.a). *Practice building culture*. Allikas: IBM: https://www.ibm.com/cloud/garage/content/culture/practice_building_culture/
- IKT arenduste toimemudel. (2023). Allikas: Rahandusministeerium: <https://www.fin.ee/media/9045/download>
- Jabbari, Ali, Petersen, Tanveer. (2016). *What is DevOps?: A Systematic Mapping Study on Definitions and Practices*. Allikas: Research Gate:

- https://www.researchgate.net/publication/308857081_What_is_DevOps_A_Systematic_Mapping_Study_on_Definitions_and_Practices
- Jeremiah, J. (2016). *Survey: Is agile the new norm?* Allikas: TechBeacon: <https://techbeacon.com/app-dev-testing/survey-agile-new-norm>
- Kalmus, V. (2015). *Kvalitatiivne sisuanalüüs*. Allikas: Tartu Ülikooli kvalitatiivsed uurimismeetodid sotsiaalteadustes: <https://sisu.ut.ee/kvalitatiivne/kvalitatiivne-sisuanal%C3%BC%C3%BCs>
- Koskinen, A. (2020). *DEVSECOPS: BUILDING SECURITY INTO THE CORE OF DEVOPS*. Allikas: <https://jyx.jyu.fi/bitstream/handle/123456789/67345/1/URN%3ANBN%3Afi%3Aju-202001171290.pdf>
- Källo, E. (2017). *Töö- ja elukeskkonnaga kohanemine kolmandate riikide töötajate näitel Eesti*. Allikas: Tartu Ülikooli Dspace: <https://dspace.ut.ee/server/api/core/bitstreams/6a7bcb26-18eb-4f2e-9220-90d477d79e18/content>
- Labs, P. (2014). *State of DevOps Report*. Allikas: <https://services.google.com/fh/files/misc/state-of-devops-2014.pdf>
- Labs, P. (2015). *State of DevOps Report*. Allikas: <https://services.google.com/fh/files/misc/state-of-devops-2015.pdf>
- Linchpinseo. (2023). *A Beginners Guide To The Agile Method*. Allikas: linchpinseo: <https://linchpinseo.com/blog/the-agile-method/>
- Loukides, M. (2012). *What is DevOps? Infrastructure as Code*. Allikas: Radar: <http://radar.oreilly.com/2012/06/what-is-devops.html>
- Mazyar, M. (2018). *DevOps: The Ultimate Way to Break Down Silos*. Allikas: DevOps: <https://devops.com/devops-the-ultimate-way-to-break-down-silos/>
- Morris, L. (2022). *Difference Between Agile and Waterfall: Software Development Methodologies*. Allikas: Project Management: <https://project-management.com/agile-vs-waterfall/>
- Mueller, E. (2020). *What Is DevOps?* Allikas: The Agile Admin: <https://theagileadmin.com/what-is-devops/>
- Myrbakken, Colomo-Palacios. (2017). *DevSecOps: A Multivocal Literature Review*. Allikas: ResearchGate:

https://www.researchgate.net/publication/319633880_DevSecOps_A_Multivocal_Literature_Review

Ojasalu, S.-M. (2021). *DevOps ja selle rakendatavuse uuring Põhja-Eesti Regionaalhaiglas*. Allikas: TÜ arvutiteaduse instituudi lõputööde register: https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=71877&year=2021

Parmakson, P. (2018). *Mikroteenusearhitektuurist*. Allikas: Riigi Infosüsteemi Amet: <https://e-gov.github.io/TARA-Stat/>

Plutora. (2021). *The Agile + DevOps Effect: 6 Keys to Software Delivery Evolution*. Allikas: Plutora: <https://www.plutora.com/blog/agile-devops-effect-6-keys-software-delivery-evolution>

Põllumäe, J. (2019). *DevOps põhimõtete juurutamine agiilses tarkvaraarendusprotsessis Elisa AS näitel*. Allikas: TalTech Digikogu: <https://digikogu.taltech.ee/>

Ravichandran, Taylor, Waterhouse. (2016). *DevOps for digital leaders*. rmt: K. T. Aruna Ravichandran, *Reignite Business with a Modern*.

Red Hat. (s.a). *Why use DevOps automation?* Allikas: RedHat: <https://www.redhat.com/en/engage/use-devops-automation-s-202106101142>

Ries, E. (2009). *Minimum Viable Product: a guide*. Allikas: Startup lessons learned: <http://www.startuplessonslearned.com/2009/08/minimum-viable-product-guide.html>

Riungu-Kalliosaari, L. M. (2019). *DevOps Adoption Benefits and Challenges in Practice: A Case Study*.

Rush, L. (2016). *DevOps printsiipide rakendamise analüüs TransferWise'i näitel*. Allikas: TalTech Digikogu: <https://digikogu.taltech.ee/>

Sanchez-Gordon, Colomo-Palacios. (2018). *Characterizing DevOps Culture: A Systematic Literature Review: 18th International Conference*. Allikas: ResearchGate: https://www.researchgate.net/publication/327675534_Characterizing_DevOps_Culture_A_Systematic_Literature_Review_18th_International_Conference_SPICE_2018_Thessaloniki_Greece_October_9-10_2018_Proceedings

Senapathi, Buchan, Osman. (2018). *DevOps Capabilities, Practices, and Challenges: Insights from a Case Study*. Allikas: ACM Digital Library: <https://dl.acm.org/doi/10.1145/3210459.3210465>

- Sharma, S. (2017). *The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT*. Allikas: Amazon: <https://www.amazon.com/DevOps-Adoption-Playbook-Multi-Speed-Enterprise/dp/1119308747?asin=1119308747&revisionId=&format=4&depth=1>
- Singleton. (2023). *Kes või mis on DevOps ja miks see arenduses nii tähtis on?* Allikas: DigiPRO: <https://digipro.geenius.ee/blogi/singletoni-blogi/kes-voi-mis-on-devops-ja-miks-see-arenduses-nii-tahtis-on/>
- Siseministeeriumi haldusala IKT teenuse arendamise ja haldamise lõpparuanne. (2018). Allikas: SMIT: <https://www.smit.ee/files/ikt-finantseerimine-pwc-lopparuanne-veeb.pdf?a8072e9c5b>
- Snyk. (s.a). *DevSecOps Overview*. Allikas: Snyk: <https://snyk.io/series/devsecops/>
- Thomas, D. (2001). *Manifesto for Agile Software Development*. Allikas: Agile Manifesto: <https://agilemanifesto.org/>
- Tucker, Pleban, Gunther. (2010). The Mediating Effects of Adaptive Skill on Values-Performance Relationships. <https://www.tandfonline.com/doi/abs/10.1080/08959280903400275>. Allikas: Tandfonline.
- Uptime. (s.a). *DevSecOps: kuidas integreerida küberturvalisus arendusprotsessi esimesest hetkest alates*. Allikas: Uptime: <https://www.uptime.ee/devsecops-kuidas-integreerida-kuberturvalisus-arendusprotsessi-esimesest-hetkest-alates/>
- Virmani, M. (2015). *Understanding DevOps & bridging the gap from continuous integration to continuous delivery*. Allikas: ResearchGate: https://www.researchgate.net/publication/308730807_Understanding_DevOps_bridging_the_gap_from_continuous_integration_to_continuous_delivery
- Õunapuu, L. (2014). Kvalitatiivne ja kvantitatiivne uurimisviis sotsiaalteadustes.
- Yigal, A. (s.a). *What is DevOps? The Beginner's Guide from Logz.io*. Allikas: Logz.io: <https://logz.io/learn/what-is-devops/>

LISAD

Lisa 1. Intervjuu küsimused

LISA 1 INTERVJU KÜSIMUSED

Uuringu eesmärk on kaardistada Eesti riigi infotehnoloogia asutuste näitel DevOps põhimõtetele üleminekut ja arendusmetoodika muutmist tinginud tegureid, et teha antud tulemuste põhjal järeldusi ja ettepanekuid, mis võiksid olla abiks teistele infotehnoloogia asutustele DevOps põhimõtetele üleminekul.

Intervjuu küsimused on koostatud DevOps põhimõtetele ülemineku/juurutamise protsessi eest vastutavatele töötajatele või töötajatele, kes on asutuses DevOps põhimõtetele üleminekuga kõige rohkem kursis. Intervjuu andmeid kasutatakse üldistatud kujul. Konfidentsiaalsuse huvides ei avaldata analüüsis uuritavate asutuste ega asutusi esindavate intervjuueeritavate nimesid või vihjata kuidagi asutusele, millest võiks teha järeldusi, soovides kaitsta asutuste mainet, huve ning poliitikat.

(Intervjuueeritavalt nõusoleku küsimine intervjuu salvestamiseks).

Kas Teil on intervjuu või teemade osas küsimusi?

Sissejuhatavad küsimused

1. Milline on Teie asutuse roll Eesti riigi infotehnoloogia teenuste osutamisel?
2. Milline on/oli Teie roll DevOps metoodikale ülemineku protsessis?

DevOps määratlemine ja arendusmetoodika muutmist tinginud tegurid

3. Selgitage palun, kuidas defineerite DevOps olemust ja eesmärki.
4. Mis olid peamised tegurid, mis tingisid asutuses arendusmetoodika muutmist?
5. Kas Teie asutus on juba alustanud üleminekut DevOps põhimõtetele või plaanite seda teha?
Kui jah, millal alustasite või plaanite alustada?

DevOps põhimõtete võrdlus ja eelised teiste tarkvaraarenduse metoodikatega

6. Kuidas kirjeldaksite senist IT- tegevuste haldamise ja arendamise protsessi Teie asutuses enne DevOps põhimõtete rakendamist?
7. Mil määral on muutunud võrreldes senise arendusmetoodikaga tööriistad ja tehnoloogiad asutuses?
8. Milliseid muutusi olete märganud või ootate DevOps põhimõtete juurutamise tulemusel, eriti seoses efektiivsuse, kiiruse ja kvaliteediga?
9. Millised on DevOps põhimõtete eelised võrreldes senise IT-tegevuste haldamise ja arendamise metoodikaga?

10. Kuidas hindaksite senist edu üleminekul DevOps põhimõtetele? Millised on olnud peamised õppetunnid ja saavutused?

DevOps mõju asutusele ning meeskondade tagasiside

11. Milliseid konkreetseid kitsaskohti näete üleminekul DevOps põhimõtetele?
12. Kuidas antakse asutuses tagasisidet, kui palju antakse töötajatele tagasisidet, kellelt ja mis kujul?
13. Milline on olnud meeskonna tagasiside DevOps töökultuurile üleminekuga seoses?
14. Kuidas kaasate ja koolitate oma meeskonda DevOps põhimõtete rakendamiseks? Kas olete korraldanud spetsiifilisi koolitusi või sertifitseerimisi?
15. Kuidas jälgite ja hindate DevOps põhimõtete rakendamise mõju Teie teenuste kvaliteedile ja kasutajakogemusele?

DevOps põhimõtetele üleminek ja juurutamine

16. Kuidas olete kavandanud ja juurutanud DevOps põhimõtteid Teie asutuses? Millised on olnud esimesed sammud?
17. Kas ja mil määral olete teinud koostööd teiste Eesti riigi infotehnoloogia asutustega, et jagada kogemusi ja parimaid praktikaid DevOps rakendamisel?
18. Millised on Teie plaanid seoses DevOps põhimõtete edasise rakendamise ja täiustamisega?

Aitäh intervjuu eest!