

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Greta Reino 193736IABB

# **Tarkvaraarenduse protsessianalüüs ja - parendus ettevõtte A näitel**

Bakalaureusetöö

Juhendaja: Karl-Erik Karu  
MSc

Tallinn 2023

## **Autorideklaratsioon**

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

Autor: Greta Reino

02.01.2023

## **Annotatsioon**

Bakalaureusetöö eesmärk on analüüsida ettevõttes A kasutusel olevat tarkvaraarenduse protsessi, leida kitsaskohad ning seda parendada. Tulenevalt analüüsi tulemustele ja tööprotsessi korrigeerimisele muutub ettevõtte töö efektiivsemaks nii töötajate kui klientide jaoks.

Töös käsitletav probleem seisneb asjaolus, et ettevõttes puudus varasemalt IT-projektijuht ning arendusmeeskonna töös on märgata segadust ja arusaamatusi. Tiimil ei ole ühtset arusaama ettevõtte vajadustest ega prioriteetidest ning sellest tulenevalt puudub kindel tegevuskava.

Eesmärkide saavutamiseks kaardistatakse AS-IS olukord arendusprotsessis ning viiakse läbi küsitlus arendusmeeskonnas. Koostatakse ka projektijuhi kohustuste täideviimise analüüs vastavalt teooriale. Praktilise osa toetamiseks antakse teoreetilises pooles selgitav ülevaade ning kirjeldatakse nii projektijuhtimise kui ka arendusprotsessi etappe põhjalikumalt. Töö tulemusena koostatakse TO-BE protsessimudelid, tuvastatakse peamised ettevõttesisesed probleemid, leitakse neile lahendused ning võimalusel implementeeritakse.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 37 leheküljel, 7 peatükki, 15 joonist ja 8 tabelit.

## **Abstract**

### **Software Development Process Analysis and Improvement on the Example of Company A**

The purpose of this bachelor thesis is to identify and find solutions to problems occurring in the software development process in company A. The analysis and correction of the work process results in the company's work becoming more effective for both employees and customers.

The problem discussed in this thesis comes from the fact that previously there was no IT project manager, which caused confusion and misunderstandings in the developing team's work. The team lacks a unified understanding of the company's needs and priorities, therefore a certain action plan for the future is missing.

To achieve the purpose of this thesis, an AS-IS developing process will be mapped out and a survey conducted within the developing team. In addition, an analysis of the performance of the project manager's duties according to the theory is formed. To support the practical part, an explanatory overview is given in the theoretical part and the stages of both project management and the development process are described in more detail. As a result of the thesis, TO-BE process models are formed, the main problems within the company are identified, solutions are suggested and, if possible, implemented.

The thesis is in Estonian and contains 37 pages of text, 7 chapters, 15 figures, 8 tables.

## Lühendite ja mõistete sõnastik

Jira	Ülesannete jälgimise seade, projektijuhtimise vahend [1]
UI	User Interface ehk disaini põhimõte, kus keskendutakse kasutajaliidesele ja selle struktuurile, visuaalsele ilmele ja stiilile [2]
UX	User Experience ehk disaini põhimõte, kus keskendutakse kasutajakogemusele [2]
<i>Front-end</i>	Kasutajaliides [3]
<i>Full-stack</i>	Kasutajaliides ja tagarakendus
Framework	Raamistik [3]
<i>Release</i>	Tarkvara avaldamisvalmis uuendus [3]
Slack	Ettevõtetele mõeldud sõnumsiderakendus [4]
<i>Backlog</i>	Loend tegevustest, mis vajavad arendustiimil tegemist [5]
<i>SD board</i>	System development ehk tarkvaraarenduse <i>board</i>
<i>CS board</i>	Customer support ehk klienditoe <i>board</i>
<i>Stand-up</i>	Igapäevane arendustiimi kiirkoosolek
<i>Task</i>	Ülesanne, tegevus
<i>Board</i>	Agiilne projektihaldustööriist, mis on loodud töö visualiseerimiseks, pooleliolevate tööde piiramiseks ja tõhususe maksimeerimiseks. [6]
<i>Weekly</i>	Korra nädalas toimuv koosolek
AS-IS	Meetod, mis kirjeldab organisatsiooni protsessi hetkeolukorda
TO-BE	Meetod, mis kirjeldab organisatsiooni protsessi tuleviku olukorda

## Sisukord

1 Sissejuhatus .....	10
1.1 Taust ja probleem .....	10
1.2 Eesmärk .....	11
1.3 Töö struktuur .....	11
2 Metoodika .....	13
2.1 Ettevõtte ja meeskonna lühituvustus .....	13
2.2 Tööriistad .....	13
2.2.1 Jira .....	13
2.2.2 Kanban .....	14
2.2.3 Kanban board .....	14
2.3 Protsess .....	16
2.3.1 Projektijuhtimise metoodika PMBOK'i baasil .....	16
2.3.2 Arendusprotsess .....	18
3 Tarkvaraarenduse protsessi kaardistamine .....	21
3.1 Nõuete kogumine .....	22
3.2 Analüüs ja planeerimine .....	23
3.3 Tarkvaraarendus ja testimine .....	24
3.4 Juurutamine ja ülalhoid .....	26
4 Läbiviidud küsitluse tulemused .....	28
4.1 Arendusprotsessi kirjeldus ja probleemid .....	29
4.2 Kanban boardi kasutus .....	30
4.3 Organisatoorne tagasiside .....	32
5 Protsessi parendusettepanekud .....	36
5.1 Arendusprotsessi pakutud TO-BE lahendused .....	36
6 Analüüs ja järeldused .....	41
6.1 Projektijuhi tegevuste analüüs .....	41
6.1.1 Projekti alustamine .....	41
6.1.2 Projektiplaanide esitamine .....	42
6.1.3 Meeskonna töö koordineerimine ja liikmete toetamine .....	43

6.1.4 Projekti kulgemise jälgimine .....	43
6.1.5 Riskide ja konfliktide haldamine .....	43
6.1.6 Kommunikatsiooni tagamine meeskondade ja liikmete vahel .....	44
6.1.7 Projekti lõpetamine .....	44
6.2 Tulemuste järelused .....	44
7 Kokkuvõte .....	47
Kasutatud kirjandus .....	48
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	50
Lisa 2 - Küsitlus.....	51

## Jooniste loetelu

Joonis 1. Kanban board [10].....	16
Joonis 2. Projektijuhtimise etapid [11] .....	17
Joonis 3. Tarkvaraarenduse elutsükkel [12] .....	19
Joonis 4. AS-IS: Ettevõtte A tarkvaraarenduse alamprotsessid .....	22
Joonis 5. AS-IS: Nõuete kogumine .....	23
Joonis 6. AS-IS: Analüüs ja planeerimine .....	24
Joonis 7. AS-IS: Nõuete olekud .....	24
Joonis 8. AS-IS: Tarkvaraarendus ja testimine .....	25
Joonis 9. AS-IS: Ticketi olekud.....	26
Joonis 10. AS-IS: Juurutamine ja ülalhoold .....	27
Joonis 11. Boardi hinnang .....	31
Joonis 12. Boardi väliste ülesannete sagedus .....	31
Joonis 13. TO-BE: Nõuete kogumine ja analüüs .....	39
Joonis 14. TO-BE: Arenduse planeerimine .....	39
Joonis 15. TO-BE: Nõuete olekud.....	40



## **Tabelite loetelu**

Tabel 1. Vastanute arvukus ja positsioon .....	28
Tabel 2. Kitsaskoht 1 .....	36
Tabel 3. Kitsaskoht 2 .....	36
Tabel 4. Kitsaskoht 3 .....	37
Tabel 5. Kitsaskoht 4 .....	37
Tabel 6. Kitsaskoht 5 .....	37
Tabel 7. Kitsaskoht 6 .....	38
Tabel 8. Kitsaskoht 7 .....	38

# 1 Sissejuhatus

Tiheda konkurentsiga tehnoloogiaektoris peavad ettevõtted edukuse nimel olema valmis pidevalt arenema ja muutuma, et pakkuda klientidele kõige uuemaid ja innovatiivsemaid lahendusi. On oluline tõestada, et ettevõtte suudab pakkuda parima lahenduse kliendi probleemile, kohandudes seejuures nii turuolukorra kui ka kliendi vajadustega. Toote arenduse ja vastavalt situatsioonile sobivaks muutmise taga on ettevõtte tiim ja selle sisesed protsessid. Kvaliteetne lõpptoode kliendile saab alguse hästi toimivast meeskonnast. Pideva ja paindliku arendamise juures on protsessid määrava tähtsusega. Seetõttu on oluline protsesse analüüsida ja parandada nende efektiivsust eesmärkide saavutamiseks.

## 1.1 Taust ja probleem

Ettevõttes on peamiselt kaks meeskonda: arendus ja äri. Kuna arendustiim on üsna väike, alla 10 arendaja, siis võivad ühe inimese lauale maanduda mitmed ülesanded. Selleks, et töövoog sujuks ja erinevad *ticketid* saaksid tehtud, on vaja ühtset süsteemi ja head kommunikatsiooni töötajate-projektijuhi ning tiimide-arendajate vahel. Täna kasutavad mõlemad töökorralduseks peamiselt ühte selleks loodud rakendust nimega Jira.

Jirat kasutatakse mitmel erineval eesmärgil - äritiim klientidelt probleemide tuvastamiseks ning arendustiim oma ülesannete järgimiseks. Mõlemal juhul kasutatakse Jiras peamiselt *board* funktsiooni töökorralduse haldamiseks. Kasutatavaid *boarde* on ettevõttes 5, kuid antud töös keskendutakse peamiselt kahele: äri ehk CS *board* ja arendus ehk SD *board*. Jira *board* on kasulik, kuna see annab hea ülevaate erinevate *taskide* hetkeseisukorrast ja ka sellest, millega erinevad arendajad parasjagu tegelevad. Selle kasulikkus tuleb aga välja ainult sellisel juhul, kui *board* vastab tegelikkusele ja seal on ühtne toimiv süsteem.

Ettevõttesse tööle asudes pani autor tähele mitmeid probleeme seoses arendajate töökorraldusega, mis põhinesid sel ajal veel ainult isiklikul kogemusel, kuid millele lahenduse leidmine oleks kõigi töö tegemise palju efektiivsemaks ja mugavamaks

muutnud. Varasemalt puudus ettevõttes IT-projektijuht ning sellest tulenevalt on arendusmeeskonna töö üsna kaootiline ja ühise eesmärgita. Tiimil ei ole ühtset arusaama äritiimi vajadustest ega prioriteetidest ning sellest tulenevalt puudub kindel tegevuskava.

## **1.2 Eesmärk**

Bakalaureusetöö eesmärk on analüüsida ettevõttes kasutusel olevat arenduse tööprotsessi ja leida kitsaskohad. Antud töös on fookuses peamiselt kolm etappi - nõuete analüüs, projekteerimine ja teostus. Tulenevalt analüüsi tulemustele korrigeerida tööprotsessi ning parandada seeläbi arendustiimi igapäevast tööd. Samuti hinnata projektijuhi kohustuste täideviimist eelmainitud kolme arenduse protsessi vältel.

Eesmärgi saavutamiseks on autor planeerinud täita järgnevad alameesmärgid:

- Kaardistada analüüsi käigus tarkvaraarenduse elutsükli AS-IS olukord ning esitada TO-BE mudel kitsaskohtade parendusega
- Küsitleda arendusmeeskonda olemasolevate puudujääkide tuvastamiseks
- Analüüsida leitud tulemusi ning vajadusel parandada järeldustest leitud kitsaskohad
- Analüüsida projektijuhi kohustusi ja täideviimist põhinedes eksisteerivatele projektijuhi meetodikatele
- Võrrelda läbiviidud muudatustest tingitud olukorda algse tööprotsessiga

## **1.3 Töö struktuur**

Antud töö järgnevates peatükkides räägitakse eelnevalt mainitud rakendusest ja selle võimalustest lähemalt. Tutvustatakse üldiselt ka tänapäeval väga populaarset tarkvaraarenduse juhtimise tehnikat, milleks on agiilse arenduse juurde kuuluv Kanban. Peale teoreetilist osa jõutakse töö põhiosani ehk uurimuse ja selle tulemusteni. Tuuakse välja täpsemad küsitluses selgunud arvamused ning selle põhjal läbiviidud analüüsi tulemused. Peale seda tehakse ettepanekud vastavalt analüüsile muudatuste kohta protsessi parendamiseks ning võimalusel ka implementeeritakse. Lisaks kirjeldatakse

projektijuhtimise alustalaid ning vastavalt nendele analüüsitakse projektijuhi tööülesandeid.

## 2 Metoodika

Käesolevas peatükis tutvustatakse bakalaureusetöö käigus kasutatavaid metoodikaid ja tööriistu. Antakse ka ülevaade ettevõtte olemusest. Valdav osa läheb töös kasutatavate teoreetiliste osade kirjeldamiseks ning nende põhimõtete tutvustamiseks.

### 2.1 Ettevõtte ja meeskonna lühitutvustus

Ettevõtte A pakub loomingulist haldustööriista, millega saavad meedia-ja digiagentuurid üle maailma luua kaasahaaravaid digitaalseid reklaamikampaaniaid, mis on efektiivsed ja tõhusad. Olgugi et ettevõttel leidub kliente üle maailma, siis suurimateks turgudeks on Ameerika ja Aasia. Ettevõtte A digilahendusi kasutavad tuntuimad ettevõtted välismaal on: Apple, Netflix, Disney, Coca Cola, McDonalds, Mercedes Benz, Nike, Tiffany & Co jpt.

Ettevõttes töötab umbes 18 inimest, kellest IT-ga on seotud 8 inimest. Lisaks projektijuhile kuulub arendustiimi veel 8 inimest: IT-juht, 5 arendajat, 1 testija ning UI/UX disainer. Arendajad saab omakorda jagada *full-stack*, *front-end*, *framework* ja infrastruktuuri arendajateks. Töömahu pideva suurenemise tõttu on suur vajadus just arendusmeeskonna laienemise vastu.

Peale arendusmeeskonna on ettevõttes ka äritiim, kelle peamine fookus on klientidega suhtlemisel ja nende abistamisel platvormis. Igale äritiimi töötajale kuulub tavaliselt kindel arv kliente, kellega toimuvad nii videokõned kui ka ollakse probleemide korral suhtluses Jiras. Samuti on nende vastutusallas ka klientide soovide edastamine arendustiimile.

### 2.2 Tööriistad

#### 2.2.1 Jira

Jira on Atlassian ettevõttele kuuluv agiilsete tööhalduslahenduste komplekt. Jira aitab meeskondadel tööd planeerida, määrata, jälgida ja hallata ning toob meeskonnad kokku

kõige jaoks, alustades agiilsest tarkvaraarendusest ja klienditoest kuni idufirmade ja ettevõtteni. Jira Software on agiilsete meeskondade kõige populaarsem tööriist. Agiilseid meetodikaid praktiseerivatele meeskondadele pakub Jira Software Scrum ja Kanban *boarde*. *Boardid* on ülesannete haldamise keskus, kus ülesanded kaardistatakse töövoogudena. *Boardid* tagavad meeskonnatöö läbipaistvuse ja iga tööüksuse oleku nähtavuse. Aja jälgimise võimalused ja reaajas tegevusaruanded võimaldavad meeskondadel oma tootlikkust aja jooksul tähelepanelikult jälgida. [1]

### **2.2.2 Kanban**

Kanban meetodika sai alguse juba rohkem kui viiskümmend aastat tagasi ühes autotootmisettevõttes nimega Toyota. 1940. aastate lõpus hakkas Toyota kasutusele võtma toidupoodides kasutatavat meetodit, mis tähendas paremat varude haldamist ja nende täiendamist sellises koguses, et rahuldada tarbijate nõudlust. Kuna laovarude tase ühtis tarbimisharjumustega, siis tõhustas see oluliselt varude haldamist, varasemalt igal ajahetkel hoitud üleliigne laovaru vähenes. Siiski oli tagatud, et vajalikud tooted oleksid alati olemas. Toyota hakkas samal põhimõttel kasutama meetodit, kus töötajad jagasid tiimide vahel kaarte, et edastada vastava materjali puudust. Kui tootmises oli materjal otsa saanud, anti kaart edasi lattu, kus neid uute materjalidega varustati. Kui ka laos olid materjalid otsa saanud, siis anti kaart edasi varustajale, kes laovarusid täiendas. Kuigi selline käitumisviis on alates 1940ndatest edasi arenenud, siis tootmisprotsess nimega "*just in time*" (JIT) on endiselt selle keskmes. [7]

Kanban on tänapäeva üks kõige populaarsemaid tehnikaid, mida kasutatakse paljudes agiilse tarkvaraarendusega tegelevates meeskondades. Kanban arvestab tiimi võimekusega, andes neile paindlikumad planeerimisvõimalused, kiirema väljundi, selgema fookuse ja läbipaistvuse kogu arendustsükli vältel. Erinevalt kanbani rakendamisest tehases, ei ole tarkvaraarenduse meeskondadel vaja protsesside muudatusi ega lisanduvaid materjale, piisab ainult tahvlist ja kaartidest, aga ka need võivad olla virtuaalsed. [7] Peamisest tööriistast Kanban *boardist* tuleb lähemalt juttu juba järgmises peatükis.

### **2.2.3 Kanban board**

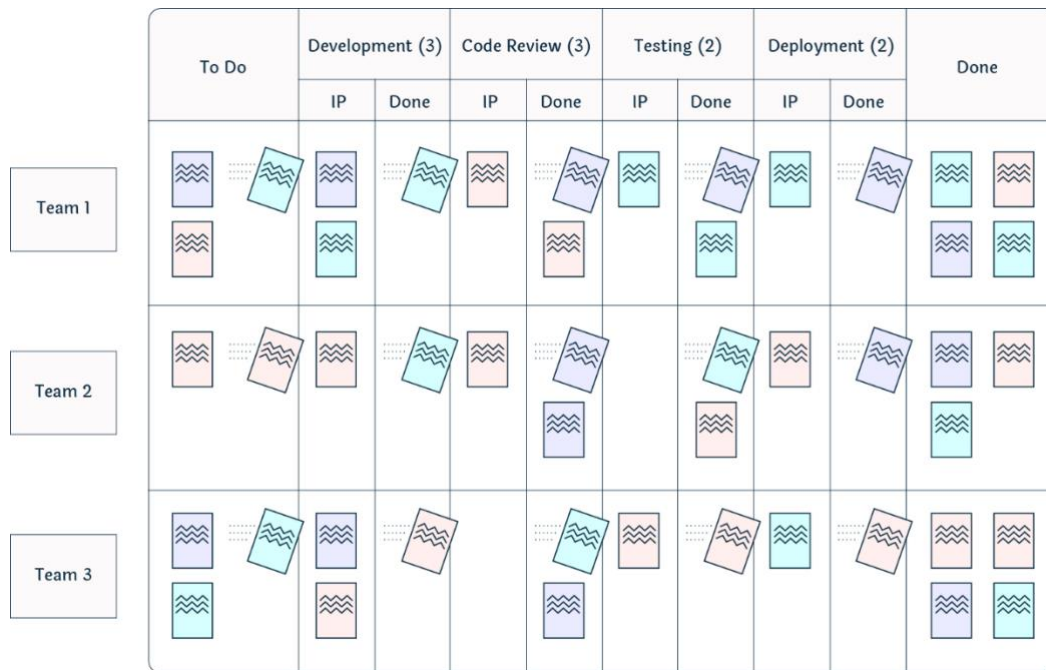
Kanban *boardi* ülesanne on tagada meeskonna töö visualiseerimine, sealne töövoog on standardiseeritud ning kõik segajad ja sõltuvused tuvastatakse ja lahendatakse kohe. [7]

*Board* koosneb tulpadest ning *taskidest* ehk ülesannetest. Tulpad jaotavad töövoogu erinevateks etappideks ning ülesanded liiguvad nende vahel vasakult paremale, kui nende staatus protsessis muutub. Iga ülesanne määratakse vastavale meeskonnaliikmele koos täiendava info ja materjalidega. [8] *Board* ehk tahvel võib olla nii füüsiline kui ka virtuaalne.

Kõige lihtsam variant kanban *boardist* on füüsiline tahvel, mis on jagatud veergudeks. Inimesed märgivad oma ülesanded märkmepaberitele ning asetavad need tahvlile, kus need liiguvad läbi töövoogu näidates edusamme. Digitaaltahvlid võimaldavad aga meeskondadel, kes ei jaga füüsilist kontoripinda, kasutada kanbani tahvleid eemalt ja asünkroonselt. Seal märgitakse ülesanded internetipõhiselt virtuaalsetele kaartidele, mida saab tulpade vahel liigutada. Digitaaltahvli eelis on ka erinevate lisalistide loomine, nagu “*backlog*” või “*up next*”. Pole oluline töötajate asukoht või ajahetk millal *boardi* vaadatakse, neile on alati kättesaadav kõige ajakohasem olek ülesannetest. [6]

Põhilisel kanban *boardil* on kolmeastmeline töövoog: *To Do*, *Progress* ja *Done*. Sõltuvalt meeskonna suurusest, struktuurist ja eesmärkidest saab töövoogu siiski kaardistada, et see vastaks konkreetse meeskonna ainulaadsele protsessile. [7]

Tarkvaraarenduses on veerge tavaliselt siiski rohkem, kuna protsessid on kolmeastmelisest arendusest keerukamad. [9] Nagu Joonisel 1 on näha, siis võib töövoog kujuneda ka selliselt: *backlog*, arendus, koodi ülevaatamine, testimine, töö avaldamine ja sulgemine. Selliste keerukate protsesside puhul võib ette tulla ka ülesannete liikumine töövoos “tagasi”, näiteks võib testimise käigus selguda viga ning ülesanne liigub tagasi arendusse.



Joonis 1. Kanban board [10]

Tahvli tulpadele on võimalik määrata ülesannete koguse limiit, mis võib igal ajahetkel ühes veerus olla. Kui limiit on täis, peavad ülesanded liikuma järgmisesse veergu enne kui uued asemele saavad tulla. Sellised piirangud on olulise tähtsusega töövoo kitsaskohtade leidmiseks ja sujuva arenduse maksimeerimiseks. Piirangud võivad ka hoiatada liigse töömahu eest. [6]

## 2.3 Protsess

### 2.3.1 Projektijuhtimise metoodika PMBOK'i baasil

Projekti käivitamiseks on vaja protsesside süsteemi, olenemata selle suuruselt ja ulatusest. Alates piasjade planeerimisest ja klientide pidevalt muutuvate nõudmiste rahuldamisest kuni tarnete õigeaegse kohaletoimetamiseni – on palju, mis võib valesti minna. Kui jagada projekt juhitavateks etappideks, millest igaühel on oma eesmärgid ja tulemused, on projekti ja väljundi kvaliteeti lihtsam kontrollida. Antud töö fookuses on välja toodud PMBOK, kuna see on kõige lähedasem projektijuhtimismetoodika ettevõttes A eksisteerivale metoodikale. Projektijuhtimise etapid jagunevad PMBOKi alusel viieks: algatus, planeerimine, täideviimine, järelevalve ja kontroll ning sulgemine. [11]



## 5 Basic Phases of Project Management



kissflow

Joonis 2. Projektijuhtimise etapid [11]

Algatamine on elutsükli esimene etapp, kus muudetakse abstraktne idee mõtestatud eesmärgiks. Selles etapis määratletakse projekti laiemalt ja arutletakse küsimuste üle nagu, mis on projekti mõte, miks seda vaja on ja kuidas on võimalik seda teostada. Selle etapi kõige olulisem osa on jõuda kokkuleppele kõigi projekti huvirühmadega enne, kui suundute järgmise etapi juurde. Selle lihtsustamiseks ja ideede koondamiseks pannakse kirja projekti üldine kava. [11], [12]

Järgmiseks sammuks on planeerimine, et välja mõelda, kuidas täpselt projekt ellu viia. Ilma agiilset tarkvaraarendust kasutamata võib see faas võtta kuni poole kogu projekti ajast. Planeerimise etapis on peamised ülesanded tehniliste nõuete väljaselgitamine,

üksikasjaliku projekti ajakava koostamine, kommunikatsiooniplaani loomine ja eesmärkide/tulemuste seadmine. Edukas täideviimise faas tugineb edukale planeerimisfaasile. See etapp on koht, kus arendusmeeskond teeb tegelikku tööd. Projektijuhi ülesanne luua tõhus töövoog ja hoolikalt jälgida meeskonna edusamme. Projektijuhi teine kohustus on järjepidevalt säilitada tõhus koostöö projekti sidusrühmade vahel. See tagab, et kõik jäävad samale lehele ja projekt kulgeb sujuvalt ja probleemideta. [11]

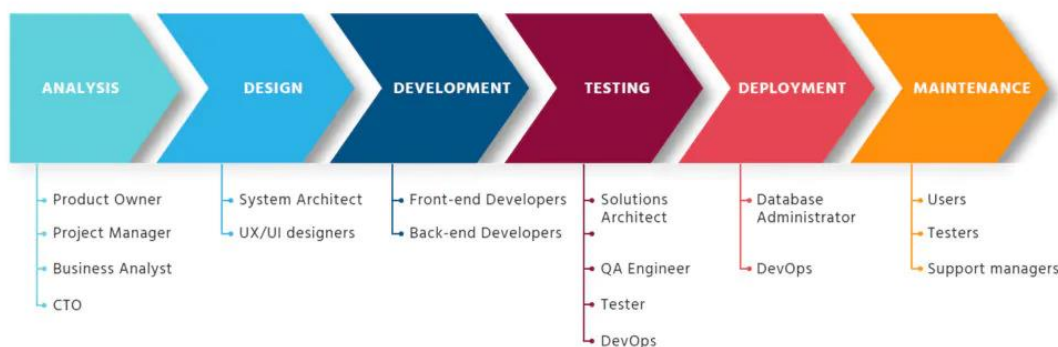
Parim viis arenemise ja täiustamise tagamiseks on projekti jõudluse jälgimine ja ülevaatamine. Teostuse faasiga paralleelselt jälgib projektimeeskond samal ajal ka projekti kulgu, lähtudes varem koostatud projektiplaanist. Projekti toimivuse jälgimine erinevate mõõdikute läbi on väga oluline tagamaks, et projekt püsiks ajakavas ning eelarve ja ulatuse piires. Nii täideviimine kui järelvalve ja kontroll on kriitilised etapid, mis võivad määrata projekti edu. Lisaks ülesannete jälgimisele püüab projektijuht tuvastada ka probleeme või riske ja annab sidusrühmadele regulaarselt aru projekti seisust. [13]

Sulgemine on projektijuhtimise protsessi viimane etapp. Projekti sulgemise etapp näitab projekti lõppu pärast lõplikku tarnimist. Mõnikord palgatakse ka väliseid spetsialiste spetsiaalselt projekti lepingu jaoks. Nende lepingute lõpetamine ja vajalike paberite vormistamine on samuti projektijuhi kohustus. Samuti enamik meeskondi korraldab pärast projekti lõppu retrospektiivi, et arutada oma õnnestumisi ja ebaõnnestumisi projekti jooksul. See on tõhus meetod pideva arengu tagamiseks, et tõsta meeskonna üldist tootlikkust tulevikus. [11]

### **2.3.2 Arendusprotsess**

Tarkvaraarendus saab alguse suure pildi nägemisest, seega enne toote valmistamist on mõistlik vaadata üle levinuimad arenduse elutsüklid, et mõista nende protsesside toimimist. Sarnaselt projektijuhtimise protsessile saab ka tarkvaraarenduse elutsüklit erinevateks etappideks jagada. Tavaliselt jaguneb see kuueks kuni kaheksaks etapiks, nagu näha ka Joonisel 3: analüüs, disain, arendamine, testimine, juurutamine, ülalhoid.

## 6 PHASES OF THE SOFTWARE DEVELOPMENT LIFE CYCLE



Joonis 3. Tarkvaraarenduse elutsükk [12]

Analüüsi eesmärk on koguda nõudeid ning defineerida tarkvaraarenduse protsessi suund. Analüüsifaasis tehtud vead on hiljem kõige kallimad parandada. See on faas, kus luuakse vundament oma tulevasele tootele, näiteks analüüsitakse konkurente ja luuakse ärieesmärke. Oluline on arvestada kliendipoolse meeskonnaga, kelle põhjalikud teadmised toote eesmärkide, nõuete ja tööstuse kohta aitavad kaasa üldisele strateegiale ja muudavad analüüsi etapi tõhusamaks. [12]

Järgmises etapis ehk disainifaasis muudetakse nõuded ümber detailseks tarkvara arhitektuuriks. Tootedisaini saab jagada kolmeks komponendiks: funktsionaalsus, välimus, kvaliteet. Funktsionaalsuse eest vastutab arhitekt. Oma fookuse kitsendamiseks aitab ta äri- ja arendustiimil paremini omavahel töötada toote nimel, mis täidab nii kliendi kui äri nõudmisi. Välimusega tegelevad UX disainerid, kes kavandavad jooniseid ja teevad prototüüpe. Kvaliteet seisneb toote vastavusest kliendi vajaduste ja ootustega. [12]

Arendusetapis tehakse disain ümber koodiks. Koodi kirjutamine jagatakse erinevate arendajate vahel ära vastavalt nende oskustele. Front-end arendajad tegelevad rakenduse või toote kasutajaliidese kuvamisega ja elementide loomisega, mida kasutajad vajavad veebilehega suhtlemiseks. Back-end arendajad vastutavad kogu toote tehnilise ja süsteemse poole eest. Hästi kirjutatud kood vähendab tulevikus testimist ja hooldusega seotud probleeme. [12]

Testimise eesmärk on teha kindlaks, kas valminud toode siiski vastab nõuetele ning on vigadeta. See hõlmab tarkvara käivitamist kasutades ühe või mitme huvipakkuva omaduse hindamiseks käsitsi või automatiseeritud tööriistu. [14]

Tarkvara juurutamine on protsess, mille käigus antakse valmis tarkvara üle kliendile. Juurutamine toimub alles pärast põhjalikku testimist tagamaks, et kõik vead on tuvastatud ja parandatud. Erinevad sammud ja tegevused, mis selles etapis tehakse sõltuvad ettevõttest ning arenduse fookusest. [15]

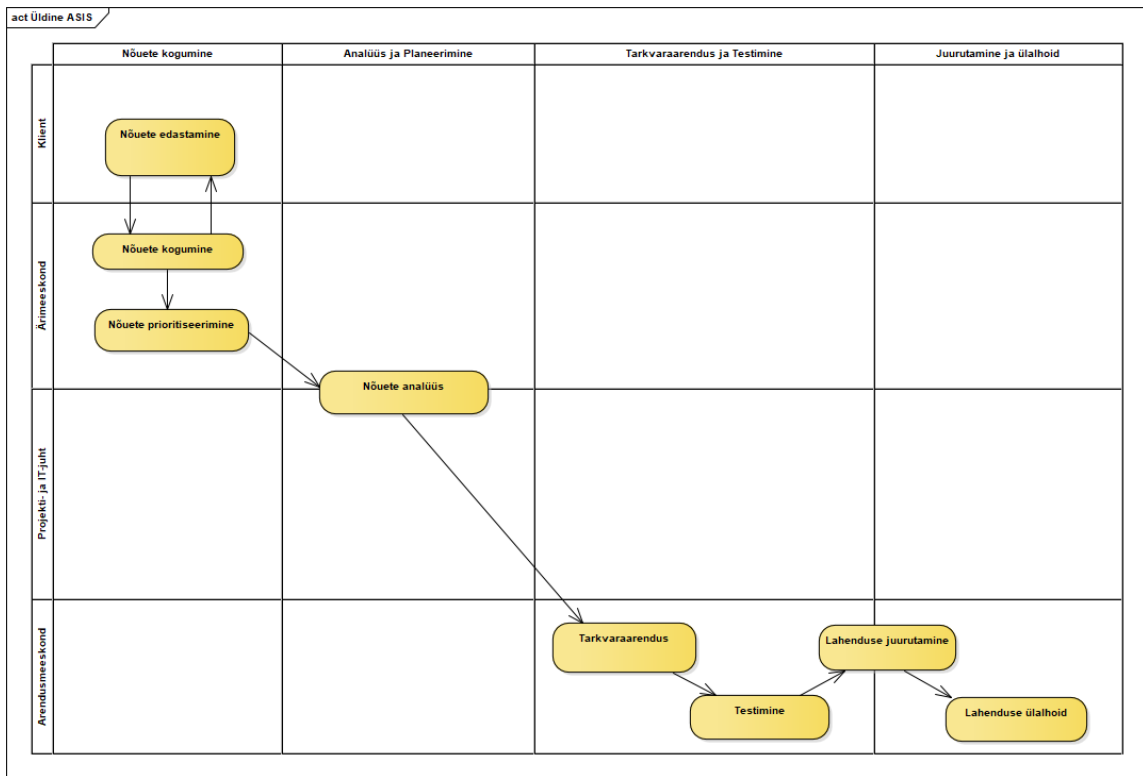
Ülalhoiu eesmärk on tagada hea kasutajakogemus, kus pideva protsessi käigus tagatakse stabiilne ja relevantne tarkvara. Probleemide ning vigade ilmnemisel parandab hooldustiim need tulevases versioonis. Ülalhoiu faas jaguneb kaheks: korrigeeriv ja kohanduv. Korrigeerivas lahendatakse probleeme, mis tekkisid arenduse käigus ning kohanduv ülalhoiid tähendab lisanõudeid, mida esialgses plaanis ei olnud. [12]

### 3 Tarkvaraarenduse protsessi kaardistamine

Käesoleva peatüki eesmärgiks on kirjeldada ettevõtte tarkvaraarenduse protsessi hetkeolukorda (AS-IS). Peamised protsessis osalevad osapooled on projektijuht, IT-juht, arendusmeeskond, äritiim ja klient. Peatükis kirjeldatav ideaalprotsess ei pruugi igapäevaelus alati selliselt kujuneda, kuna tuleb ette ootamatusi ja eriolukordi, kuid see on mudel, mida valdavalt järgitakse. Ettevõttel on kirja pandud erinevad dokumendid erinevate protsesside juhtimisest, samuti ka tarkvaraarendusest. Antud töö osas kõige asjakohasem dokument määrab kindlaks tarkvaraarenduse nõuete kogumise, täitmise, vastutuse ja rollide jaotuse põhinõuded ettevõttes. Siiski on antud dokument kirjutatud aastaid tagasi ning kogu sealne info pole tänapäeval enam relevantne, on toimunud mõningaid muudatusi nagu vähenenud meeskond ja scrumilt üleminek kanbanile.

Ettevõtte A tarkvaraarenduse tervikprotsessi võib vaadata käsitlevas töös kui nelja alamprotsessi:

1. Nõuete kogumine
2. Analüüs ja planeerimine
3. Tarkvaraarendus ja Testimine
4. Juurutamine ja Ülalhoid



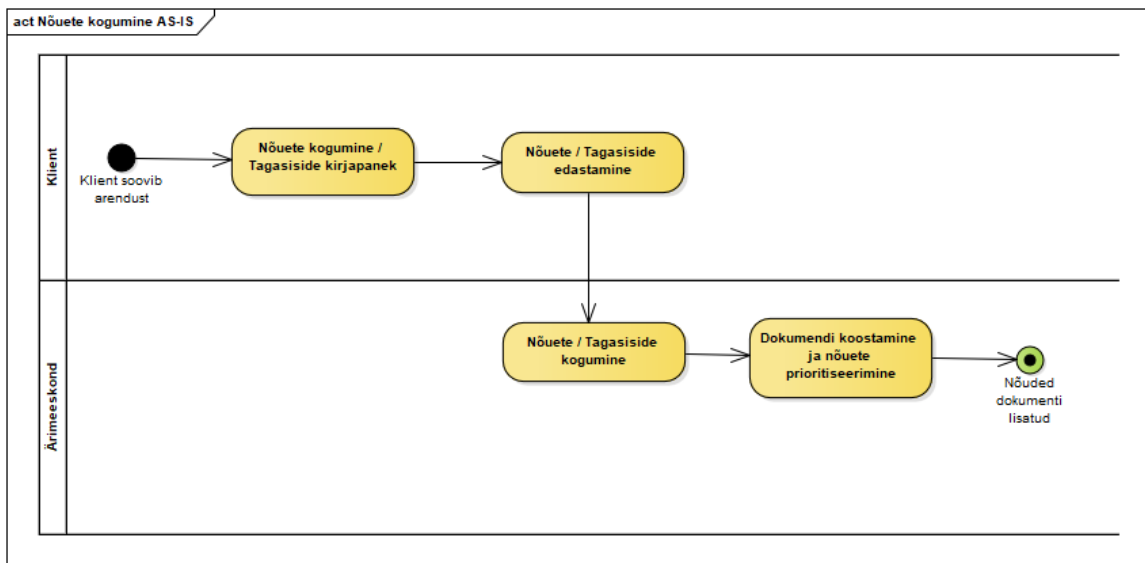
Joonis 4. AS-IS: Ettevõtte A tarkvaraarenduse alamprotsessid

Järgnevatel alampeatükkidel eesmärgiks on kirjeldada ettevõtte kehtivat tarkvaraarenduse protsessi lähtudes eelpool mainitud kaardistatud protsessi etappidest. Tuuakse välja tarkvaraarenduse protsessi alamprotsesside hetkeolukorra (AS-IS) joonised ning äriprotsessi selgitused, mis sisaldavad tegevuste kirjeldusi, teostajaid ning tulemeid.

### 3.1 Nõuete kogumine

Äritiimi töötajatel on kõigil omad kliendid, kellega ollakse pidevas iganädalases suhtluses. Kliendid annavad tagasisidet ja jagavad uusi ideid, mida saaks platvormil paremini teha või mis uuendusi neil vaja oleks, et mõlemale ettevõttele rohkem kasu teenida. Kui mitmel kliendil on sarnased soovid ning nende tulemusena kasvaks ka ettevõtte A kasum, siis kogutakse kõik nõuded kokku. Seejärel prioritseerib äritiim nõuded, et need anda edasi projektijuhile ja arendusmeeskonnale. Uued arendused võivad saada alguse ka klientide probleemidest, millest antakse märku Jiras või e-maili teel. Sellist tüüpi *ticketeid* hoitakse *CS boardil*. Sagedastest klientide probleemidest kujunevad tihti suuremad muudatused üle terve platvormi, näiteks muudetakse analüütika

eksportimist kiiremaks. Kogutud nõuded koos prioriteetidega paneb äritiim kokku ühte dokumenti, mis seejärel edastatakse IT- ja projektijuhile.

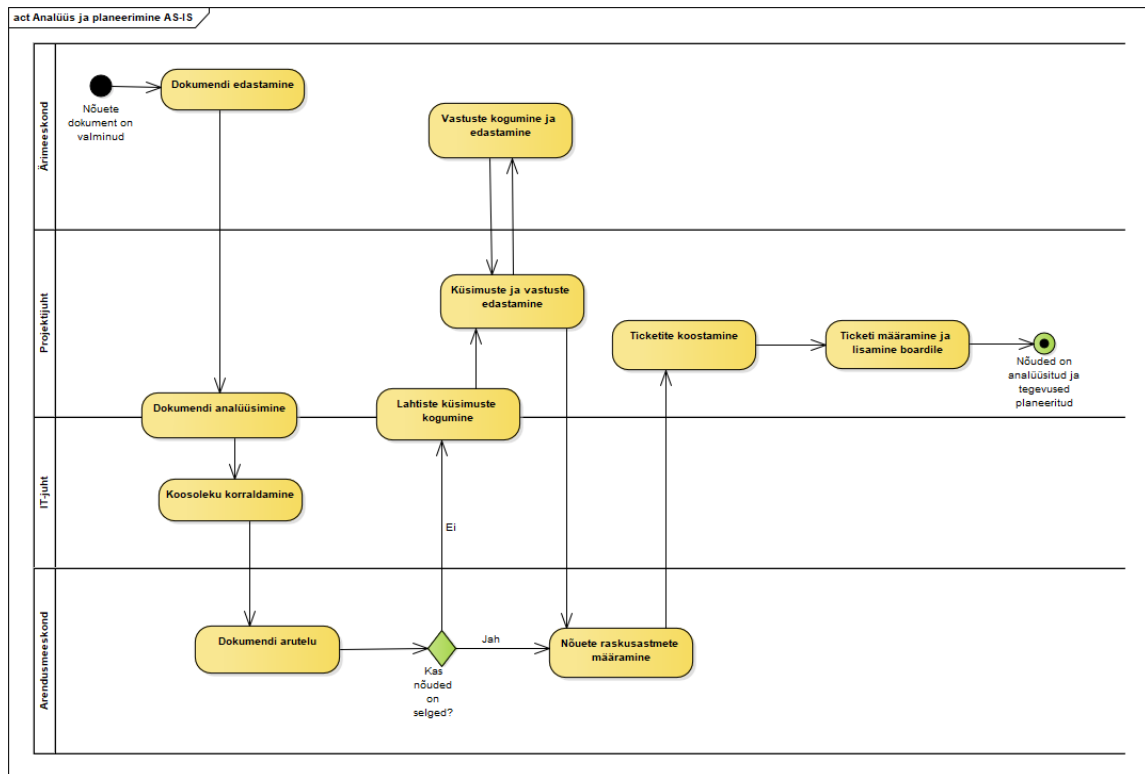


Joonis 5. AS-IS: Nõuete kogumine

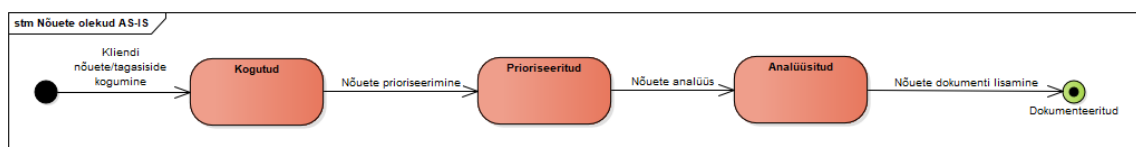
### 3.2 Analüüs ja planeerimine

Analüüsi käigus saavad IT- ja projektijuht dokumendi klientide nõuetega, mida edaspidi arendustiimi plaanidesse kaasata. Protsessi efektiivsemaks muutmiseks planeerib IT-juht koosoleku koos arendusmeeskonnaga, et dokument üle käia, teemad läbi arutada ning küsimused kirja panna. Seal arutatakse ka läbi erinevate teemade raskusastmed ja kas oleks veel vaja täiendavat uurimist arenduse poolel. Koosoleku tulemusena on kogu arendusmeeskond ühel järjel edasiste plaanidega ning projektijuht saab tekkinud küsimustega pöörduda tagasi äritiimi poole, kes siis vajaduse korral saab suhelda kliendiga. Kui nõuded on selgust leidnud ja konkreetsed ootused on paigas, siis pannakse need kirja Jira *ticketiks System Development* (edaspidi SD) *boardil*. Kui ülesanne nõuab disaini sisendit, tehakse *ticket* ka Design *boardile*, mis seotakse ära samateemalise SD *ticketiga*. Jira SD *ticketid* jagunevad kolmeks: *story*, *task* ja *bug*. *Story* kujutab endast suuremat eesmärki kasutaja probleemi lahendamiseks, samas kui *task* esindab selle jaoks tehtavaid vajalikke üksikuid samme. *Bug* väljendab milline oli halb või ootamatu lõpptulemus kasutaja vaatenurgast. Kui projektijuht on saanud *ticketid* kirja koos vajaliku infoga, hakatakse neid jooksvalt vastavalt vajadusele töösse võtma. Arenduses alustatakse kõrge prioriteediga ülesannetega, kuid kuna need võivad olla kerged ehk lühikese ajaga tehtavad või rasked ehk nädalate pikkuse tööajaga, siis pannakse arenduse

lauale erineva prioriteediga ülesandeid, et tööaegade pikkused varieeruksid ja kliendid uuendusi pikalt ootama ei peaks. Selle tulemusena tehakse vaheldumisi kergeid ning raskeid kõrge prioriteediga ülesandeid ja samuti ka keskmise prioriteediga kergeid ülesandeid. Töösse võetavad *ticketid* pannakse kanban *boardil* esimesse veergu nimega “*Ready for development*” ning neile määratakse vastav arendaja.



Joonis 6. AS-IS: Analüüs ja planeerimine



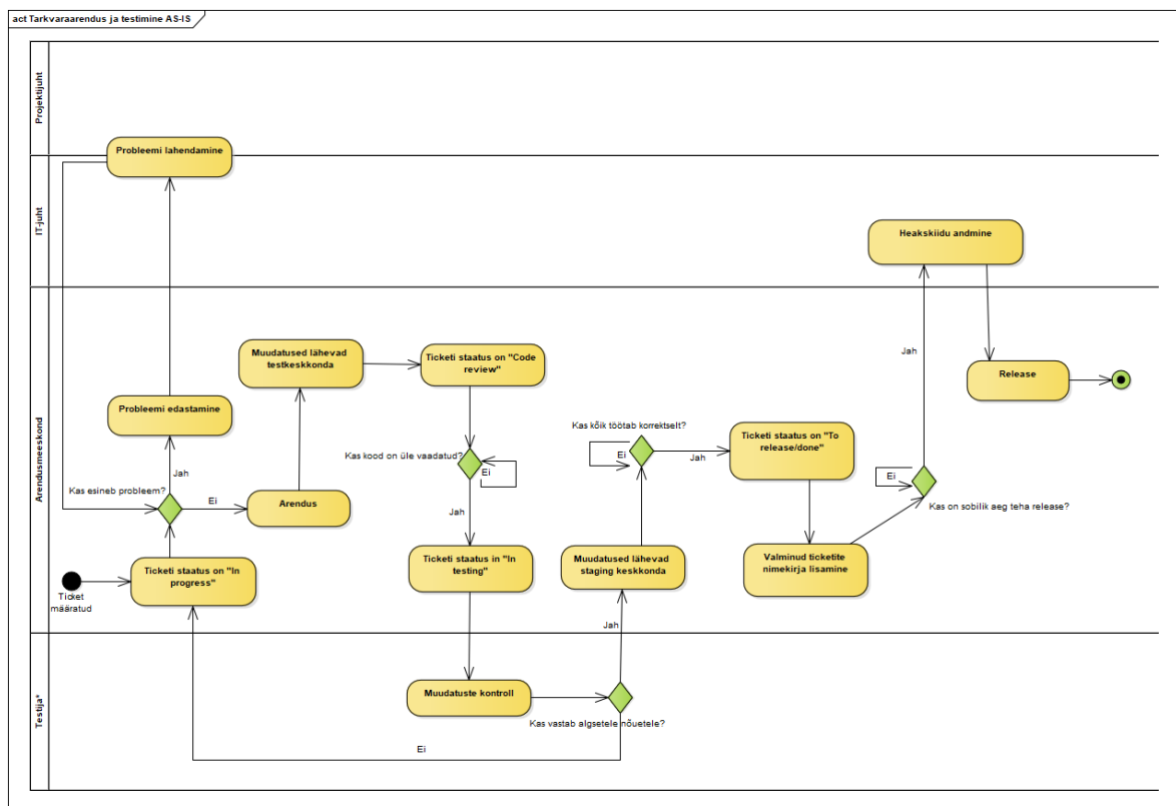
Joonis 7. AS-IS: Nõuete olekud

### 3.3 Tarkvaraarendus ja testimine

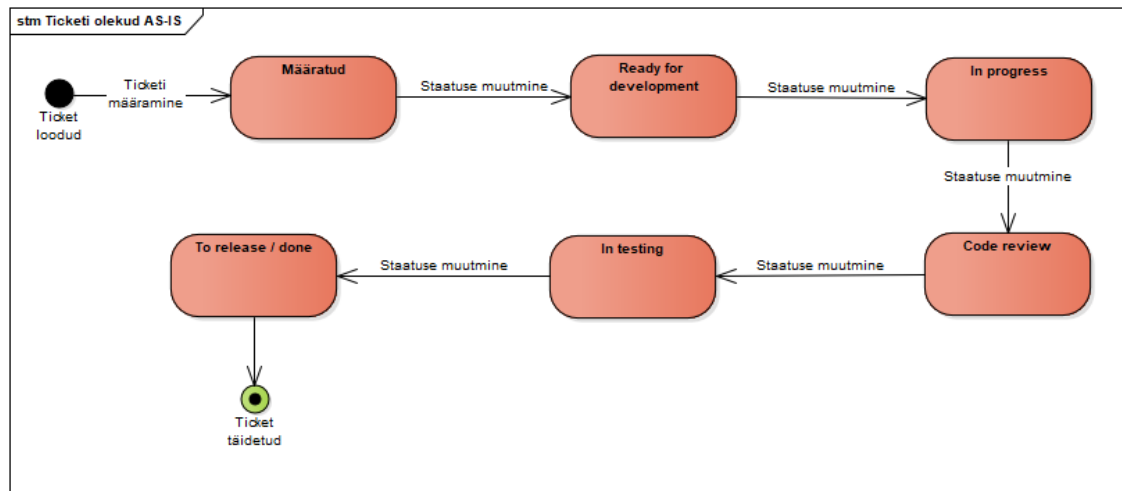
Kui *ticket* asub kanban *boardil* ja arendaja näeb, et see on talle määratud, saab ta alustada selle ülesande täitmist. Kui on vajalik ka disainipoolne sisend, siis oodatakse see ära. *Ticket* liigutatakse järgmisesse veergu nimega “*In progress*”. Ülesannete täitmine sõltub nende mahust ja raskusastmest, need võivad aega võtta alates kümnest minutist kuni mitmete nädalateni. Arendustiimil toimuvad igapäevased *stand-upid*, mis on 10-15



minutilised koosolekud, kus räägitakse päevastest tegevustest ja saab arutada erinevaid muresid või probleeme, mis kellelgi on tekkinud. Kui laiemal ringil küsimusi saab arutada *stand-upil*, siis tehnilisema probleemi korral pööratakse IT-juhi poole, ülesande lisainfo tuleb projektijuhi käest. Kui *ticket* on arendaja poolt valmis saanud, laetakse kood üles testkeskkonda ning *ticket* tõstetakse edasi “Code review” veergu, kus teine arendaja vaatab tehtud koodi üle. Seejärel jõuab *ticket* “In testing” tulpa, kus testija kontrollib üle, kas arendaja poolt tehtud muudatused vastavad *ticketis* kirja pandud nõuetele. Kui juhtub, et testija leiab mõne vea, siis edastab ta selle arendajale *ticketi* kommentaariumis ning *ticket* liigub tagasi “In progress” tulpa. Seejärel läheb kood üles *staging* keskkonda, mis on suures osas identne *live* keskkonnaga ning seal vaadatakse, et kogu tarkvara koos uuendustega töötaks korrektselt. Kui mõlemas keskkonnas on *ticketi* sisu ära testitud, siis liigub *ticket* viimasesse *boardi* tulpa “To release/Done” ehk ülesanne on valmis klientidele edastamiseks ning platvormile minekuks. Kui “To release/ Done” veergu on kogunenud hulk *ticketid* ja IT-juht koos arendusmeeskonnaga otsustab, et oleks sobiv aeg teha *release*, siis kommunikeeritakse vastav info üle ka äritiimiga, et saada nende heakskiit.



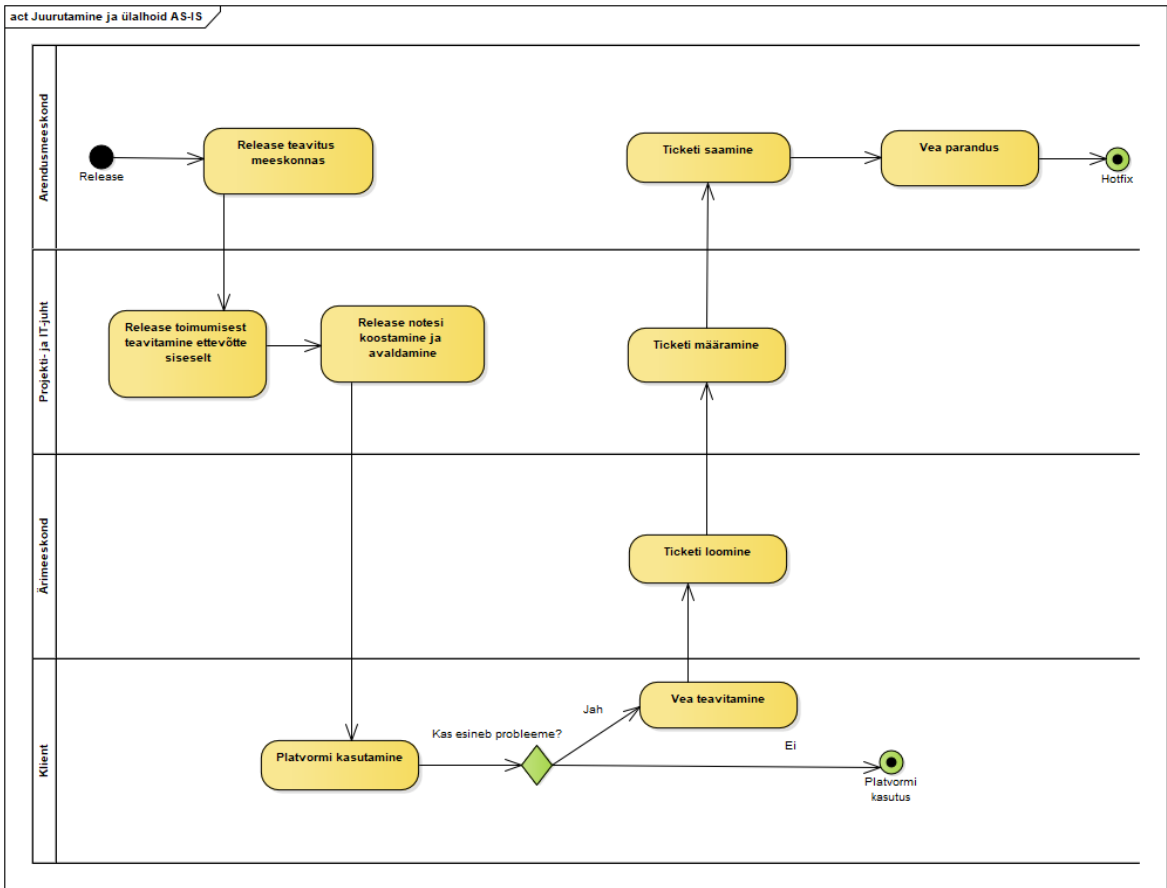
Joonis 8. AS-IS: Tarkvaraarendus ja testimine



Joonis 9. AS-IS: Ticketi olekud

### 3.4 Juurutamine ja ülalhoold

Peale *release'i* toimumist teavitab IT- või projektijuht esmalt äritiimi viimastest peamistest muudatustest. Seejärel koostab projektijuht dokumendi ehk *release notesi*, kuhu pannakse samuti kirja peamised muudatused, mis antakse üle turundusele, kes koostavad antud infost ilusa illustratiivse artikli. *Release notes* läheb üles kodulehele ning see annab kasutajatele infot viimaste uuenduste kohta koos õpetustega. Pärast uute muudatuste platvormile jõudmist võivad siiski klientidelt ilmnedä mõningad probleemid, mis arenduse ja testimise käigus ei selgunud. Olgu klientide probleemid seotud, kas hiljutise relüisi või muude vigadega platvormis, lähevad need kõik kirja Jira *Customer Support* (edaspidi CS) *boardile ticketitena*. Kui keegi aga arendusmeeskonnast tuvastab mõne vea, tehakse sellest *bug ticket SD boardil*. Ettevõttes on kaks *full-stack* arendajat, kes vaheldumisi üle nädala tegelevad klientitoe *ticketitega*. Siiski läheb aeg-ajalt vaja ka *front-end* või *framework* arendajat. Äritiim määrab klientitoe *ticketid* projektijuhile, kes siis omakorda määrab need vastavalt probleemile õigele arendajale.



Joonis 10. AS-IS: Juurutamine ja ülalhoold

## 4 Läbiviidud küsitluse tulemused

Arendustiimi töökorralduse konkreetsete probleemideni jõudmiseks viidi meeskonnas läbi uuring. Käesolevas peatükis tutvustatakse ettevõttes A läbiviidud küsitlust, mille täpne struktuur on Lisas 2, ja selle tulemusi. Küsitlus saadeti välja 6 inimesele arendusmeeskonnas ning vastused saadi kõigilt. Küsitluse edastamisest jäeti välja üks arendaja, kelle töö ei ole arendustiimi ega Kanban *boardiga* niivõrd seotud. Tabelis 1 on vastanud isikud jaotatud nende ametite järgi.

Tabel 1. Vastanute arvukus ja positsioon

Positsioon ettevõttes	Vastanute arv
Disainer	1
Full-stack arendaja	2
Frameworki arendaja	1
Front-end arendaja	1
Testija	1

Anonüümne küsimustik viidi läbi Google Forms keskkonnas ja see oli jaotatud kolme osasse:

1. Küsimused igapäevatöö kohta. Sammud, mida läbivad protsessi vältel ning mis on probleemid, mis võivad takistada efektiivset töötegemist.
2. Küsimused Kanban *boardi* kohta. Kui palju seda kasutatakse, kas vastab tegelikkusele ning probleemid seoses tööülesannete saamise ja seisu jälgimisega.
3. Küsimused organisatoorse poole kohta. Mida arvatakse koosolekutest, millised on eesmärgid, kas mõistetakse erinevate osapoolte vajadusi, projekti- ja IT-juhi hindamine. Lõpetusena ettepanekud protsessiparenduseks.

## 4.1 Arendusprotsessi kirjeldus ja probleemid

Kuuest vastanust pooled töid *taski* valmimise protsessis välja *ticketi* lugemisele järgneva lisaarutelu kolleegidega. Peale esmast *ticketi* analüüsi suheldakse kas tiimisiselt või pöördutakse ka lausa äritiimi poole, et ühine arusaam oleks ülesandest. Kui *ticketit* lugedes tekib küsimusi või ettepanekuid, siis töökaaslastega suheldes saadakse nende osas selgusele. Kaks vastanut töid lisaks välja, et esialgu on edasi-tagasi kommunikeerimist rohkem, arendajal võib endal mõttes olla teine lahendus kui *ticketis* kirjas või mõni ettepanek lahenduseks ning disaineril on vaja tegevuse ulatus ja detailid teiste tiimidega üle arutada. Disaineri puhul võib edasi-tagasi suhtlus jätkuda ka *ticketi* analüüsile järgnevas arendusperioodis, kus tehakse esialgsed kavandid, millele on vaja kinnitust saada erinevatelt osapooltelt. Muudatuste ettepanekul läheb töö uuesti ümber tegemisele ning seetõttu võib kogu protsess võtta kaua aega. Lisaks tõi üks vastanu ka välja ka arendusprotsessis endale *bullet pointide* kirja panemise, et töökäik täpsemalt välja mõelda. Ideaalis luuakse Jirasse *ticketi* alla *subtaskid* ehk alamtegevused, kuid see võib osutuda mitmel põhjusel keerukaks.

Kuuest vastanust viis tõi igapäeva protsessi probleemina välja prioriteetide puudumise. Selle lahendusena kas uuritakse eelisjärjekord ise üle, mõeldakse milline tegevus suurimat väärtust pakub või suurima olulisusega on või määrab prioriteedi kliendipoolne ajakriitilisus. Kuna osad arendustiimi liikmed teevad tööd ka teiste osapoolte jaoks, siis võivad vahepeal SD *boardi* ülesanded olla vähem tähtsamad. Probleemina töid pooled vastanud välja ka kommunikatsiooni. Ei ole teada konkreetne plaan tulevikuks ning on palju teadmatust, selles osas pakkus üks inimene lahendusena välja *sprint planning* koosolekud. Lisaks toodi välja, et varasemad *ticketid* on olnud liiga vähese, vananenud või vale infoga. *Ticketite* valesti määramist on pigem vähe, kuid ka seda on ette tulnud. Disainer tõi välja, et tihti ootavad asjad pikalt kellegi teise taga, enne kui saab protsessiga edasi minna. Mainiti ka ajapuudust ning *boardi* mittevastavust tegelikkusele. Kui kõrvutada mitu erinevat *boardi*, siis saaks mingi aimduse arendajate tegevustest, aga eks tehakse ka nende väliselt ülesandeid. Full-stack arendaja probleemiks oli ka see, et kui klientoe ja tavalise arenduse nädalad vahetuvad kordamööda, on arendusnädalal raske 100% sellele keskenduda, kuna segavaid faktoreid on palju. Õhku on jäänud kas eelneva nädala teemad, tekivad uued kiireloomulised asjad või muud küsimused Slacki kaudu.

Kokkuvõtlikult on igapäevatöoga seotud probleemid:

1. Kommunikatsioon
2. Info puudulikkus *ticketis*
3. Ülesannete prioritseerimatus erinevate osapoolte vahel
4. Ajapuudus
5. *Boardi* mitte vastamine tegelikkusele
6. Teiste osapoolte taga ootamine
7. Valesti määratud *ticket*
8. Liigselt keskendumist segavaid faktoreid

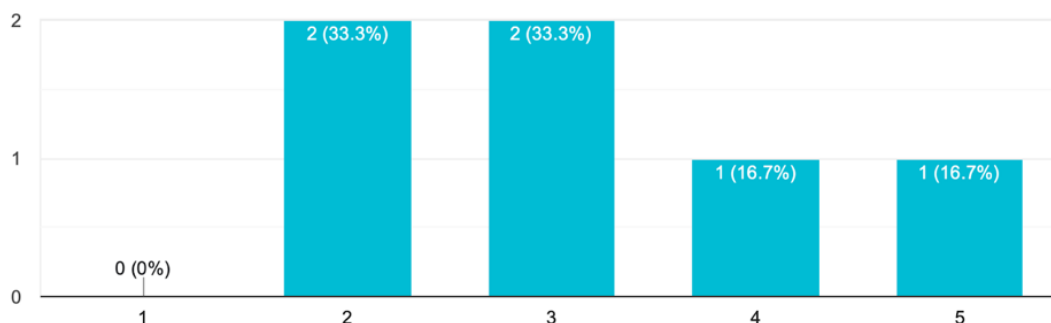
Järgmise küsimusega uuriti vastanutelt, kuidas nad eelnevalt välja toodud probleeme seni on lahendanud ning kui tugevalt need segavad töö tegemist. Üks inimene vastas, et tema tööle keskendumist segavad need üsna tihti, ülejäänud selle üle ei kurtnud. Pooled vastanutest ütlesid *ticketite* probleemide lahenduseks kolleegidega suhtlemise, kuid see varieerus nende vahel, et kas suheldakse *ticketi* tegijaga või erinevate osapooltega, et detailides ja eesmärgis kindel olla. Üks arendaja tõi välja, et probleemi ei lahendatagi ja tehakse tööd ilma infota. Samuti tõi ta prioriteetide küsimuses välja juhi poole pöördumise, kuid et ka temal pole tihti selget visiooni. Keskendumist segavate faktorite kohta öeldi nii palju, et kõige rangem lahendus oleks lülitada Slack välja, kuid on harjutud tegema mitut asja korraga, mistõttu on ka tekkinud keskendumisraskused. Sellisel juhul on parim aeg töötamiseks alates kella viiest õhtul, kui aktuaalsed teemad on lahendatud ja saab rahulikult enda tööga tegeleda.

## **4.2 Kanban boardi kasutus**

Kõik vastanud pidasid *boardi* olemasolu mingil määral oluliseks nende tööülesannete täitmisel. Viiepallisüsteemis pandi keskmiseks hindeks 3,167. Nagu Jooniselt 5 näha, siis vastati enim hindeks kaks ja kolm. Sellisest tulemusest saab järeldada, et *boardi* olemasolu ei mängi eriti olulist rolli arendustiimi igapäevatöös. *Boardi* kõige kõrgemalt hinnanud isik oli testija, kelle kogu töö tuleb *boardil* olevast “*In testing*” tulbast.

### Hinda kui suurt rolli mängib sinu tööülesannete täitmisel kanban board?

6 responses



Joonis 11. Boardi hinnang

Tööülesannete osas, mida täidetakse *boardi* väliselt saab teha olulise järelduse: kõik vastanud täidavad selliseid ülesandeid vähemal või suuremal määral. Kõige vähem ehk paar korda kuus täidab ülesandeid *boardi* väliselt testija ning ülejäänud 5 inimest vähemalt korra nädalas. Ükski vastus antud küsimuses ei kordunud ning nagu Jooniselt 6 näha, siis täidavad arendajaid ülesandeid, mis ei tule *boardilt* alates korra nädalas kuni mitu korda päevas.

### Kui tihti täidad tööülesandeid, mis ei ole tulnud SD/CS/PD boardilt?

6 responses



Joonis 12. Boardi väliste ülesannete sagedus

Kõik vastanud andsid järgnevas küsimuses teada, et peamiselt tekivad eelnevalt mainitud ülesanded Slackist. Seal toimuvad enamus ettevõttesisestest vestlustest, küsitakse küsimusi, arutatakse teemade üle ning jagatakse infot, seega on uued ülesanded arendustiimile kerged tulema. Lisaks tõid neli inimest välja oma initsiatiivi, kui nähakse,

et koodis on puudujääke või vigu, mida nad saaksid ise lahendada. Kaks inimest mainisid ka ära suusõnalised vestlused.

Viimase *boardi*-alase küsimusega uuriti ettepanekute kohta, et muuta *boardi* kasutamine veelgi efektiivsemaks. Siin vastuste puhul ühtsust ei tekkinud ning üks inimene ei osanud ühtegi soovitusi anda, kuna proovime ja katsetame pidevalt erinevaid lähenemisi. Esimese ettepanekuna öeldi, et tuleks lihtsalt rohkem *ticketeid* teha selle asemel, et tegevusi enda *to-do* listi panna. Üks vastanutest tõi välja, et võiksime tegevusi rohkem *epicutena* kirja panna. *Epic* on üldisem eesmärk või suurem uuendus, mida tahetakse saavutada ning see jaotatakse väikemateks *storydeks* ehk konkreetseteks alamtegevusteks. Front-end arendaja jaoks oleks oluline, kui "*Ready for development*" veerg oleks *boardil* alati uuendatud, et oleks ülesandeid, mida sealt võtta ja töö otsa ei saaks. Samuti et *ticketid* oleksid konkreetsemad, et vältida kahetimõistmist. Järgmise ettepanekuna toodi välja, et võiks arendusboardis prioriteete kasutada, mis oleks kaasajastatud ka teiste ettevõttesiseste *boardidega*. Viimane soovitus ei olnud otseselt *boardiga* seotud, kuid kirjutati sprinti metoodikast. Sprindile eelnevalt võiksid olla *ticketid* tehtud, millele järgneks *sprint planning*, kus kogu tiim vaataks need üle ja jõuaks ühisele arusaamale ning arenduse lõppedes paneb arendaja ja testija saavutatud tulemused kirja.

Kokkuvõtlikult on *boardi* parendamiseks ettepanekud:

1. Teha *ticketeid*, mitte lisada asju enda *to-do* listi
2. Rohkem *epicuid*
3. Prioriteetide kasutamine
4. Alati kaasajastatud "*Ready for development*" nimekiri
5. Parem ülevaade tiimiliikmete tegevustest ja tegevuskavast

### **4.3 Organisatoorne tagasiside**

Kõigi vastanute keskmine ajakulu koosolekutele on nädalas kaks kuni kolm tundi. Miinimum väljatoodud aeg on kaks tundi ning maksimum neli.



Kui uuriti vastanute rahulolu kohta seoses koosolekute ajakuluga, oli 4 inimest kuuest sellega rahul. Toodi välja, et selline korraldus ei sega tööd, aga saab siiski tiimi tegevuskavast aimdust. Kaks inimest väljendasin oma soovi koosolekute ajakulu suurendamise vastu, kuna on suurem vajadus tekkinud.

Seoses ootustega koosolekute osas vastasid pooled inimesed, et eelkõige on oluline saada selgust teemade osas, mis on hetkel töös ning mis on järgmisena tulemas. Soov on olla kursis kogu arendustiimi tegemistega ning läbi rääkida põhjused uute ülesannete taga. Väljendati ka soovi suurema ülesannete planeerimise vastu. Full-stack arendaja jaoks on oluline igasugune info, kuna ta viibib kõige keskel ja seetõttu oskaks ka äritiimi poolt esitatud küsimustele vastata. Lisaks toodi välja, kuidas tihti on meeskonnal mitu ülesannet töös, aga viimased pingutused jäävad tegemata ja seetõttu *ticketid* lahendamata.

Kuuest inimesest neljal on isiklikke tööalaseid eesmärke, millega suudavad panustada ettevõtte eesmärkidesse. Nendest kolm tõid konkreetselt välja isiklikud plaanid, mida tahetakse saavutada, kuid läbi nende aidata kaasa ettevõtte visiooni täitmisele ja klientide rahulolule. Lisaks väärtustatakse seda, et ettevõttes võetakse arvesse arendajate arvamusi ja soove ning kogu tegevus ei käi ainult juhtide järgi.

Enamus vastanutest tunnevad, et neile on ettevõttes üsna selged teiste osapoolte vajadused ja nõudmised. Põhjendusena toodi välja firma suurus, enda huvi ja initsiatiiv küsimusi esitada ebaselguste korral ning äritiimi kommunikeerimisoskus. Lisaks tõi üks nendest vastanutest välja, et tema arvates on tunda tasapisi aina vähem selgust. Ettevõttel puudub roadmap ning pikema vaatega ülevaatlikud koosolekud, seega on keeruline mõista, miks mingeid tegevusi tehakse ja kuidas see kedagi aitab. Kahele vastanule ei ole selged teiste osapoolte vajadused. Kõige selgemad on klientide nõudmised, kuid ettevõttepoolne info on puudulik. Tehakse küll ärialaseid koosolekuid, kuid sellegipoolest on edasine tegevuskava ebaselge.

Projektijuhi tööga oldi üldiselt rahul, eelkõige arvestades minimaalset kogemust antud vallas ning ettevõttes. Samuti lisati, et *taskid* on valdavalt hästi üles ehitatud ja juurde annaksid lisateadmised süsteemide omavahelisest tööst. Ühe vastanu arvates, oleks hinnangu andmiseks oleks vaja täpsustada, millised tegevused kuuluvad projektijuhi kohustuste hulka ja millised IT-juhi alla. Pooled vastanutest siiski jagasid enda nägemust

projektijuhi tööst. Üheks ootuseks oli projektijuhipoolne arendustiimi surveamine tähtaegadest kinni pidamiseks ning ülevaade tehtavatest ülesannetest ja nende valmimisest. Projektijuht võiks ka jagada arendustiimiga suuremat pilti ehk arusaama, miks midagi tehakse ja kuidas see toodet paremaks teeb. Samuti tuleks hoolitseda selle eest, et *boardi "Ready for development"* veerg oleks *ticketeid* täis erinevatele arendajatele, et kõigil tööd jaguks.

IT-juhi hinnangu osas jagunesid vastanute arvamused kaheks. Ühest küljest on suhtumine positiivne, kuna ta on antud valdkonnas rohkete teadmistega ja omab visiooni tootest, tänu millele on toimunud nii arenduses kui süsteemides erinevad muutused. Lisaks ei toimu mikromanageerimist, juht laseb töötajatel enda tööga tegeleda ning oma rahuliku olemuse tõttu ei tekita tiimile lisastressi pingelistel aegadel. Samas leitakse, et IT-juhi kommunikatsioon meeskonnaga on liiga vähene ja segane ning sellest tulenevalt puuduvad tiimil teadmised ümbritsevast ja tegevuskavast. Negatiivsest poolest toodi lisaks välja ka töökorralduse ajaplaneerimine ning üldine juhtimine. Arenduse poole pealt toodi välja fakt, et suurte arenduste puhul jääb kogu vastutus konkreetse arendaja peale, kuid IT-juhil võiks olla roll antud projekt finišisse aidata. Halba ei teeks ka tööülesannete jaotamine või sujuva arendusprotsessi kindlustamine. Samuti toodi välja, kuidas arendajate protsessid on varasemalt konflikti läinud, kuna puudub ühtne edasiviiv struktuur, selle asemel iga inimene toimetab oma ülesande kallal ja lisab selle suurele tootele juurde.

Viimasena toodi välja ettepanekud arendustiimi töökorralduse parendamiseks:

1. Paarissessionid, kus juunior arendajad saavad seenioritelt õppida ja seeläbi areneda, aitaks kiirendada *ticketite* valmimist tulevikus
2. Rohkem struktuuri protsessis ja ülevaadet tegevustest, roadmap ülesannetest
3. Arendajate arvukuse tõstmine
4. Hoida koosolekute ja muude ajaröövliite arvukus madalana
5. Suurendada vastutust koodi üles laadimisega, et seda ei tehtaks mõtlematult ja ilma testimata, oluline kvaliteet mitte kvantiteet

6. Suurendada automaattestide ulatust, kuna mida rohkem koodi muudetakse, seda suurem on tõenäosus, et midagi läheb katki

Kõik viimases punktis väljatoodud ideed arendusprotsessi parandamiseks ei ole kitsaskohad, mis lahendust vajavad, vaid ka ettepanekud, mida tulevikus kaaluda. Küsimustikus antud tagasiside vastanute poolt oli osaliselt juba projektijuhile teada, kuid saadi ka uut perspektiivi, et arendustiimi igapäevatööle kaasa aidata.

## 5 Protsessi parendusettepanekud

Järgnevas peatükis tuuakse välja arendusmeeskonna peamised kitsaskohad ning kirjeldatakse lahendusi nende parendamiseks. Kuna osad pakutud parendused pole otseselt seotud protsessi muudatusega, vaid konkreetse etapiga, siis esitatakse TO-BE joonised ainult protsessidele, kus toimub ka üldisel tasemel muudatus.

### 5.1 Arendusprotsessi pakutud TO-BE lahendused

Läbi tarkvaraarenduse AS-IS protsessi kaardistamise, läbiviidud küsitluse tulemuste, projektijuhi kohustuste analüüsi ja omapoolse projektijuhi hinnangu saab esitada kitsaskohad ning nende parendused vastavalt:

Tabel 2. Kitsaskoht 1

Arendusprotsess	<b>Kitsaskoht 1</b>	Info puudulikkus <i>ticketis</i>
	<b>Lahendus</b>	IT-juhi kaasamine äritiimi nõuete dokumendi detailsemasse täitmisesse, mille abil saaks projektijuht teha täpsemaid <i>ticketeid</i> .
	<b>Protsess (Etapp)</b>	Arenduse planeerimine ( <i>Ticketi</i> koostamine)
	<b>Staatus</b>	Pakutud

Tabel 3. Kitsaskoht 2

Arendusprotsess	<b>Kitsaskoht 2</b>	Koodi üles laadimine ilma testimata ning automaattestide vähesus
	<b>Lahendus</b>	Lõputöö kirjutamise käigus lahkus ettevõttest testija, seega kitsaskoha pakutud lahenduseks oleks uue testija palkamine.
	<b>Protsess (Etapp)</b>	Tarkvaraarendus ja testimine (" <i>In testing</i> " faas)
	<b>Staatus</b>	Pakutud

Tabel 4. Kitsaskoht 3

<b>Arendusprotsess</b>	<b>Kitsaskoht 3</b>	Ülesannete prioritseerimatus
	<b>Lahendus</b>	Antud probleemi lahenduses tegeletakse ainult <i>SD boardiga</i> , kuna peamiseks probleemiks on sealsete tegevuste prioritseerimine. Töö kirjutamise ajal on projektijuhil pooleli <i>SD boardi</i> “ <i>Ready for development</i> ” listi korrastamine, kus vaadatakse kõik <i>ticketid</i> üle ning eemaldatakse aegunud. “ <i>Ready for dev</i> ” list oleks aluseks ülesannete prioritseerimisel, kus kõige tipus paiknevad kõige olulisemad <i>ticketid</i> .
	<b>Protsess (Etapp)</b>	Nõuete kogumine ja analüüs (Nõuete prioritseerimine). Arenduse planeerimine (Tähtaegade selgitamine; Tegevuskava paika panemine; <i>Ticketi</i> koostamine; <i>Ticketi</i> määramine ja lisamine <i>boardile</i> )
	<b>Staatus</b>	Teostamisel

Tabel 5. Kitsaskoht 4

<b>Kanban board</b>	<b>Kitsaskoht 4</b>	<i>Board</i> ei vasta tegelikkusele, To-Do listide tekitamine <i>ticketite</i> asemel
	<b>Lahendus</b>	Tõsta <i>ticketite</i> arvukust projektijuhi ja arendajate endi poolt*, et kõik tegevused oleksid <i>boardil</i> kajastatud ja ei tekiks personaalseid To-Do liste. <i>Stand-uppidel</i> tuletada arendajatele meelde, et nende tegevused <i>boardil</i> nähtaval oleksid ja paikneksid õiges veerus.
	<b>Protsess (Etapp)</b>	Arenduse planeerimine ( <i>Ticketi</i> koostamine; <i>Ticketi</i> määramine ja lisamine <i>boardile</i> ). Tarkvaraarendus ja testimine
	<b>Staatus</b>	Planeeritud
	<b>Selgitus*</b>	Arendajal endal on võimalik lõppeesmärgiga <i>ticket</i> tükeldada alamtegevusteks ehk <i>sub-taskideks</i> . Põhiticketite prioriteetne järjestus ei muutu selle tegevuse käigus.

Tabel 6. Kitsaskoht 5

<b>Kanban board</b>	<b>Kitsaskoht 5</b>	Aegunud “ <i>Ready for dev</i> ” list, kõikidele arendajatele pole määratud ülesandeid
	<b>Lahendus</b>	Projektijuht käib korra nädalas <i>boardi</i> üle, et “ <i>Ready for development</i> ” veerg oleks kaasajastatud ja <i>ticketid</i> poleks kuhjunud pikema perioodi jooksul. Lisaks vaadatakse igapäevaselt üle, et veeru ülemised <i>ticketid</i> oleksid erinevatele arendajale ära määratud ja kõigil jaguks tegevust. Sujuva arendusprotsessi tagamiseks nõuab see veeru pidevat ülevaatamist ja tegevuste planeerimist tulevikuks.

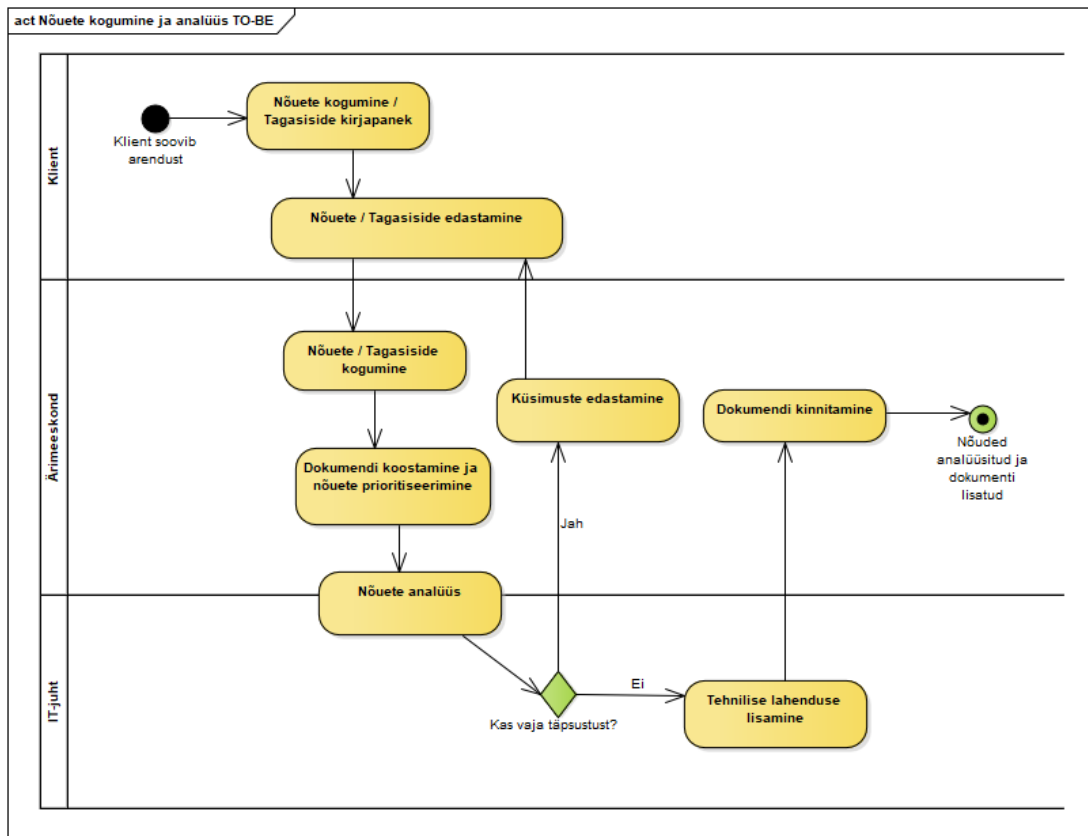
<b>Kanban board</b>	<b>Protsess (Etapp)</b>	Arenduse planeerimine (Tegevuskava paika panemine; <i>Ticketi</i> koostamine; <i>Ticketi</i> määramine ja lisamine <i>boardile</i> ). Tarkvaraarendus ja testimine
	<b>Staatus</b>	Planeeritud

Tabel 7. Kitsaskoht 6

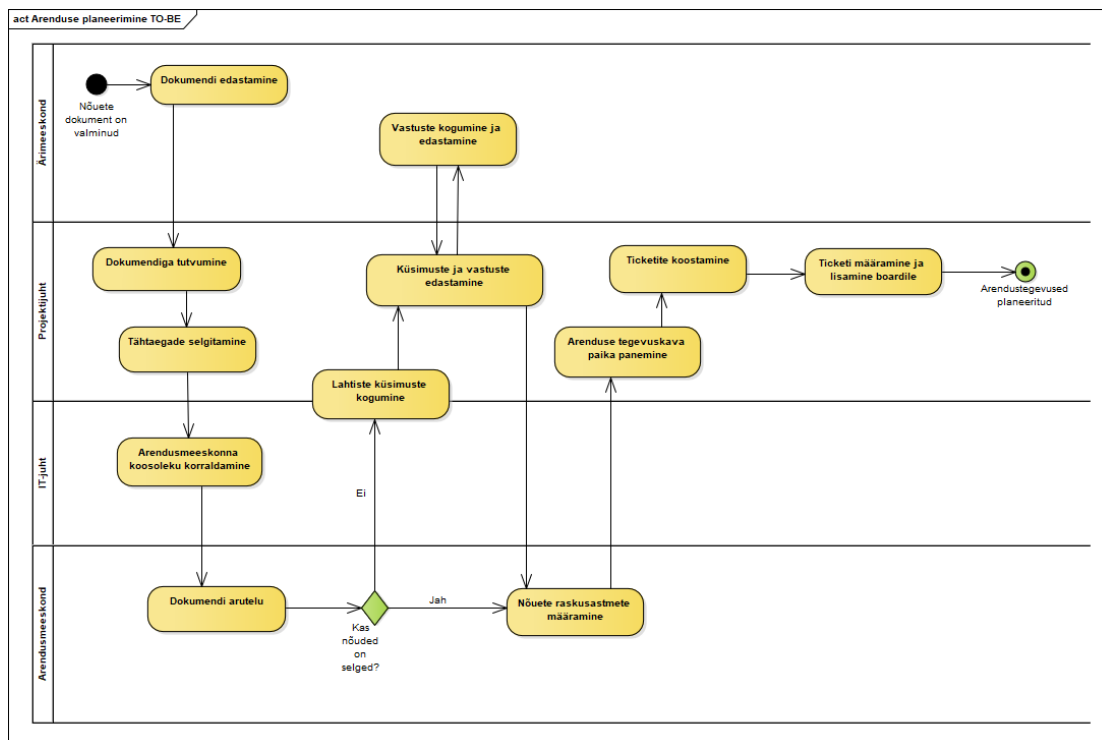
<b>Organisatoorne</b>	<b>Kitsaskoht 6</b>	Tegevuskava ja pikema vaate puudumine
	<b>Lahendus</b>	Iganädalased <i>weekly</i> koosolekud, mis kestavad maksimaalselt 45 minutit, et anda meeskonnale parem ülevaade tegevustest. Koosoleku eesmärk on arutada, mis sai eelmisel nädalal tehtud ja milline on uue nädala plaan.
	<b>Protsess (Etapp)</b>	Arenduse planeerimine (Tähtaegade selgitamine; Tegevuskava paika panemine). Tarkvaraarendus ja testimine
	<b>Staatus</b>	Teostatud

Tabel 8. Kitsaskoht 7

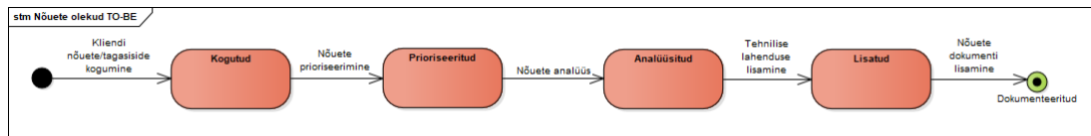
<b>Organisatoorne</b>	<b>Kitsaskoht 7</b>	<i>Weekly</i> koosolekute ülesehitus parema ülevaate tagamiseks
	<b>Lahendus</b>	Projektijuht paneb koos IT-juhiga algse tegevuskava kirja lähitulevikuks ning koos arendustiimiga vaadatakse see <i>weekly</i> koosolekul üle. Iga kahe nädala tagant käiakse dokument üle ja uuendatakse, vaadatakse mis on tehtud ning lisatakse uued eesmärgid, et <i>weekly</i> arendajatega jagada. Kuna arendustiimi <i>weekly</i> koosolekud tekkisid alles antud töö kirjutamise ajal, siis on see veel kõigile võrdlemisi uus ning läheb aega enne kui kogu protsess sujuma hakkab.
	<b>Protsess (Etapp)</b>	Arenduse planeerimine (Tähtaegade selgitamine; Tegevuskava paika panemine). Tarkvaraarendus ja testimine
	<b>Staatus</b>	Pakutud



Joonis 13. TO-BE: Nõuete kogumine ja analüüs



Joonis 14. TO-BE: Arenduse planeerimine



Joonis 15. TO-BE: Nõuete olekud

Kõik küsimustikus mainitud kitsaskohad pole antud peatükis koos lahendustega välja toodud, kuna mõned neist esinevad äärmiselt harva või on isikupõhised ning teised on tingitud ettevõtte olemusest. Samuti oleksid osad probleemid lahendatavad uute töötajate palkamisega.

Enne projektijuhi analüüsimist, saab ka IT-juhi kohta mõned järeldused välja tuua. IT-juhi tegevused vastavad hetkel rohkem juhtiv arendaja ehk *lead developeri* kohustustele, seega on olnud keeruline eristada projektijuhi kohustusi IT-juhi omadest, seda nii projektijuhil endal kui teistel töötajatel. IT-juhi soov oli vähendada oma töökoormust projektijuhi palkamisega, kuid ta ei andnud konkreetseid ülesandeid projektijuhile üle, seega on projektijuhi tegevused kujunenud jooksvalt vastavalt ettevõtte vajadustele. Lisaks on nende omavaheline tööalane kommunikatsioon olnud üsna minimaalne ning see on seganud projektijuhi sisse elamist ja hiljem ka töö tegemist. IT-juht on avaldanud arvamust, et sooviks tegeleda ainult arendusega ehk on tulevikus võimalik, et jääb ainult üks juhtiv roll.



## **6 Analüüs ja järeldused**

Käesolevas peatükis analüüsitakse projektijuhi tööd ja võrreldakse seda teooriast tulenevate ülesannetega. Tehakse ka järeldused eelmises peatükis väljatoodud lahenduste ja nende implementeerimise osas ettevõttes.

### **6.1 Projektijuhi tegevuste analüüs**

Projektijuhi tööülesannete nimekirja leiab rohkete raamatute ja artiklite seast, mis on kirjutatud mitmete autorite poolt, kuid tihti päris elu ja ettevõtted erinevad teooriast. Antud peatükis kirjeldatakse projektijuhi peamisi kohustusi teooriast lähtuvalt ning võrreldakse ettevõtte A projektijuhi tegevustega.

Projektijuhi peamisi kohustusi võib kategoriseerida nii [16]:

1. Projekti alustamine (planeerimine, eesmärgid, ajakava, eelarve)
2. Projektiplaanide esitamine
3. Meeskonna töö koordineerimine ja liikmete toetamine
4. Projekti kulgemise jälgimine (ja kvaliteedi tagamine)
5. Riskide ja konfliktide haldamine
6. (Töökeskkonna ja) kommunikatsiooni tagamine meeskondade ja liikmete vahel
7. Projekti lõpetamine

#### **6.1.1 Projekti alustamine**

##### **6.1.1.1 Projekti planeerimine, eesmärkide seadmine**

Üheks projektijuhi ülesandeks on projekti planeerimine ja eesmärkide seadmine. Ettevõttes A ei ole konkreetseid eristatavaid projekte, vaid toimub pidev agiilne platvormi arendamine. Erinevate uute arenduste planeerimine toimub vastavalt äritiimilt saadud

infole ning sellega tegelevad projektijuht ja IT-juht ühiselt. Siiski on IT-juhil peamine vastutus ja sõnaõigus tulevaste tegevuste osas.

**Ettepanek tulevikuks:** Tõsta arendustiimi juhtide ja äritiimi vahel kommunikatsiooni, et arendustiim saaks parema sisendi prioritseeritud tegevustest, mis vajavad tegemist. IT-juhi poolt vaja tugevamat planeerimist arendusmeeskonna jaoks.

#### **6.1.1.2 Projekti ajakava ja eelarve pidamine**

Järgmiseks alamkohustuseks on projekti ajakava ja eelarve pidamine. Ettevõttes A eelarve pidamisega projektipõhisel kujul ei tegeleta, küll aga kui klient soovib uut arendust, mis on suure töömahuga ja on eelkõige ainult neile vajalik, siis küsitakse selle eest lisaraha. Klientide tasumiste ja eelarvetega tegelevad enamjaolt siiski ettevõtte teised töötajad. Projekti ajakava osas otsustab tavaliselt arendustiim tervikuna, mitte projektijuht ainuisikuliselt. Tavaliselt ei tule tähtajad koos klientide nõudmistega kaasa, kuid antakse mõista nende olulisus või kriitilisus. Vastavalt nendele otsustavad arendajad ise, kui kaua ülesande tegemine aega võtab ning projektijuht jälgib, et arendusprotsess liiga pikale ei veniks.

**Ettepanek tulevikuks:** Projektijuht suurendab suhtlust äritiimiga, et mõista paremini klientide ootusi nõuete tähtaegade osas ning vähendada arendajate kaasamist tegevuste ajakava hindamisel. Samuti katsetada enda paremini kurssi viimist tellija ettevõtte tasumispoliitikaga, et tulevikus peale nõuete saamist anda kohe tagasiside, kas see on väärt arendusse võtmist, kui sealt vastavat kasumit tagasi ei tule.

#### **6.1.2 Projektiplaanide esitamine**

Projektijuhi teist ülesannet saab nimetada projektiplaanide esitamiseks. Kuna antud ettevõttes eraldiseisvad projektid puuduvad, siis puudub ka vajalikkus konkreetsete projektiplaanide järele. Siiski on vaja kommunikatsiooni hoida nii arendustimisiselt kui ka edastada nende tegemisi äritiimile. Kui projektijuht ja IT-juht äritiimiga tulevased tegevused läbi arutavad, on tegevuskava vaja ka arendustiimile edastada.

**Ettepanek tulevikuks:** Parem tegevuskava paika panemine ning selle edastamine arendusele.

### 6.1.3 Meeskonna töö koordineerimine ja liikmete toetamine

Järgmine projektijuhi kohustus on koordineerida meeskonnaliikmete tööd ning neid motiveerida ja toetada. See tegevus on ettevõttes peamiselt IT-juhi kanda, kuna tema on arendajate otsene ülemus, kuid teda aitab kõrvalt ka projektijuht. Planeeritakse koosolekuid ja tegevusi uuteks nädalateks, aidatakse arendajaid nende küsimuste korral, juhitakse tähelepanu probleemide korral jpm. Arendajate igapäevategevustega tegeleb siiski peamiselt projektijuht, kes määrab neile *ticketeid* ning otsib ülesannete puuduse korral uued ülesanded.

**Ettepanek tulevikuks:** IT-juhi konkreetsem juhtimine ning tegevuste jagamine arendajatele.

### 6.1.4 Projekti kulgemise jälgimine

Neljandaks projektijuhi ülesandeks on jälgida projekti kulgu. Ettevõttes A hoiab projektijuht ennast igapäevaselt kursis arendustiimi tegevustega valdavalt tänu *stand-uppidele*. Proovitakse ka jälgida, et tegevused päriselt tehtud saaksid, et takistuste ilmnedes ülesandeid kõrvale ei jäetaks. Projektijuhil on oluline teada, kui kaugel erinevad ülesanded on, et seda infot vajadusel äritiimiga jagada.

**Ettepanek tulevikuks:** Hoida paremat ülevaadet alustatud tegevustest ning projekti edukaks kulgemiseks parendada arendajate distsipliini läbi sagedasema suhtluse.

### 6.1.5 Riskide ja konfliktide haldamine

Viienda projektijuhi tegevusena võib välja tuua riskide hindamise, nõuetekohaste meetmete valimise ja konfliktide lahendamise. Sellega tegeleb peamiselt ettevõtte juht, kes teab nii äri kui arenduse poolt ning on väga kaasatud organisatsiooni igapäevategemistesse. Samuti on ka äritiimi juht, kes toob välja ettepanekuid riskide maandamiseks.

**Ettepanek tulevikuks:** Luua riskide haldamise dokument tulevaste projektide jaoks ning täiendada iga *release*'i ajal.

### **6.1.6 Kommunikatsiooni tagamine meeskondade ja liikmete vahel**

Eelviimaseks projektijuhi tegevuseks on kommunikatsiooni tagamine meeskondade ja liikmete vahel. Erinevate meeskondade vahel koosolekuid ettevõttes A valdavalt ei ole, äri- ja arendustiimil on eraldi tiimisisised koosolekud ning korra nädalas on ühine kohtumine arendusmeeskonna juhtide, äritiimi ning ettevõtte juhtide vahel. Kui on arendustiimi koosolek, siis seda juhib IT-juht ning tema puudumisel projektijuht. Kui projektijuhil on tekkinud teemasid, mis vajavad ühist arutamist, siis võtab tema juhtimise üle.

**Ettepanek tulevikuks:** Projektijuhi ja IT-juhi parem ettevalmistus enne *weekly* koosolekut ning sagedasem arendustiimi tegevuste edastamine äritiimile.

### **6.1.7 Projekti lõpetamine**

Viimaseks projektijuhi tegevuseks on projekti lõpetamine. Ettevõttes A viiakse korra kuus arendustiimis läbi retrospektiiv, mille eesmärk on läbi arutada, mis on viimasel ajal läinud meeskonnas või protsessis hästi ja halvasti. Retrospektiiv toimub koosolekuvormis, kus kasutatakse veebilehte, mille kaudu saavad meeskonnaliikmed jätta anonüümset tagasisidet. Tagasisidest tulenevalt arutatakse teemad läbi ning mõeldakse tegevused probleemide lahendamiseks, mis seejärel tiimiliikmete vahel ära jagatakse. Lisaks arendustiimisisesele “projekti lõpetamisele” on vaja nii kliente kui ülejäänud töötajaid teavitada ettevõtte *release*-ist. Klientidele koostatakse *release notes*, kuhu pannakse kirja olulisimad uuendused platvormil ning nende eest vastutab projektijuht. Ettevõtte ülejäänud töötajate teavitamisega on tegelenud nii IT-juht kui projektijuht ja see käib läbi Slacki vestluskanalite.

**Ettepanek tulevikuks:** Töötajate teavitamine muuta ainult projektijuhi kohustuseks ning seda teha kohe *release*-i järgselt, et eelkõige äritiim oleks kursis.

## **6.2 Tulemuste järelused**

Lõputöö tulemusena saadi teada arendustiimi peamistest murekohtadest ning töö autor liikus selles suunas, et leida neile lahendus. Selleks, et välja pakutud lahendusi valideerida, peavad paigas olema kindlad mõõdikud, et hinnata nende efektiivsust. Mitmed kitsaskohtade parendamised on juba teostamisel, kuid nende tulemusi on liiga

vara hinnata. Selliste muudatuste resultaadid ilmnevad tavaliselt paari kuu järel, kuid esmane tagasiside arendusmeeskonnalt on olnud väga positiivne. Antud muudatused lihtsustavad nende igapäevatööd ja annavad rohkem selgust juurde miks ja mida teha.

Kitsaskohtade lahenduste hindamiseks toome välja planeeritud mõõdikud:

1. Klientide rahulolu uuring (tihedamad *release*'id, platvormil olevate vigade arv pärast *release*'i, sagedasemad uuendused jne)
2. Katkiste *ticketite* arvukus peale *release*'i
3. Lahendamata *ticketite* arv *release*'is (kas saadi valmis või jäeti katkisenä välja)
4. *Release*'i toimumine määratletud ajal (kordade arv - enneaegselt, õigeaegselt, hilinenud)
5. Arendustiimisisene küsitlus ja tagasiside

Aasta viimases kvartalis otsustas ettevõtte mitte uusi muudatusi platvormil läbi viia. See otsus tuli klientide soovist, kuna aasta lõpp on neile kõige olulisem aeg ning nad ei saa riskida, et midagi reklaamikampaaniate või platvormiga juhtub. Kui varasemalt toimusid *release*'id keskmiselt iga kahe nädala tagant, siis nüüd ei ole novembri ega detsembri jooksul ühtegi uuendust läbi viidud. Sellegipoolest on arendustiim jätkanud hoolsalt tööd ning kanban *boardi* “*In testing*” ja “*To release*” tulbad on *ticketeid* täis. Antud töö raames tulemuste valideerimise tegi selline ootamatu otsus keeruliseks. Autor sai implementeerida arendusprotsessis muudatusi, kuid planeeritud mõõdikute järgi ei saa neid hinnata ilma *release*'ideta, piirduti vaid suulise tagasisidega.

Peamiseks muutuseks AS-IS ja TO-BE vahel on vähenev suhtlus arendajate ja teiste osapoolte vahel, kuna kanban *boardile* ja *ticketitesse* tuleb rohkem selgust. Kanban *boardi* “*Selected for development*” veerg korrastati vanadest *ticketitest* ära ning uued pandi prioriteetsesse järjekorda. Suhtluse vähenemisele erinevate osapooltega on aidanud kaasa ka projektijuhi olemasolu, keda varasemalt ettevõttes ei olnud. Arendajad olid harjunud suhtlema otse äritiimiga, kuid nüüd on märgata suhtluse vähenemist nende vahel ja küsimuste korral pööratakse otse projektijuhi poole.

Teine suurem muudatus AS-IS olukorras oli testija lahkumine ettevõttest. See tuli kõigile arendustiimiliikmetele kahjuks, kuna see tõstis nende töökoormust. Nüüd testivad arendajad enda uusi lahendusi ise või aitavad kaasa projektijuht ja IT-juht. Kuna pole ühte konkreetset testijat, siis see muudab protsessi ebasujuvaks ja aeglasemaks. TO-BE protsessi koostamisel arvestati siiski testija vajadust ja olemasolu.

Kolmandaks tekkis juurde iganädalased *weekly* koosolekud, et tõsta tiimisisest suhtlust ning jagada omavahel rohkem infot hetkese ja tuleviku töö kohta. Selle tagasiside arendustiimi poolt on olnud positiivne, kuid nende ettepanekute tulemusena proovivad juhid edaspidi anda paremat ülevaadet ja tegevuskava.

Neljandana tuleb vastavalt eelmistele punktidele kaasata IT-juhti rohkem juhtimisse. IT-juhi panus on oluline nõuete dokumenteerimisel ning *weekly* koosolekute jaoks tegevuskava paika panemisel, kuid tema hetkene soov on need tegevused lasta ainult projektijuhil teha.

Lisaks kasvab projektijuhi tööülesannete hulk ning IT-juhi tegevuste osas on vajalik arutelu, et kas need klapiivad tema ametinimetusega.

## 7 Kokkuvõte

Bakalaureusetöö käsitleb ettevõtte A tarkvaraarenduse protsessis tekkivaid probleeme ning nende lahendusi. Käesoleva lõputöö eesmärk oli analüüsida ettevõttes kasutusel olevat arenduse tööprotsessi ja leida kitsaskohad, et muuta ettevõtte töö efektiivsemaks nii töötajate kui klientide jaoks. Töös käsitletav probleem seisnes asjaolus, et ettevõttes puudus varasemalt IT-projektijuht ning arendusmeeskonna töös oli märgata segadust ja arusaamatusi.

Eesmärkide saavutamiseks kaardistati ettevõtte AS-IS kogu arendusprotsessi vältel. Samuti viidi arendusmeeskonnas läbi küsitlus, et saada töötajate hinnang ettevõtte igapäevatööle ning tuvastada levinuimad probleemid. Küsimused olid jaotatud kolme kategooriasse: üldised igapäevatöö probleemid, kanban *boardi* kasutamine ning organisatoorne analüüs. Enim väljatoodud probleemidena toodi välja tegevuskava puudumine, ülesannete prioritseerimatus, info puudulikkus *ticketites* ja aegunud *board*. Positiivsest küljest on koosolekutele kuluv aeg optimaalne, kuid *Weekly* kvaliteeti tuleks veel tõsta.

Töö analüüsi ja järelduste all toodi välja peamised probleemid koos pakutud lahendustega ning koostati TO-BE protsessimudelid. Töö käigus alustati arendusmeeskonnas iganädalaste *Weekly* koosolekute tegemisega ning mitmed teised probleemide pakutud lahendused võeti töösse. Esmane tagasiside meeskonnalt oli positiivne, kuna lahendused muudavad nende töö lihtsamaks ja efektiivsemaks, kuid ametliku valideerimise saab teha alles ettevõtte järgmises kvartalis. Viimasena koostati ka põhjalik projektijuhi ülesannete analüüs, kus võrreldi teooriat tegelikkusega ning toodi välja ettepanekud tulevikuks.

## Kasutatud kirjandus

- [1] „Jira,“ Atlassian, [Võrgumaterjal]. Available: <https://www.atlassian.com/software/jira/>. [Kasutatud 20 Detsember 2022].
- [2] E. Canziba, Hands-On UX Design for Developers, Birminham: Packt, 2018.
- [3] H. Vallaste, „e-Teatmik,“ [Võrgumaterjal]. Available: <http://www.vallaste.ee/>. [Kasutatud 22 September 2022].
- [4] Slack, „Slack,“ [Võrgumaterjal]. Available: <https://slack.com>. [Kasutatud 4 Detsember 2022].
- [5] D. Radigan, „The product backlog: your ultimate to-do list,“ Atlassian, [Võrgumaterjal]. Available: <https://www.atlassian.com/agile/scrum/backlogs>. [Kasutatud 5 Oktoober 2022].
- [6] M. Rehkopf, „What is a Kanban board?,“ Atlassian, [Võrgumaterjal]. Available: <https://www.atlassian.com/agile/kanban/boards>. [Kasutatud 4 Detsember 2022].
- [7] D. Radigan, „Kanban,“ Atlassian, [Võrgumaterjal]. Available: <https://www.atlassian.com/agile/kanban>. [Kasutatud 2 Detsember 2022].
- [8] L. Hennigan, „What Is A Kanban Board? The Ultimate Guide,“ Forbes Advisor, [Võrgumaterjal]. Available: <https://www.forbes.com/advisor/business/software/what-is-kanban-board>. [Kasutatud 8 Oktoober 2022].
- [9] R. Lynn, „What is a kanban board,“ Planview, [Võrgumaterjal]. Available: <https://www.planview.com/resources/guide/introduction-to-kanban/what-is-kanban-board>. [Kasutatud 20 Detsember 2022].
- [10] S.Siderova, „Improve Task Organization: Kanban Swimlanes,“ Nave, [Võrgumaterjal]. Available: <https://getnave.com/blog/kanban-swimlanes>. [Kasutatud 19 Detsember 2022].
- [11] „5 Phases of Project Management – A Complete Breakdown,“ Kissflow, [Võrgumaterjal]. Available: <https://kissflow.com/project/five-phases-of-project-management/>. [Kasutatud 14 Oktoober 2022].
- [12] „6 phases of the software engineering life cycle,“ Erbis, [Võrgumaterjal]. Available: <https://erbis.com/blog/6-phases-of-the-software-engineering-life-cycle>. [Kasutatud 15 Oktoober 2022].
- [13] H.Donato, „5 Phases of Project Management Life Cycle You Need to Know,“ Project-management, [Võrgumaterjal]. Available: <https://project-management.com/project-management-phases>. [Kasutatud 16 Detsember 2022].
- [14] T.Hamilton, „What is Software Testing? Definition,“ Guru99, [Võrgumaterjal]. Available: <https://www.guru99.com/software-testing-introduction-importance.html>. [Kasutatud 21 Oktoober 2022].
- [15] „What is Software Deployment?,“ GoodFirms, [Võrgumaterjal]. Available: <https://www.goodfirms.co/glossary/software-deployment>. [Kasutatud 27 September 2022].



- [16] P. M. Institute, A Guide to the Project Management Body of Knowledge (PMBOK Guide)  
– Fifth edition, Newtown Square, 2013.

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Greta Reino

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Tarkvaraarenduse protsessianalüüs ja -arendus ettevõtte A näitel“, mille juhendaja on Karl-Erik Karu
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

02.01.2023

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 - Küsitlus

1. Mis on sinu roll arendusmeeskonnas?
2. Kirjelda ühe taski valmimist läbi enda rolli. Näiteks millised on sinu tegevused ülesande täitmisel
3. Milliseid probleeme on sul tekkinud enda igapäevatöö protsessis? Kui tihti neid ette tuleb? (*valesti määratud ticket, kommunikatsiooniprobleemid, ticket puudu, vähene info ticketis, tööülesannete puudumine, ajapuudus, ülesannete prioritseerimatus, board ei vasta tegelikkusele*)
4. Kuidas oled seni lahendanud eelmainitud olukordi? Kui tihti takistavad need sul enda tööle keskendumist?
5. Hinda kui suurt rolli mängib sinu tööülesannete täitmisel kanban board?
6. Kui tihti täidad tööülesandeid, mis ei ole tulnud SD/CS/PD boardilt?
7. Millest need ülesanded tekkinud on? (*Slacki vestlused, suusõnalised vestlused, oma initsiatiivil tehniline muudatus*)
8. Kas on ettepanekuid, kuidas muuta board efektiivsemaks arendusprotsessile? (*Lisada/Eemaldada veerge, rohkem ticketeid arenduse siseselt, TODO-list, jne*)
9. Kui palju aega kulub sul koosolekutele ja stand-uppidele? (tundi nädalas)
10. Oled sa antud ajakuluga rahul, peaks seda aega vähendama või hoopis suurendama?
11. Millist infot sa koosolekutelt ootad ja kas tunned, et on kasumlik ja ajakulu väärt? Kas ootused vastavad reaalsusele?
12. Millised on sinu tööalased eesmärgid? Kas tunned, et saad ja tahad panustada ettevõtte eesmärkidesse?
13. Enda arvates, kui selge on sulle teiste osapoolte (äritiim, kliendid, ettevõtte üldiselt) vajadused ja nõudmised? Põhjenda.
14. Sinu hinnang projektijuhi tööle. (Mis on sinu ootused talle, mis on positiivset, mida saaks tulevikus paremini)
15. Sinu hinnang IT-juhi tööle. (Mis on sinu ootused talle, mis on positiivset, mida saaks tulevikus paremini)
16. Sinu ettepanekud, kuidas tarkvaraarenduse töökorraldust veelgi efektiivsemaks muuta.