

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Reelika Lõhmus 144332IAPM

TESTIHALDUSVAHENDI VALIK ASA QUALITY SERVICES OÜ NÄITEL

Magistritöö

Juhendaja: Maili Markvardt
Magistrikraad

Tallinn 2017

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: Reelika Lõhmus

08.05.2017

Annotatsioon

Käesolev magistritöö kirjeldab testihaldusvahendi valikut tarkvara testimisega tegelevale ettevõttele. Töö eesmärgiks on välja selgitada milliseid erinevaid testihaldusvahendeid on olemas ning nõuetele vastavuse põhjal leida ettevõttele uus ja sobivam tööriist. Esimesena antakse ülevaade erinevatest arendusmetoodikatest, mille kasutamisega ettevõttes kõige rohkem kokku puututakse. Seejärel pannakse paika nõuded, millele otsitav testihaldusvahend vastama peab. Eesmärgi saavutamiseks võrreldakse erinevaid testihaldusvahendeid ning viiakse läbi küsitlus testihaldusvahendite kasutamise kohta Eestis. Lõpuks lahendatakse nelja parima tööriistaga 9 erinevat ülesannet ning tulemuste põhjal valitakse välja sobivaim tööriist.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 46 leheküljel, 6 peatükki, 19 joonist, 7 tabelit.

Abstract

The selection of test management tool on the example of ASA Quality Services OÜ

The current Master's thesis describes the selection of test management tool for software testing company. ASA Quality Services OÜ uses in their work test management tool named TestLink. The problem is that the test management tool is outdated and needs replacing.

The purpose of the work is to explore different test management tools and to find the best test management tool based on the requirements. To achieve the purpose of the Master's thesis, various test management tools are compared and inquiry about usage of different test management tools in Estonia is conducted. In addition, there are different tasks that the tool must be able to solve.

In the second chapter different software development methodologies are described. Third chapter contains the requirements the tools must meet. The fourth chapter is composed of two parts. First part contains comparisons of 9 test management tools. Second part contains analysis of the inquiry results. Finally, 9 different tasks are carried out that the test management tool must be able to solve.

To conclude, the most important outcome of the thesis were:

- Test management tools comparison;
- Analysis of the test management tools use in Estonia;
- Created tasks to control the tools capability to solve them.

As a result the best test management tool is chosen.

The thesis is in Estonian and contains 46 pages of text, 6 chapters, 19 figures, 7 tables.

Lühendite ja mõistete sõnastik

ASA	Ettevõtte ASA Quality Services OÜ
TPA	<i>Test-driven development</i> , Testipõhine arendus
WIP	<i>Work in progress</i> , hetkel töös olev ülesanne
ISO 29119	Tarkvara testimise standard

Sisukord

1 Sissejuhatus	9
1.1 Probleemi taust ja kirjeldus	9
1.2 Ülesande püstitus	10
1.3 Ülevaade tööst	10
2 Testimine erinevates arendusmetoodikates	12
2.1 Scrum.....	12
2.2 Kanban.....	15
2.3 V-mudel.....	16
3 Nõuded testihaldusvahendile.....	19
4 Testihaldusvahendid ja nende kaardistamine	21
4.1 Testihaldusvahendite võrdlus	22
4.2 Testihaldusvahendite kasutamine Eesti ettevõtetes.....	23
5 Testihaldusvahendi väljavalimine	27
5.1 Testülesanded	27
5.2 qTest Manager	32
5.3 Zephyr for JIRA.....	34
5.4 TestLink.....	36
5.5 Xray for JIRA	39
5.6 Testihaldusvahendite kasutamise analüüs	40
5.7 Järelduste tegemine.....	41
6 Kokkuvõte	42
Kasutatud kirjandus	44
Lisa 1 – Testihaldusvahendi küsitluse küsimused.....	47

Jooniste loetelu

Joonis 1. Scrumi raamistik	14
Joonis 2. Scrumi tiim.....	14
Joonis 3. Lihtne Kanban tahvel.	15
Joonis 4. Põhjalikum Kanban tahvel.	16
Joonis 5. V-mudel.....	18
Joonis 6. Testimiskogemuse jaotus.	24
Joonis 7. Testijuhi kogemus.	24
Joonis 8. Testihaldusvahendi omaduste olemasolu olulisus.....	25
Joonis 9. Testihaldusvahendi funktsionaalsuste kasutamise tihedus.....	26
Joonis 10. Testide nimekiri.....	32
Joonis 11. Käivitamiste ajalugu.....	33
Joonis 12. Testitulemuste kujutamine graafiliselt.	34
Joonis 13. Viimased tegevused.....	34
Joonis 14. Testiloo lisamise sammud.	35
Joonis 15. Testide käivitamine erinevate testitsükli poolt.....	36
Joonis 16. Nõuete haldamine.....	37
Joonis 17. Testilugude lisamine/eemaldamine testiplaanis.	38
Joonis 18. Testitulemuste kuvamine gruppide kaupa.....	38
Joonis 19. Testide tulemuste kuvamine ühe testiplaani raames.	39

Tabelite loetelu

Tabel 1. Olulised nõuded.....	19
Tabel 2. Vähemolulised nõuded.....	20
Tabel 3. Testihaldusvahendite võrdlus.....	22
Tabel 4. Sisselogimine mobiil-ID'ga.....	28
Tabel 5. Juhiloa vahetamise protsess.....	28
Tabel 6. Kasutajate muutmise protsess.....	30
Tabel 7. Sisselogimine ID-kaart.....	31

1 Sissejuhatus

Tarkvara testimisel on tarkvara arendusprotsessis üha olulisem roll. Ettevõtted on hakanud mõistma, et kvaliteetse tarkvara loomiseks tuleb rohkem tähelepanu pöörata tarkvara testimisele. Kuna vigade parandamine on odavam projekti algusfaasis, siis on tähtis, et vead avastataks võimalikult varakult. Testimisprotsessi lihtsamaks ja sujuvamaks läbiviimiseks on loodud erinevad testihaldusvahendid. Testihaldusvahendi näol on tegemist tööriistaga, mis aitab testijatel ja testijuhtidel paremini testimisprotsessi läbi viia. Testihaldusvahendis on kirjas testide nõuded ja täpsemad testiloo kirjeldused. Lisaks on võimalik kirja pandud testilugusid antud tööriistaga käivitada ning seejärel testitulemused kirja panna. Paljud vahendid võimaldavad raporteerida ka leitud vigu. Tänu sellele, et testihaldusvahendis saab testilugude läbimise tulemusi märkida, saavad testijuhid ülevaate testimisprotsessi seisust. Käesoleva magistr töö eesmärgiks on leida ettevõtte nõuetele vastav testihaldusvahend, mis aitaks testimisprotsessi lihtsustada.

1.1 Probleemi taust ja kirjeldus

Täna päeval kasutab ettevõtte ASA Quality Services OÜ oma töös peamiselt testihaldusvahendit nimega TestLink. Selle testihaldusvahendi kasutamisel on selgunud mitmed põhjused, miks antud testihaldusvahend enam ettevõttele ei sobi. Esiteks on ettevõttes kasutusel olev TestLink'i versioon 1.9.2 ning kõige uuem olemasolev versioon on 1.9.16 ehk tegemist on väga vana versiooniga. Kogemus on näidanud, et antud programmi kasutamine ei ole kõige mugavam ning testilugude koostamiseks tuleb teha palju samme, mida oleks võimalik tunduvalt lihtsamini lahendada. Antud tööriist ei ole kõige sobivam agiilse arendusprotsessi läbiviimiseks, sest selleks, et testiplaani muuta tuleb luua sellest uus versioon ning jooksvalt seda täiendada ei saa. Keeruliseks muudab asjaolu ka see, et testilugude käivitamisel saab määrata tulemuse ainult kogu testiloo kohta üldiselt. See on halb seetõttu, et kui tegemist on pika testilooga, siis on hiljem raske aru saada kus täpsemalt viga tekkis. Vanema versiooni puuduseks on ka see, et puudub võimalus testilugude käivitamiseks ja vigade haldamiseks. Kuna aastate

jooksul on testihaldusvahendeid palju juurde tulnud ning olemasolevaid täiustatud, siis otsustas ettevõtte, et versiooni uuendamise asemel uuritakse kas leidub testihaldusvahendit, mis sobiks ettevõtte vajadustega paremini.

1.2 Ülesande püstitus

Magistritööd alustatakse erinevate arendusmetoodikate kirjeldamisega, et saada ülevaade peamiselt kasutusel olevate arendusmetoodikate kohta. Seejärel kirjeldatakse ettevõtte poolt paika pandud nõuded, mida leitav testihaldusvahend täitma peaks. Nõuete põhjal koostatakse võrdlev tabel erinevatest testihaldusvahenditest koos nende plusside ja miinustega. Kuna võrdlusesse minevate tööriistade valik tehakse välismaiste võrdlevate edetabelite põhjal, siis luuakse autori poolt küsimustik testihaldusvahendite kasutamise kogemuse kohta. Küsimustiku eesmärgiks on välja selgitada kas ja kui palju eelpool nimetatud tööriistu Eestis asuvates ettevõtetes kasutatakse. Parema ülevaate saamiseks viiakse nelja testihaldusvahendiga läbi ka lõputöö autori poolt loodud testülesanded.

Töö tulemusena selguvad erinevate testihaldusvahendite plussid ja miinused. Tänu praktilistele ülesannetele saab testihaldusvahenditest põhjalikuma ülevaate. Lisaks sellele selgub milline testihaldusvahend vastab kõige enam antud töös püstitatud hindamiskriteeriumitele. Ülesannete lahendamise järgselt analüüsib autor testülesannete tulemusi ning teeb selle põhjal järeldused erinevate testihaldusvahendite kasutamise osas.

1.3 Ülevaade tööst

Teises peatükis kirjeldatakse tarkvara testimise vajalikkust ning antakse ülevaade erinevatest arendusmetoodikatest, mida ettevõttes ASA Quality Services OÜ kõige rohkem kasutatakse.

Kolmandas peatükis kirjeldatakse nõudeid, mis otsitava testihaldusvahendi valimise puhul olulised on. Nõuded on tähtsuse järgi jagatud kahte gruppi. Esimesse gruppi kuuluvad nõuded, mille olemasolu testihaldusvahendi juures on väga tähtis. Teise gruppi kuuluvad nõuded, mille olemasolu oleks hea, kuid need ei ole nii tähtsad.

Neljandas peatükis kirjeldatakse testihaldusvahendi kasutamise vajalikkust. Erinevate testihaldusvahendite kohta on loodud võrdlev tabel, kus on kirjas kolmandas peatükis kirjeldatud nõuete olemasolu. Lisaks sellele analüüsitakse neljandas peatükis töö autori poolt loodud testihaldusvahendite kasutamise küsimustiku tulemusi.

Viiendas peatükis on kirja pandud töö autori poolt loodud ülesannete kirjeldused, mis on koostatud etteantud nõuete põhjal. Testihaldusvahendeid võrdleva tabeli ning küsimustiku analüüsi tulemuste põhjal on välja valitud neli erinevat tööriista, millega antud ülesandeid lahendada hakatakse. Ülesannete lahendamise järgselt analüüsitakse töö tulemusi ning tehakse järeldused milline testihaldusvahend täidab kõige paremini ettevõtte ASA Quality Services OÜ vajadused.

2 Testimine erinevates arendusmetoodikates

Kvaliteedi tagamine on oluline osa tarkvaraarenduse protsessist [4]. Kvaliteetne tarkvara ei sisalda vigu, on õigel ajal kliendile üle antud ning vastab nõuetele [9]. Selleks, et teada saada, kas loodud tarkvara vastab nõuetele, viivad kvaliteedispetsialistid läbi erinevaid teste ning kontrollivad kas süsteem vastab nõuetele. Nende ülesandeks on kindlaks teha, et süsteemis ei oleks vigu ja kui on, siis tuleb need tagasi arendustiimile parandamiseks saata [4]. Tarkvara testimine on oluline, sest kõik inimesed teevad vigu. Mõned leitud vigadest on vähetähtsad, kuid mõned neist väga olulised ja nende parandamata jätmine võib osutuda väga kalliks [18]. Testimine aitab vähendada suuremaid ja väiksemaid riske, mis vigade süsteemi jäämisega tekkida võivad. Kahjuks on paljude ettevõtete arvates tegemist vähetähtsa osaga, mille tõttu testimise kvaliteeti väga palju ei panustata ning see on ka üks suuremaid põhjuseid, miks on olemas nii palju vigast tarkvara. Testimist võib läbi viia väga erinevatel viisidel. On projekte, kus testimine toimub alles peale arenduse lõppemist ning sedagi vaid põgusalt, kuid teisest küljest on üha enam neid projekte, kus testimine on väga põhjalik ning ka osa arendusprotsessist. Alljärgnevalt vaatamegi kolme peamist arendusmetoodikat, millega antud lõputöö autor oma töö juures kõige enam kokku puutub.

2.1 Scrum

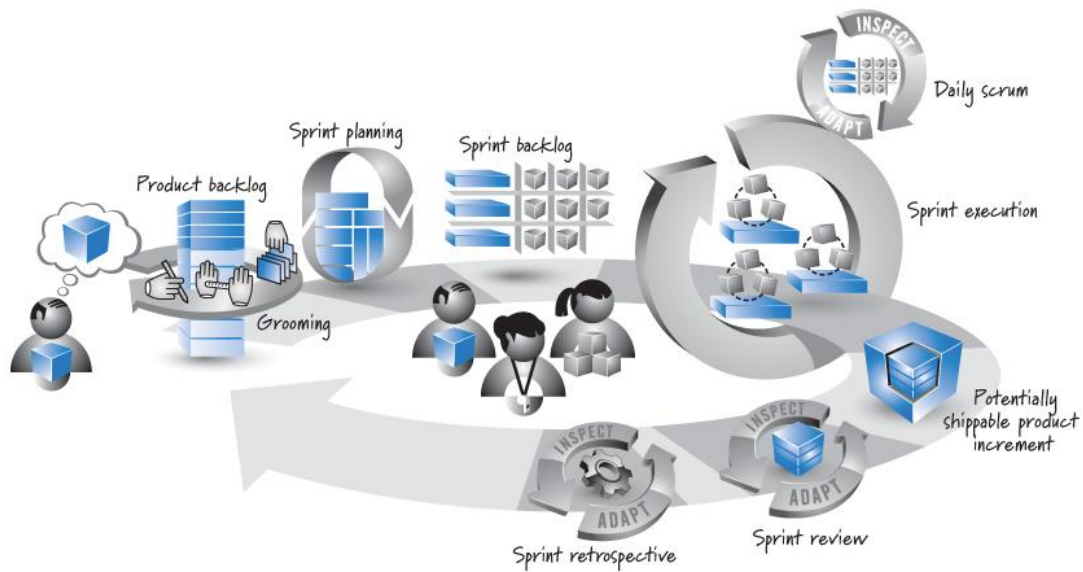
Scrum on agiilse arendusmetoodika üks harudest. Scrumi kõige tuntumaks omaduseks on sprintide olemasolu, mis tähendab seda, et tal on kindla pikkusega iteratsioonid, mis tavaliselt kestavad 1-4 nädalat. Sprindi täpne pikkus sõltub tiimi nõuetest ja võimekusest. Scrum tiimi suuruseks on 5-9 inimest. Scrumil on neli tähtsat komponenti, mis ühestki sprindist ei puudu [1, 19, 20]:

- Sprindi planeerimine
- Igapäevane koosolek

- Sprindi demo
- Sprindi tagasivaade

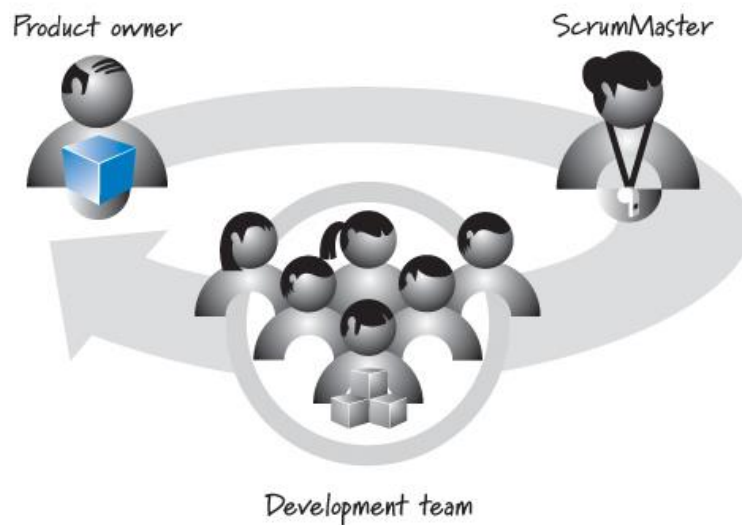
Testimise seisukohast on Scrumi puhul oluline see, et ta on tihedalt arendusprotsessiga seotud ning iga meeskonna liige peaks omama nii palju teadmisi, et ta suudab ise ülesande algusest lõpuni ära teha. Testimine saab alguse projekti varajases etapis, kus arendajate tööga samaaegselt hakkavad testijad nõudeid üle vaatama ning selle põhjal testilugusid looma. Testipõhise arenduse (TPA) korral on testide koostajaks arendajad ise, kes peale testide valmimist hakkavad nende põhjal programmi koodi kirjutama. Kuna testimine on osa arendusprotsessist, siis avastatakse ka vead palju varem ning seetõttu vähenevad ka paranduste tegemise kulud, sest mida varem viga avastatakse, seda odavam see on. Kui traditsiooniliselt oli testimine täiesti eraldiseisev osa ning testimist alustati peale arenduse lõpetamist, siis nüüd teeb testija tööd koos arendustiimiga ning ka vastutust jagatakse kogu protsessi eest üheskoos [1, 2, 9].

Joonisel 1 on kujutatud ülevaade Scrum raamistikust. Iga uus sprint algab sprindi planeerimisega ning lõppeb tagasivaatega sprindi tegevustele. Esmalt vaadatakse üle tegemata tööde nimekiri, mis peab olema järjestatud nii, et kõrge prioriteediga tööd on nimekirja alguses ning madala prioriteediga tööd nimekirja lõpus. Sprindi planeerimisel pannakse paika järgmise sprindi eesmärk, mille põhjal vaatab arendustiim üle tegemata tööde nimekirja ning paneb paika ülesanded, mida järgmise sprindi ajal arendama hakatakse. Kui Scrumi tiim on planeerimise lõpetanud, siis hakatakse sprinti valitud ülesandeid täitma. Sprindi ajal toimuvad igapäevased tiimi kokkusaamised, kus räägitakse töödest, mida tehti eelmisel päeval ja mis on plaanis järgmisel päeval ning lisaks sellele arutatakse erinevate probleemide üle, mis sprindi ajal tekkinud on. Sprint lõppeb sellega, et valmis on saanud potentsiaalselt tarnitav toode, mis ei taga seda, et valminud töö kindlasti kliendile üle antakse. Otsuse tarnimise kohta teeb ettevõtte äri pool. Kui toode on valmis, siis toimub tehtud töö ülevaatamine sprindi tiimi, sponsorite ja kliendiga koos. Vestluse käigus saavad arendustiimist väljaspool olevad inimesed ülevaate tehtud funktsionaalsusest. Peale sprindi lõppu toimub tagasivaade tehtud tööle, et arutada asjade üle, mis läksid hästi ja mida saaks järgmisel korral paremini teha [3].



Joonis 1. Scrumi raamistik [3]

Joonisel 2 on näha tiimi jaotus. Scrumi tiimis on 3 erinevat rolli: toote omanik, *ScrumMaster* ja arendustiim. Toote omanik on isik, kes vastutab selle, eest millised omadused ja funktsionaalsus peavad tootel olema ja millises järjekorras neid luua. Tema on vastutav ka üldise edukuse eest lahenduse arendamisel. *ScrumMaster* aitab kõikidel seotud osapooltel mõista Scrumi väärtusi, põhimõtteid ja tavasid. Tal ei ole kontrolli tiimi üle, kuid ta aitab neil lahendada erinevaid probleeme ja täiustada Scrumi kasutamist projektis. Arendustiimi kuulvad erinevad inimesed, kes vastutavad toote kavandamise, arendamise ja testimise eest. Tiim ise otsustab milline on parim viis eesmärgiks seatud töö saavutamiseks [3].



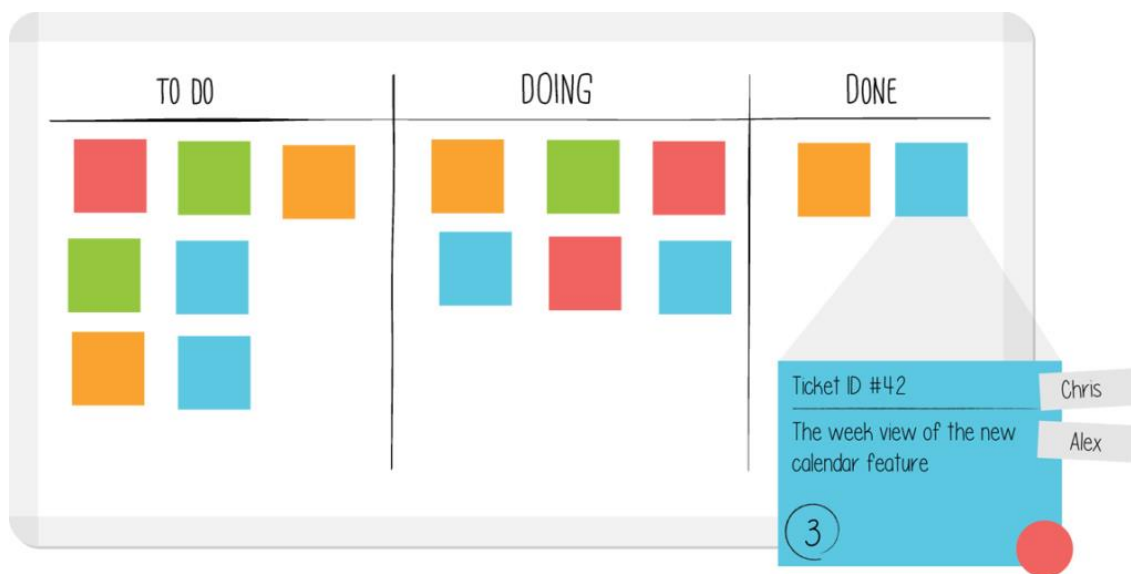
Joonis 2. Scrumi tiim [3].

2.2 Kanban

Kanban on Scrumi kõrval teine suur haru agiilses arenduses. Kanban erineb Scrumist peamiselt selle poolest, et seda metoodikat kasutav tiim võib olla ükskõik kui suur ning tal ei ole kindlaks määratud protsessi kestvust. Kanbani puhul võib muudatusi teha igal ajal ning nende uuenduste edastamine on pidev protsess, mille jaoks ei pea ootama sprindi lõppu [5].

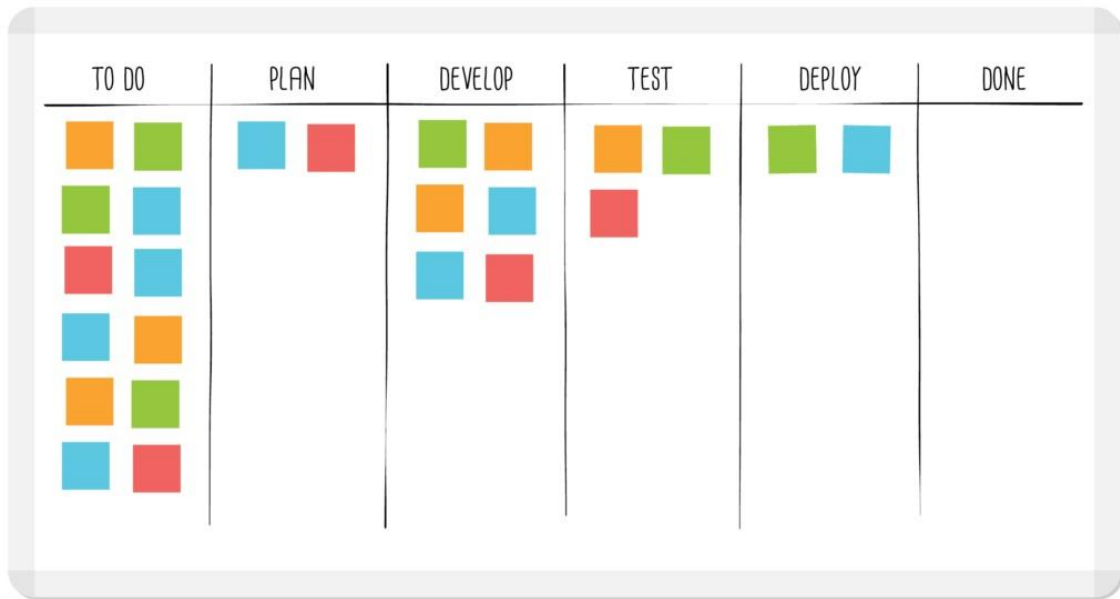
Kanbani kasutusele võtmiseks ei ole vaja muud kui tahvlit, märkmepaberit ja pliiatsit. Kanbani tahvel võib kasutusel olla nii füüsilisel kujul kui ka virtuaalsel kujul arvutis. Oluline on visuaalne efekt, mille tahvli kasutamine annab, sest inimese mälu on piltidele palju vastuvõtlikum kui tekstile. Tahvel on jagatud mitmeks erinevaks osaks ning nende osade arv sõltub juba iga tiimi enda vajadustest. [5, 6, 7].

Joonisel 3 on kujutatud kõige lihtsamat Kanban tahvlit, mis koosneb kolmest osast: mida on vaja teha, mis on tegemisel ja mis on tehtud [1].



Joonis 3. Lihtne Kanban tahvel [21].

Joonisel 4 on kujutatud keerulisemat ja suurema valikuga Kanban tahvlit. Sellel tahvil on eraldi välja toodud ka ülesanded, mis on olulisemad ning mis tuleks esmajärjekorras töösse võtta. Tegemise osa on antud joonisel jagatud mitmeks osaks: kõigepealt on eraldi osana välja toodud ülesanded, mis on arendamisel, seejärel ülesanded, mida parasjagu testitakse ning viimaseks ülesanded, mis on valmis keskkonda paigaldamiseks [1].



Joonis 4. Põhjalikum Kanban tahvel [21].

Kanbani võtmesõnaks on *WIP (work in progress)* ehk ülesanded, mis on hetkel töös. Ilma selleta ei ole Kanbani kasutamisel mõtet. Kanbani tiim keskendub ainult sellele, mis on hetkel töös ning igas töö faasis võib tavaliselt olla 1-2 ülesannet. WIP aitab piirata seda, et korraga on töös liiga palju ülesandeid ning hoiab ära nn pudelikaelade (*bottlenecks*) tekkimise. Selleks, et töösse saaks võtta uusi ülesandeid, tuleb eelnevad ülesanded ära lõpetada. Tööd ootavad ülesanded on koondatud ühte nimekirja ning seejuures on oluline, et kõige tähtsama kaaluga ülesanded oleks nimekirja tipus. See tagab selle, et tööd tehakse maksimaalse efektiivsusega. Kuna tahvlil on korraga väga väike osa ülesandeid, siis annab see hea visuaalse ülevaate tööprotsessist ning selle edenemisest. Probleemide tekkimisel paistavad need kohe silma ning samuti on kõigile teada millega keegi tegeleb [5].

Kanbani juures on oluline ka see, et oma teadmisi jagatakse väga palju üksteisega ning eraldi rollid seal puuduvad. Nii ei teosta ka testimist ainult kvaliteediinsenerid, vaid vajadusel saavad seda teha ka näiteks arendajad [5].

2.3 V-mudel

V-mudel jaguneb oma olemuselt kaheks: verifitseerimine ja valideerimine ning seda võib pidada kosemudeli edasiarenduseks. Mudeli loomise üheks põhjuseks on see, et kosemudeli juures toimus testimine alles viimases etapis, mistõttu leiti vead väga hilja

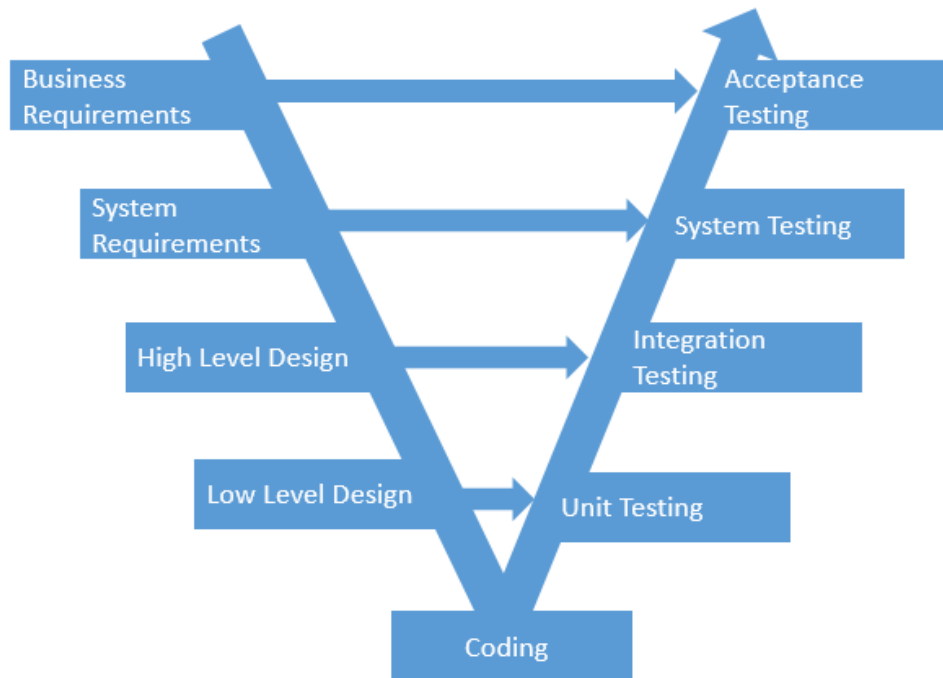
ning nende parandamine oli väga kallis. V-mudeli puhul teevad testijad oma tööd paralleelselt arendajatega ning seda juba projekti algusfaasist. Iga mudeli faas peab olema lõpetatud enne järgmise osa alustamist [8,9]. Tavaliselt jaguneb testimine neljaks osaks:

Komponendi testimine ehk teisisõnu ühiktestimine on mooduli osa, kus iga süsteemi osa testitakse eraldiseisva osana. Peamiselt tegelevad nende osade testide kirjutamisega arendajad. Selle testimise eesmärgiks on kontrollida, et iga süsteemi osa töötaks korrektselt ka siis kui ta on teistest süsteemi osadest eraldi. Ühiktestimine aitab vähendada kulusid, sest nii varases faasis leitud vigade parandamine ja analüüsimine on odavam kui mõnes hilisemas faasis [11, 12].

Integratsioonitestimise osas kontrollitakse süsteemi erinevate osade omavahelist ühildumist. Kui komponendi testimine keskendub süsteemi osadele eraldi, siis integratsioonitestimise juures vaadatakse süsteemi kui tervikut. Lisaks ühildumisele erinevate komponentide vahel kontrollitakse süsteemi erinevate osade koostöövõimet, näiteks operatsioonisüsteem, failisüsteem ja riistvara [9,13].

Süsteemitestimise eesmärgiks on kontrollida kuidas süsteem käitub ning kas see käitumine vastab kirja pandud nõuetele. Süsteemitestimine hõlmab endas nii funktsionaalseid kui ka mittefunktsionaalseid nõudeid. Kui eelnevates osades on testimisega tegelenud pigem arendajad, siis siinkohal on tavaliselt olemas eraldi testijad [14].

Vastuvõtutestimine võib olla tehtud nii lõppkasutaja kui ka kliendi poolt. Selle eesmärgiks on veenduda kas süsteem täidab kasutaja vajadusi ja nõudmisi. Testimisprotsessi lõpus tuleb vastu võtta otsus, kas süsteem on kasutusele võtmiseks piisavalt töökindel [9,15]. Joonisel 5 on kujutatud klassikalise v-mudeli ülesehitus.



Joonis 5. V-mudel [22].

Kuna erinevatel arendusmetoodikatel on oma plussid ja miinused, siis esineb tänapäeval ka erinevate metoodikate segunemist. Näiteks esineb antud lõputöö skoobis oleva ettevõtte igapäevatöös olukordi, kus Scrumi kasutatakse koos v-mudeli elementidega. Suurtes projektides võib testimine olla korraldatud nii, et osa testimisest on korraldatud agiilselt ning teine osa on v-mudeli põhimõttel. Erinevate metoodikate segunemine annab meile rohkem kindlust selle osas, et programmi osad, mis töötavad korrektselt eraldiseisvana, töötavad korrektselt ka siis kui kõik osad kokku pannakse. Lisaks sellele on tavaliseks muutunud ka olukorrad, kus kasutusele võetakse ainult mõned Scrumi põhimõtted.

3 Nõuded testihaldusvahendile

Kuna testihaldusvahendite võimalused ning ASA klientide vajadused on väga erinevad, siis on sobiva testimisvahendi valimiseks vaja kirja panna lõputöös käsitletava ettevõtte jaoks vajalikud nõuded. Alljärgnevalt on kirja pandud nõuded, mis on testihaldusvahendi valikul olulised ASA jaoks. Peatükis 4 koostatakse nende nõuete põhjal erinevate testihaldusvahendite võrdlus. Nõuded on jaotatud kahte erinevasse gruppi, neist esimeses on kõige tähtsamad nõuded ning teises nõuded, mis võiksid olemas olla, kuid ei ole nii olulised. Esimesse gruppi kuuluvad nõuded, mis peaks otsitaval testihaldusvahendil kindlasti olemas olema on eesliitega ON. Teise gruppi kuuluvad väiksema prioriteediga nõuded, mille olemasolu on soovitatav, kuid mis ei ole nii olulised on eesliitega VN.

Järgnevalt on tabelis 1 toodud välja nõuded, mis testihaldusvahendil kindlasti olemas peavad olema.

Tabel 1. Olulised nõuded.

Nõude ID	Lühikirjeldus	Selgitus
ON_1	Testilugude grupeerimine	Võimalus panna sarnase funktsionaalsusega testilood ühte gruppi
ON_2	Testitulemuste märkimine	Võimalus lisada testide käivitamisel testiloo juurde saadud tulemus
ON_3	Agiilse arendusprotsessi sobivus	Võimalus sisestada samaaegselt testitavaid stsenaariume ning testitulemusi, testiplaani jooksvate muudatuste tegemine
ON_4	Lisainformatsiooni lisamine	Lisainfo alla kuuluvad näiteks testimise andmed, eeltingimused, üldine kirjeldus
ON_5	Manuste lisamine	Võimalus lisada testitulemustele erinevaid pilte, logisid, videosid
ON_6	Testiraport	Testiraportite koostamine on võimalik ning testimise tulemustest saab põhjaliku ülevaate (erinevad testitulemused, graafik)

Nõude ID	Lühikirjeldus	Selgitus
ON_7	Vigade raporteerimine	Võimalus sisestada leitud vigu ja vigade täpsemaid kirjeldusi
ON_8	Testitulemused sammu kaupa	Võimalus lisada testimise tulemus iga testiloo sammu juurde, et hiljem oleks lihtsam aru saada, kus täpselt viga tekkis
ON_9	Testilugu -> testitulemused	Kasutaja kirjutab esmalt valmis testilood ning seejärel saab sisestada nende testilugude põhjal testimise tulemused
ON_10	Testiloo korduvkasutamine	Võimalus teha ühest testiloost mitu erinevat versiooni. See on vajalik näiteks erinevate veebibrauseritega testimise puhul

Tabelis 2 on välja toodud nõuded, mis võiksid testihaldusvahendil olemas olla, kuid mis ei ole nii olulised kui Tabelis 1 olevad nõuded.

Tabel 2. Vähemolulised nõuded.

Nõude ID	Lühikirjeldus	Selgitus
VN_1	IT ja ärinõuete lisamine	Võimalus sisestada testihaldusvahendisse nõudeid ning nende sidumise võimalus kindlate testilugudega
VN_2	Nõuded -> testilugu -> testitulemuste jälgitavus	Kasutaja saab lisaks testilugudele ja tulemustele kirja panna ka IT või ärinõuded
VN_3	Kasutajaliidese lihtsus ja mugavus	Testihaldamise protsess on hästi arusaadav ning kasutamine on mugav nii testijal kui projektijuhil
VN_4	Projekti haldusvahendi liidestus	Nt JIRA
VN_5	Liidestus veahaldusvahendiga	Võimalus siduda omavahel testihaldusvahendis asuvat testilugu veahaldusvahendis oleva veaga
VN_6	Rollihaldus	Erinevate õigustega rollide lisamine
VN_7	Automatiseerimisvahendi liidestus	Nt Jenkins, Selenium, Cucumber

4 Testihaldusvahendid ja nende kaardistamine

Kõrge kvaliteediga tarkvara loomisel on testijuhtimisel väga oluline roll. Keerukamate projektide ning erinevates asukohtades olevate tiimide puhul võib head testijuhtimist pidada testimisprojekti edukuse võtmeks. Testijuhtimine sisaldab endas testide planeerimist, loomist, läbiviimist, testimisprotsessi jälgimist ja tulemuste raporteerimist [10].

Testijuhtimise protsessi lihtsustamiseks on loodud erinevaid testihaldusvahendeid, mis pakuvad kasutajale erinevaid võimalusi [10, 16, 17]:

- Testilugude haldamine
- Testitulemuste märkimine
- Vigade raporteerimine ja haldamine
- Testimisprotsessi jälgimine ja ülevaate saamine
- Testilugude värskena hoidmine, sest süsteemis on näha millal viimati testilugusid uuendatud on
- Integreerimine teiste tööriistadega

Kuna testihaldusvahendi kasutaja nõudmised ja vajadused on erinevad ning erinevaid tööriistu on palju, siis on sobiva vahendi leidmine üpris keeruline ja aeganõudev protsess. Testihaldusvahendite omadustest ja puudujääkidest põhjaliku ülevaate saamiseks loodi võrdlev tabel keskendudes antud ettevõttele vajalikele nõuetele. Töö autor leidis ka 2 varasemalt koostatud võrdlustabelit, kuid nende tulemusi sai ta kasutada ainult osaliselt, sest esiteks ei ole neis kõiki testihaldusvahendeid ja samuti ei ole kõik nõuded samad [27]–[28]. Lisaks sellele loodi küsitlus testijatele ja testijuhtidele, et saada infot testihaldusvahendite kasutamise kogemustest Eestis tegutsevates ettevõtetes.

4.1 Testihaldusvahendite võrdlus

Testihaldusvahendite rohkuse tõttu ei ole antud lõputöö raames võimalik kõiki erinevaid vahendeid võrdlusesse võtta. Seetõttu tegi autor nelja erineva testihaldusvahendite top'i põhjal valiku ning valis välja 9 erinevat testihaldusvahendit, mida esialgu võrdlema hakatakse [24]–[27]. Selleks, et saada testihaldusvahendite omadustest parem ülevaade koostas autor nõuete põhjal võrdlema tabeli (joonis 3). Omaduste olemasolu või puudumine pandi kirja testihaldusvahendite koduleheküljel oleva info ning autori enda katsetuste põhjal. Töö hilisemas osas viib autor parimate testihaldusvahenditega läbi ka testülesanded. Testihaldusvahendid, mida autor järgnevalt võrdlema hakkab on: qTest [29], Xray [30], TestLink [31], PractiTest [32], Zephyr [33], TestCollab [34], TestRail [35], TestLodge [36], TestCaseLab [37].

Tabel 3. Testihaldusvahendite võrdlus.

	qTest	PractiTest	Zephyr	TestCollab	TestRail	TestLodge	TestLink 1.9.16	TestCaseLab	Xray
Testilugude grupeerimine	+	+	+	+	+	+	+	+	+
Lisainformatsioon	+	+	+	+	-	+	+	+	+
Manuste lisamine	+	+	+	+	+	+	+	+	+
Testiraport	+	+	+	+	+	+	+	-	+
Agiilne sobivus	+	+	+	+	+	+	+	-	+
Ärinõue -> Testilugu -> Testitulemus	+	+	+	-	-	+	+	+	+
Testilugu -> Testitulemus	+	+	+	+	+	+	+	+	+
Nõuete sidumine testilugudega	+	+	+	-	-	+	+	+	+
Kasutajaliidese mugavus/lihtsus	+	+	+	-	+	+	-	+	+
Rollihaldus	+	+	+	+	+	+	+	-	+
JIRA liidestus	+	+	+	+	+	+	+	+	+
Testitulemuste märkimine	+	+	+	+	+	+	+	+	+
Testitulemuste märkimine sammu kaupa	+	+	+	+	+	-	+	-	+
Liidestamine veahaldusvahenditega	+	+	-	+	+	+	+	+	-
Vigade raporteerimine	+	+	+	+	+	+	+	+	+
Liidestus automatiseerimisvahenditega	+	+	+	-	+	-	-	-	+

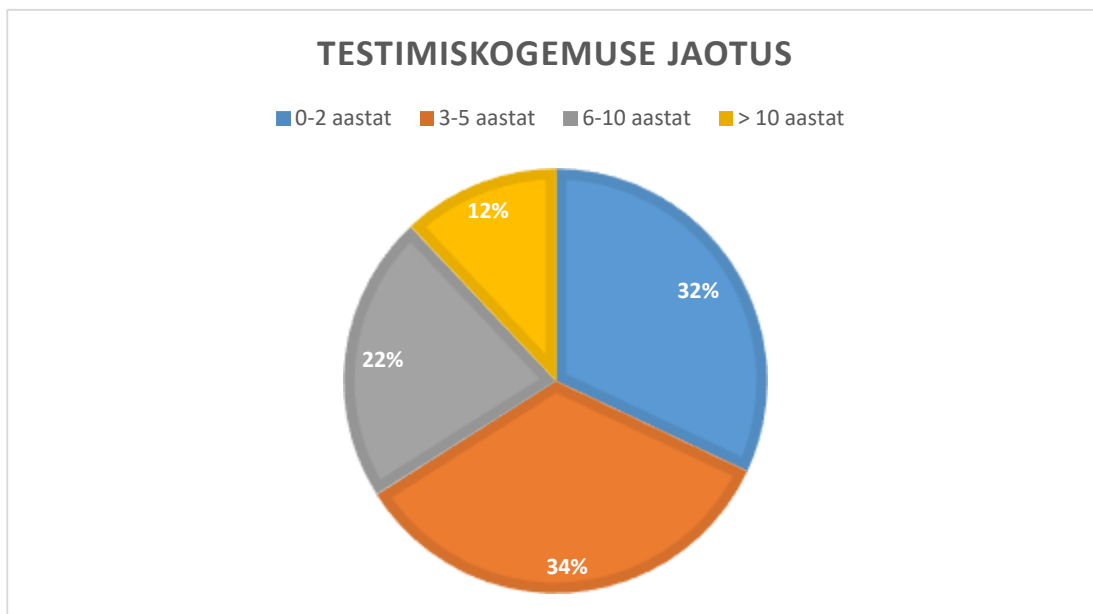
Testihaldusvahendite võrdluse tabelit (tabel 3) analüüsid selgub, et kõige paremini vastavad nõuetele testihaldusvahendid qTest, PractiTest, Zephyr ja Xray. Esimesed 2 vastavad kõikidele kolmandas peatükis esitatud nõuetele. Testihaldusvahendid Zephyr ja Xray vastavad kõikidele nõuetele peale veahaldusvahendi nõude. Kuna aga Zephyr'i ja Xray puhul on tegemist JIRA lisadega ning JIRA ise on veahaldusvahend, siis ei ole antud funktsionaalsus eraldi vajalik.

Testihaldusvahendite hindasid uurides selgus, et qTest'i hinnakiri ei ole avalikult kättesaadav ning seetõttu ei saa seda antud töö tulemustes kajastada. Zephyr'i puhul sõltub hinnakiri asjaolust, kas testihaldusvahendit soovitakse kasutada JIRA pilves (*cloud*) või enda serveris. Pilveteenusena maksab teenus 15 kasutaja korral 35 dollarit kuus, kuhu tuleb lisaks osta JIRA pilveteenus, mis 15 kasutajaga maksab 75 dollarit kuus [38]–[39]. Enda serverit kasutades on võimalus teenust osta 25 kasutajale, mis tähendab, et JIRA eest tuleb tasuda ühekordne makse 1800 dollarit ning Zephyr'i eest tuleb tasuda ühekordne makse 750 dollarit [40]–[41]. PractiTest testihaldusvahendi kasutamise õiguse eest Entepriise valikuga 15 testijale tuleb tasuda 45 dollarit kuus [42].

4.2 Testihaldusvahendite kasutamine Eesti ettevõtetes

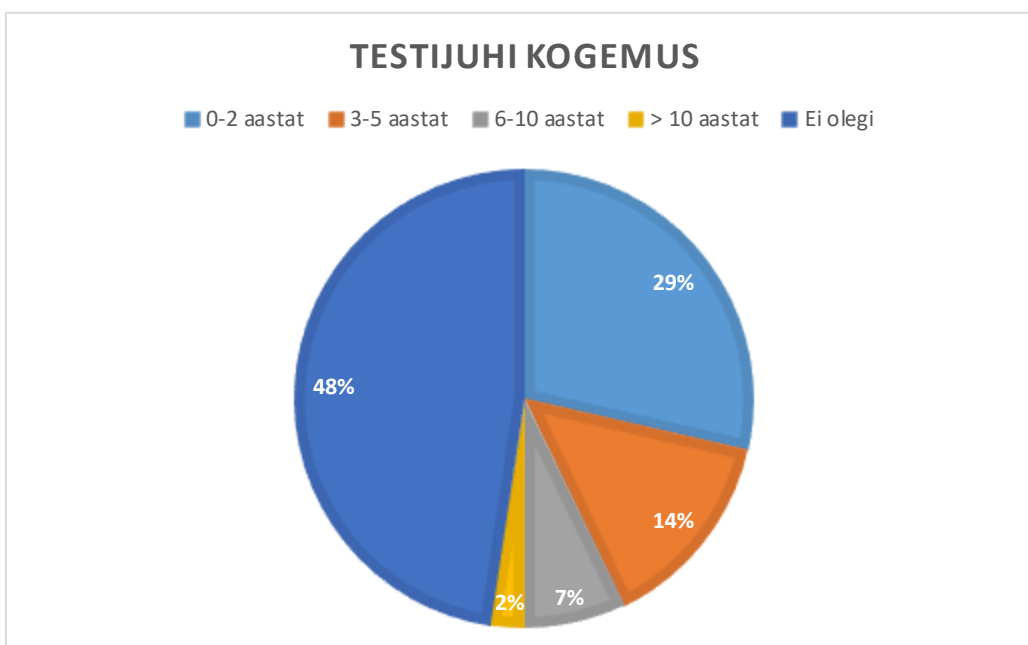
Testihaldusvahendi küsimustiku loomise põhjuseks on asjaolu, et puudub teadmine, kas ja kui palju Eestis asuvates ettevõtetes testihaldusvahendeid kasutatakse. Küsimuste loomisel lähtus autor põhimõttest, et saadud vastused talle olukorrast parema ülevaate annaksid. Küsimusi oli kokku 9 ning 8 neist olid valikvastustega. Kuna autori valitud testihaldusvahendid ei ole kindlasti ainsad, mida kasutatakse, siis oli üheks valiku variandiks võimalus ka ise vabatekstiline väärtus sisestada. Testihaldusvahendi küsimused on kujutatud lisa nr 1.

Töö autori poolt loodud küsimustikule vastas kokku 50 inimest. Vastanutest rohkem kui pooled (66%) olid alla 5 aastase testimiskogemusega. Joonisel 6 on näha täpsem jaotus inimeste testimiskogemuse osas.



Joonis 6. Testimiskogemuse jaotus.

Vaadates joonist 7, selgub, et testijuhi kogemus oli neist veidi rohkem kui pooltel (52.38%). Lisaks selgus, et tervelt 80% küsimustikule vastanutest oli kokku puutunud arendustestimisega ning 62% vastuvõtutestimisega. Väiksem osa ehk 20% oli kokku puutunud ka äritestimisega.

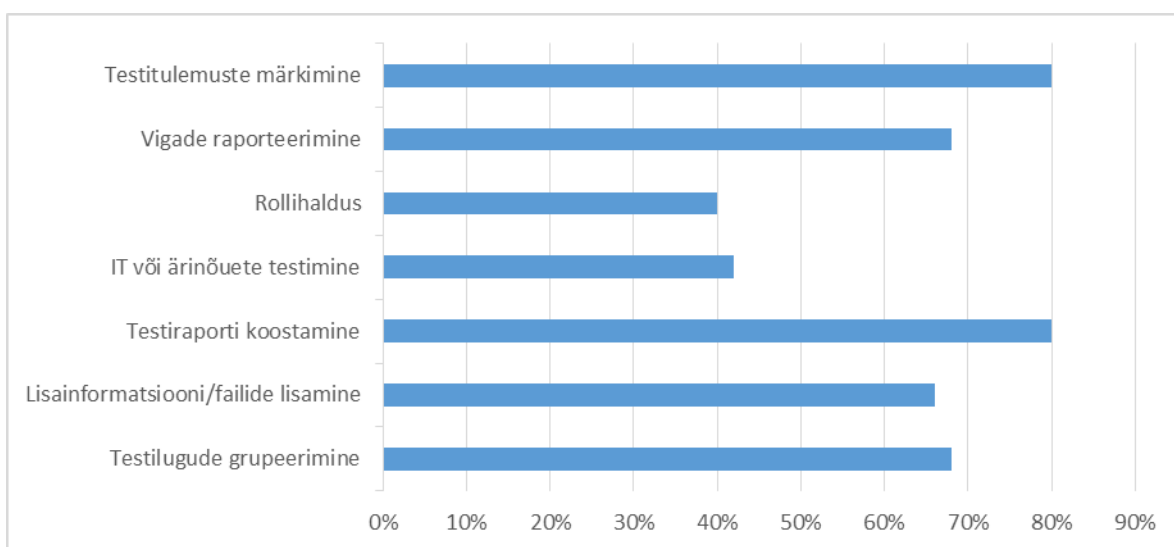


Joonis 7. Testijuhi kogemus.

Vastanute seas oli 9 inimest, kes polnud ühegi testihaldusvahendiga kunagi kokku puutunud ning lisaks 5 inimest, kes ei olnud ühegi töö autori poolt nimetatud

testihaldusvahendiga kokku puutunud. Kõige levinumaks testihaldusvahendiks oli TestLink, mille kasutamisega oli kokku puutunud 56% inimestest. TestLinki levinud kasutamise põhjuseks on tõenäoliselt ka asjaolu, et tegemist on nimekirjas ainsa tasuta kättesaadava tarkvaraga.

Enim soovitaksid testijad teistele kasutamiseks TestLink'i ja Zephyr'i, kuid samas oli TestLink ka peamiseks tööriistaks, mille kasutamisega kõige vähem rahule jäädi. Tulemustest selgus, et testihaldusvahendite erinevaid funktsionaalsusi kasutatakse palju. Joonisel 8 selgub, et kõige olulisemaks peetakse testiraporti koostamise (80%) ja testitulemuste märkimise (80%) võimalust. Veidi vähemtähtsad on testilugude grupeerimine (68%), failide lisamine (66%) ja vigade raporteerimise (68%) võimalused.



Joonis 8. Testihaldusvahendi omaduste olemasolu olulisus.

Lisaks uuriti küsitluses seda, kui tihti erinevaid testihalduse funktsionaalsusi testijad oma töös tegelikult kasutavad. Tulemustest joonisel 9 selgub, et kõige rohkem kasutatakse igapäevaselt testitulemuste märkimise võimalust (59.18%). 1–2x kuus kasutatakse kõige rohkem testilugude grupeerimise (40.82%) ja testiraporti koostamise (38%) võimalust. Tulemustest selgub ka, et nõuete lisamise võimalust kasutatakse kõige vähem. Tulemused on ootuspärased, sest testimisega tegeletakse üldiselt igapäevaselt, kuid näiteks testiraporteid koostatakse tavaliselt testimisprotsessi lõpus. Joonisel 9 on näha ka vigade raporteerimise kasutamise küsimus, kuid töö autor otsustas selle peale esimeste vastuste saamist eemaldada, sest vigade raporteerimise tihedus sõltub ikkagi süsteemist ja sellest kui palju seal esinevaid vigu leitakse.

	Mitte kunagi	Harva (1-2x aastas)	Vahepeal (1-2x kuus)	Igapäevaselt	Total
Testilugude grupeerimine	18.37% 9	22.45% 11	40.82% 20	18.37% 9	49
Lisainformatsiooni/failide lisamine	14.29% 7	14.29% 7	32.65% 16	38.78% 19	49
Testiraporti koostamine	22.00% 11	18.00% 9	38.00% 19	22.00% 11	50
IT või ärinõuete lisamine	36.17% 17	17.02% 8	31.91% 15	14.89% 7	47
Vigade raporteerimine	0.00% 0	12.50% 1	12.50% 1	75.00% 6	8
Testitulemuste märkimine	10.20% 5	6.12% 3	24.49% 12	59.18% 29	49

Joonis 9. Testihaldusvahendi funktsionaalsuste kasutamise tihedus.

Testitulemuste kasutamise kogemuse küsimustikust saab järeldada, et Eestis kasutatakse kõige rohkem TestLink'i testihaldusvahendit. TestLink on küll populaarne, kuid ilmselt seisneb selle populaarsus selles, et paremat tasuta tarkvara ei ole leitud. Palju kasutatakse ka JIRA't ning selle erinevaid pluginaid, kuid üldiselt on tasuline testihaldusvahendite tarkvara aga pigem vähe levinud. Tulemustest selgus, et testihaldusvahendit pidasid ebaoluliseks eelkõige arendustestijad ning testijad, kes polnud üldse testihaldusvahenditega kokku puutunud.

5 Testihaldusvahendi väljavalimine

Peatükis 4.1 tehtud testihaldusvahendi võrdleva tabeli põhjal valiti välja 4 tööriista, mille vastavus kolmandas peatükis kirjeldatud nõuetele oli kõige täpsem. Ülesannete läbiviimiseks olevad programmid on tasuta kättesaadavad testversioonid. Antud töö osast on välja jäänud PractiTest, sest prooviversiooni saamiseks tehtud taotlusele ettevõtte ei vastanud ning seetõttu polnud võimalik ülesandeid selle testihaldusvahendiga läbi viia. Kolm testihaldusvahendit valiti võrdleva tabeli põhjal ning Xray sai valitud küsimustikus tehtud testihaldusvahendite soovitude põhjal. Ülesanded loodi töö autori enda töökogemusele tuginedes.

5.1 Testülesanded

Testülesanded loodi Maanteeameti e-teeninduse põhjal. Eelnevalt nimetatud infosüsteemi valiku puhul sai määravaks see, et tegemist on mittetriviaalse avaliku teenusega ning töö autor on sellega oma töös varasemalt kokku puutunud. Ülesannete seas leidub nii tüüpilisi testilugusid, kui ka teste mis on koostatud selleks, et kontrollida testihaldusvahendi vastavust ettevõtte poolt seatud nõuetele. Testilugude formaat vastab ISO 29119 standardile [23].

Ülesanne 1

Tabelis 4 on kirjeldatud testilugu TC-1 e-teenindusse sisselogimise kohta. Ülesande eesmärgiks on kontrollida testihaldusvahendi vastavust nõudele ON_10. Kuna sisselogimine on väga oluline osa toimingute tegemiseks, siis tuleb seda testilugu kindlasti erinevate veebibrauseritega (IE 11, FF 53, Chrome 57) korrata ja seetõttu on oluline, et testihaldusvahend võimaldaks hõlpsasti luua antud testiloost mitut versiooni.

Tabel 4. Sisselogimine mobiil-ID'ga.

TC-1 Sisselogimine Mobiil-ID'ga	
Eesmärk: Kontrollida, et sisselogimine Mobiil-ID kasutamisega toimib	
Eeltingimused: Kasutaja on e-teeninduse avalehel ning ei ole sisse logitud. Kasutajal on olemas mobiil-ID	
Tegevus	Oodatud tulemus
Kasutaja vajutab nuppu „Sisenen“	Kuvatakse sisselogimise leht
Kasutaja sisestab oma mobiiltelefoninumbri	Mobiiltelefoninumber on sisestatud
Kasutaja vajutab nuppu „Mobiil-ID“	Kasutajale kuvatakse sisenemine Mobiil-IDga leht, kus on kontrollkood. Mobiiltelefoni ekraanile kuvatakse sama kontrollkoodiga aken, kuhu saab sisestada pin1 koodi
Kasutaja sisestab telefonis pin1 koodi	Kasutaja on e-teenindusse sisse loginud

Ülesanne 2

Tabelis 5 on kirjeldatud testilugu TC-2 juhiloa vahetuse protsessi kohta. Juhiloa vahetamise puhul on oluline, et kasutajal on eelnev juhiluba, mille ta soovib välja vahetada. Selle ülesande juures soovib autor kontrollida kahe erineva nõude täitmist. Manuse ehk antud juhul ekraanipiltide lisamisega testiloole kontrollitakse nõude ON_5 täitmist ning andmebaasipäringute teksti lisamisega kontrollitakse nõude ON_4 täitmist.

Tabel 5. Juhiloa vahetamise protsess.

TC-2 Juhiloa vahetus	
Eesmärk: Kontrollida, et kasutaja saab e-teeninduses juhiluba vahetada	
Eeltingimused: Kasutaja on e-teenindusse sisse loginud ning tal on olemas kehtiv juhiluba	
Tegevus	Oodatud tulemus
Kasutaja vajutab nuppu „Juht“	Kasutajale kuvatakse juhi toimingute leht
Kasutaja vajutab nuppu „Juhiloa vahetamine“	Kasutaja suunatakse juhiloa vahetuse lehele
Kasutaja teeb valiku, et vahetatav juhiluba on olemas	Kasutajale kuvatakse foto ja allkirja väli
Kasutaja valib ja kinnitab sobiva foto ja allkirja	Kasutaja suunatakse tellimisviisi lehele

Tegevus	Oodatud tulemus
Kasutaja valin tavatellimuse viisi ja kinnitab valiku	Kasutajale kuvatakse kättesaamiskoha valik
Kasutaja valib kättesaamiskohaks büroo ning rippmenüüst büroo, kuhu uus juhiluba saadetakse. Kasutaja kinnitab valiku	Kasutajale kuvatakse riigilõivu valik
Kasutaja vajutab endale sobiva internetipanga nimetusel	Kasutaja suunatakse internetipanka
Kasutaja teeb internetipangas makse ning lõpus vajutab nuppu „Tagasi kaupmehe juurde“	Kasutajale kuvatakse teade taotluse eduka esitamise kohta ning info juhiloa kättesaamise kohta

Ülesanne 3

Ülesande eesmärgiks on kontrollida testihaldusvahendi vastavust nõudele ON_2. Aluseks võetakse eelnevalt kirjeldatud testilugu TC-1. Testitulemus lisatakse kogu testiloo kohta üldiselt. Testilugu esmakordsel käivitamisel märgitakse tulemuseks, et testiloo läbimine ei õnnestunud. Testiloo teisel läbimisel märkida tulemuseks, et testilugu on korrektselt läbitud ja saab oodatud tulemuse. Kui testitulemuseks saab märkida nii läbimise kui ka läbi kukkumise olekut, siis on ülesande täitmine edukas.

Ülesanne 4

Ülesande eesmärgiks on kontrollida testihaldusvahendi vastavust nõudele ON_8. Aluseks võetakse varasemalt kirjeldatud testilugu TC-2, sest see sisaldab palju erinevaid samme. Ülesande raames määratakse testitulemus iga sammu kohta eraldi. Vähemalt 1 sammudest määratakse läbikukkunuks ning vähemalt 1 läbituks.

Ülesanne 5

Ülesande eesmärgiks on kontrollida testihaldusvahendi vastavust nõudele ON_7. Leitud vigade raporteerimise võimalus on testimisprotsessis väga oluline, sest nii on kohe näha kus on viga tekkinud ning millise testiloo see seotud on. Seekord võetakse aluseks testilugu TC-1 millele proovitakse lisada uut viga. Vea raporteerimine võib toimuda mitut erinevat moodi. Üheks võimaluseks on, et viga saab raporteerida testihaldusvahendi enda raames. Teine variant on see, et vea raporteerimisel suunatakse kasutaja kas JIRAsse või mõnda teise veahaldusvahendisse, kuhu saab täpsemalt kirja

panna leitud vea parameetrid. Vea raporteerimise raames kontrollitakse ka vastavust nõuetele VN_4 ja VN_5.

Ülesanne 6

Selle ülesande eesmärgiks on kontrollida mitut erinevat nõuet. Esiteks kontrollitakse uue testiloo lisamisega vastavust nõudele ON_9 ehk kõigepealt sisestatakse uus testilugu TC-3, mis on kirjeldatud tabelis 6 ning seejärel märgitakse testi tulemus. Kui protsessi teostamine sellises järjekorras on võimalik, siis loetakse ülesande esimene pool täidetuks. Kui antud testiloole on võimalus eelnevalt lisada ka testiloo nõuete kirjeldus, siis loetakse läbituks ka ülesande teine pool, mis vastab nõudele VN_2.

Tabel 6. Kasutajate muutmise protsess.

TC-3 Kasutajate muutmine	
Eesmärk: Kontrollida, et sõiduki kasutajate muutmine on võimalik	
Eeltingimused: Kasutaja on e-teenindusse sisse loginud ning tal on olemas sõiduk, mille omanik ta on	
Tegevus	Oodatud tulemus
Kasutaja vajutab küljeribal „Sõiduk“ ikoonile	Kasutajale kuvatakse sõidukiga seotud valikute leht
Kasutaja vajutab „Sõidukite nimekirja“ valikule	Kasutajale kuvatakse temaga seotud sõidukite nimekiri
Kasutaja teeb valiku „Toimingud sõidukiga“	Kasutajale kuvatakse toimingute nimekiri
Kasutaja valib toingu „Kasutajate muutmine“	Kasutajale kuvatakse kasutajate muutmise leht
Kasutaja lisab sõidukile uue tavakasutaja ja vajutab nuppu „Edasi“	Kasutaja suunatakse tunnistuse lehele
Kasutaja teeb valiku büroo kasuks ning valib menüüst „Jõhvi“ büroo, vajutab nuppu „Edasi“	Kuvatakse kasutajate koondvaade
Kasutaja vajutab nuppu „Edasi“	Kasutajale kuvatakse kinnituse teade
Kasutaja vajutab nuppu „Kinnita“	Kasutaja suunatakse riigilõivu lehele
Kasutaja teeb valiku pangas tasumise kasuks ning vajutab nuppu „Suundun maksta“	Kasutajale kuvatakse erinevate pankade valik
Kasutaja teeb valiku „SEB“ panga kasuks ning sooritab makse	Kasutajale kuvatakse teade riigilõivu tasumisest ning muudatuste kandmisest liiklusregistrisse

Ülesanne 7

Agiilse arendusprotsessi sobivuse testimiseks lisatakse uus testilugu alles siis kui eelnevaid testilugusid juba käivitatakse. Antud testiloo puhul on tegemist lihtsa testiloo, mis peab sobima sisselogimise toimingu juurde. Uue testiloo lisamine peab olema võimalik ka olemasolevasse testiplaani. Lisaks võiks olemas olla võimalus testiloo käivitamiseks ilma, et see oleks lisatud testiplaani. Nõue, mida antud tabelis 7 kujutatud testiloo TC-4 kaetakse on ON_3.

Tabel 7. Sisselogimine ID-kaart.

TC-4 Sisselogimine ID-kaardiga	
Eesmärk: Kontrollida, et kasutaja saab sisse logida id-kaardiga	
Eeltingimused: Kasutaja on e-teeninduse avalehel ning ta ei ole sisse logitud	
Tegevus	Oodatud tulemus
Kasutaja vajutab nuppu „Sisenen	Kasutajale kuvatakse sisselogimise leht
Kasutaja vajutab nuppu „ID-kaart“	Kontrollitakse isikule vastava id-kaardi olemasolu, kasutajale kuvatakse pin1 sisestamise aken
Kasutaja sisestab korrektse pin1	Kasutaja on e-teenindusse sisse logitud

Ülesanne 8

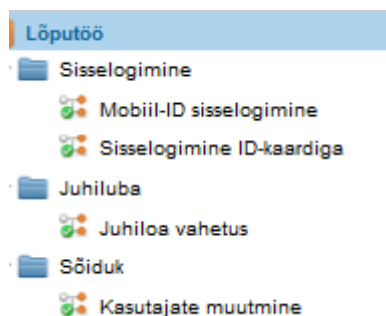
Ülesande eesmärgiks on kontrollida testihaldusvahendi vastavust nõudele ON_1. Kuna valminud on 2 testilugu, mis on seotud sisselogimisprotsessiga, 1 testilugu, mis on seotud sõiduki toimingutega ning 1 testilugu, mis on seotud juhiloa toimingutega, siis proovitakse antud ülesande raames lisada kolme erinevasse gruppi: sisselogimine, sõiduk ja juhiluba. Kui grupeerimine õnnestub, siis on ülesande lahendamine edukas.

Ülesanne 9

Ülesande eesmärgiks on kontrollida testihaldusvahendi vastavust nõudele ON_6. Hea testiraport sisaldab testide käivitamise tulemusi testilugude kaupa. Lisaks võiks olemas olla graafiline ülevaade testitulemustest, et oleks lihtne aru saada, kui suur osa testitulemustest sai positiivselt läbitud ning kui suur osa testidest kukkus läbi.

5.2 qTest Manager

Võrdleva tabeli tulemusena selgus, et qTest vastab kõikidele nõuetele, mis töö alguses kirja pandud said. Selleks, et aga veenduda, et see tegelikult ka nii on, viidi töö autori poolt läbi mõned testülesanded. Selles peatükis kirjeldatakse läbi viidud testide tulemusi. Alljärgnevalt on joonisel 10 kujutatud qTesti testide nimekiri testi disainimise lehel.



Joonis 10. Testide nimekiri.

Ülesannete läbiviimise tulemusena selgus, et qTest vastab ASA poolt ette seatud nõuetele. Testihaldusvahend sobib kasutamiseks nii klassikalises arendusprotsessis, kus muutuseid on vähe ning kõik toimingud tehakse järjest ära kui ka agiilsesse arendusprotsessi, kus toimuvad pidevalt muutused ja täiendused.



Testilugude läbimisega seotud nõuete täitmine

Esmalt kirjeldatakse tulemusi, mis on seotud testilugude loomise protsessiga. Testimise tulemusena selgus, et testiloo sisestamiseks tuleb eelnevalt teha uus testimoodul, sest muidu testilugu sisestada ei saa. Testiloo loomisel oli võimalus jagada testilugu erinevateks sammudeks ning iga sammu juurde kirjutada ka oodatud tulemus. Testiloo valmimisel oli võimalus kasutajalugu kas lihtsalt salvestada või määrata olek, et testilugu on valmis ka testimise käivitamise protsessi jaoks.

Manuste lisamine on lihtne ning lisada saab kuni 50MB suurust faili. Manust saab lisada nii nõude juurde kui ka konkreetse testiloo alla. Kirjelduse väljale on võimalus lisada andmebaasi päringuid, mida testi läbimiseks vajalike andmete saamiseks vaja on.

Tulemuse valimiseks on 2 võimalust. Kiirem variant on sisestada tulemus kogu testiloo kohta üldiselt. Põhjalikum on aga tulemuste määramine sammu kaupa, mille viimaseks

sammuks on valik, kas testiloo läbimine üldiselt oli edukas või mitte. Jooniselt 11 testide käivitamise ajalugu vaadates on staatusest näha millised testilood läbiti ja millised mitte. Lisaks on ajaloo alt näha kui testimise käigus raporteeriti viga.

EXECUTED START	EXECUTED END	STATUS	
		<input type="text"/>	
05/02/2017	05/02/2017	PASSED	
05/02/2017	05/02/2017	BLOCKED	
05/02/2017	05/02/2017	FAILED	
05/02/2017	05/02/2017	PASSED	
05/02/2017	05/02/2017	FAILED	
05/02/2017	05/02/2017	PASSED	
		UNEXECUTED	

Joonis 11. Käivitamiste ajalugu.

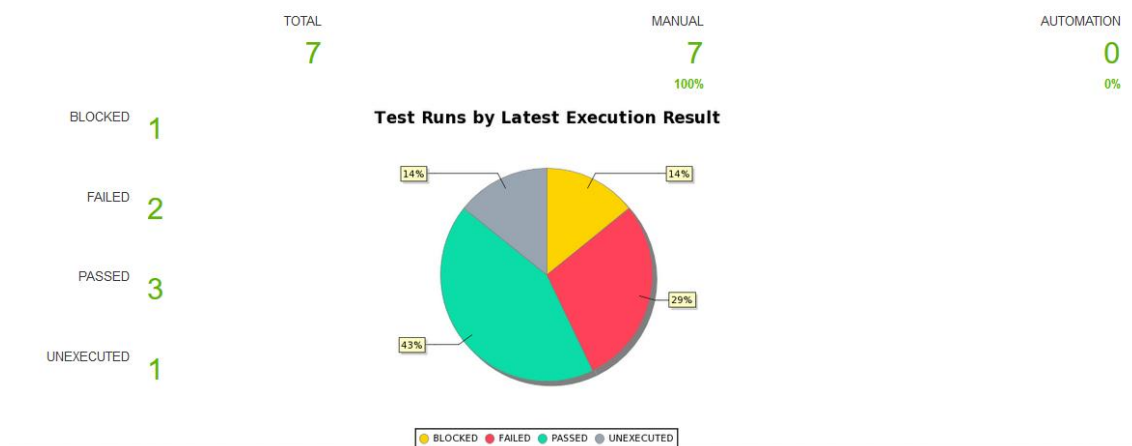
Muude nõuete täitmine

Leitud vigade raporteerimine toimus antud ülesannete juures testihaldusvahendi enda sees ning integreeritud vahendeid ei kasutatud. Raporteeritud viga saab siduda konkreetse testiloo sammuga nii, et on kohe näha kus viga tekkis. Kuna lõputöö raames on tegemist prooviversioonide kasutamisega, siis seadete all oli olemas koht kus sai JIRA ja veahaldusvahenditega integreerida, kuid nende vahendite puudumise tõttu kasutaja ülesannete raames raporteerimist nendega ei proovinud.

Ülesannete lahendamise tulemusena selgus, et ülesannete teostamine on võimalik järjekorras testilugu -> testitulemus kui ka nõue -> testilugu -> testitulemus. Samuti on võimalus uusi teste iga hetk testiplaani juurde lisada ning ülesandeid saab käivitada ka üksikult nii, et neil puudub konkreetne testiplaan. See teeb testilugude läbimise paindlikumaks. Testilugusid saab grupeerida erinevatesse kaustadesse sel moel, et näiteks sisselogimisega seotud toimingud on koondatud ühte, sõidukiga teise ning juhilubadega kolmandasse kausta.

Viimase sammuna kontrolliti testitulemuste kuvamist raportis. Testitulemusi saab vaadata nii kõikide käivitamiste kohta koos kui ka kindla testimistsükli kaupa. Tulemused kuvatakse kasutajale tabelina, kus on kirjas testitsükkel, käivitamise tulemus ning kas tegemist on manuaalse või automaatse testiga. Lisaks sellele kuvatakse

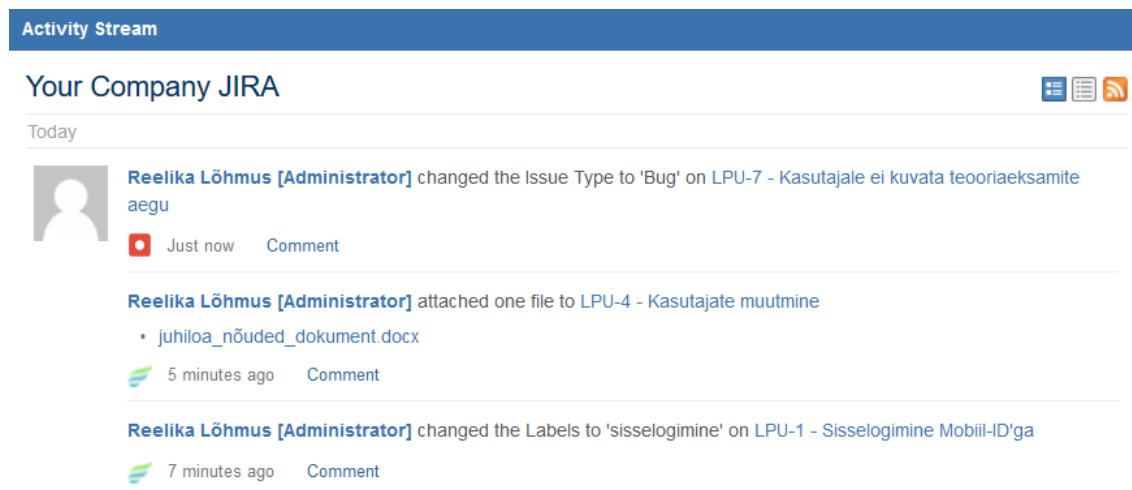
graafiline pilt testimise tulemustega. Joonisel 12 tuuakse välja testimise tulemused, kust selgub, et ühe testiloo käivitamisel tekkisid probleemid, mistõttu ei saanud antud testilugu läbida, 2 testilugu kukkusid läbi, 3 testilugu läbiti edukalt ning ühte testilugu ei ole ühtegi korda käivitatud.



Joonis 12. Testitulemuste kujutamine graafiliselt.

5.3 Zephyr for JIRA

Teiseks testitavaks haldusvahendiks sai võrdleva tabeli põhjal Zephyr. Testide läbiviimiseks kasutati Zephyr for JIRA Cloud versiooni. Joonisel 13 kujutatakse testihaldusvahendi avalehe osa, kus on ülevaade kõikidest viimastest tegevustest.



Joonis 13. Viimased tegevused.

Tegemist on vahendiga, mida on mugav kasutada, kuid mis esmakordsele kasutajale, kes pole varem JIRA süsteemidega kokku puutunud, võib päris keeruline olla. Kui aga JIRA süsteemid on varasemast tuttavad, siis ei teki vahendi kasutamisega raskusi.

Testilugude läbimisega seotud nõuete täitmine

Testilugude loomise juures on antud testihaldusvahendil lisaks tegevuse sammude ja oodatud tulemuste lahtritele olemas ka koht testandmete sisestamiseks (Joonis 14). See funktsionaalsus on väga mugav juhul kui on palju testandmeid, mida tuleb erinevatest testi sammudes kasutada. Ühte testilugu on võimalik korduvalt kasutada. Testiloo sisestamise järgselt on võimalus lisada see kindlasse testitsükklisse (*Test cycle*) või käivitada testilugu ilma testiplaani lisamata.

Test Details				
	Test Step	Test Data	Expected Result	
1	Kasutaja vajutab nuppu „Juht“	isikukood 49004203112	Kasutajale kuvatakse juhi toimingute leht	
2	Kasutaja vajutab nuppu „Juhiloa vahetamine“		Kasutaja suunatakse juhiloa vahetuse lehele	
3	Kasutaja teeb valiku, et vahetatav juhiluba on olemas		Kasutajale kuvatakse foto ja allkirja väli	
4	Kasutaja valib ja kinnitab sobiva foto ja allkirja		Kasutaja suunatakse tellimisviisi lehele	

Joonis 14. Testiloo lisamise sammud.

Manusena on võimalik lisada nii pildifaile kui ka videosid. Testitulemuste sisestamine on võimalik nii sammhaaval kui ka kogu testiloo kohta üldiselt. Testide käivitamisel on võimalus lisaks testitsükklile määrata ära, millise süsteemi versiooni põhjal see testilugu läbitakse.

Muude nõuete täitmine

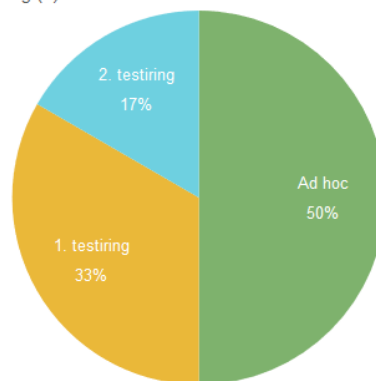
Vigade raporteerimiseks on JIRA'1 põhjalik veahalduse osa. Vigu on võimalik raporteerida testiloo käivitamise ajal ning loodud vea saab siduda kindla testiloo sammuga. Testilugude grupeerimine ei toimu Zephyr'i puhul kaustadesse lisamise võimalusena vaid hoopis testilugudele siltide (*labels*) lisamise kaudu. Nimelt saab testilugude haldamise lehel otsida testilugusid, mis on seotud kindla sildiga.

Ülesannete lahendamise tulemusena selgus, et ülesannete teostamine on võimalik järjekorras testilugu -> testitulemus. Testihaldusvahendi funktsionaalsuste juures oli kirjas, et Zephyr võimaldab nõudeid testilooga siduda, kuid töö autor testihaldusvahendi katsetamise ajal seda võimalust ei leidnud. Küll aga on selline võimalus olemas Zephyr Enterprise juures.

Testiraporti koostamiseks pakub Zephyr erinevaid võimalusi. Esiteks pakub antud testihaldusvahend võimaluse saada ülevaade testimistulemustest testimistsükklite kaupa. Tsükli juures on kuvatud ka testiloo läbimisel leitud vead. Zephyr pakub ülevaadet ka testide käivitamise hulgast ja nende tulemustest päevade kaupa, testide käivitamise ülevaadet erinevate testijate poolt ning üldist käivitatud testide listi, kus on kuvatud kõik tulemused olenemata testitsüklist. Joonisel 15 kuvatakse testide käivitamise osa protsentuaalselt ehk kui suure osa kogu testimisprotsessist moodustasid erinevad testitsüklid.

TEST EXECUTIONS BY TEST CYCLE

● Ad hoc (3) ● 1. testiring (2) ● 2. testiring (1)

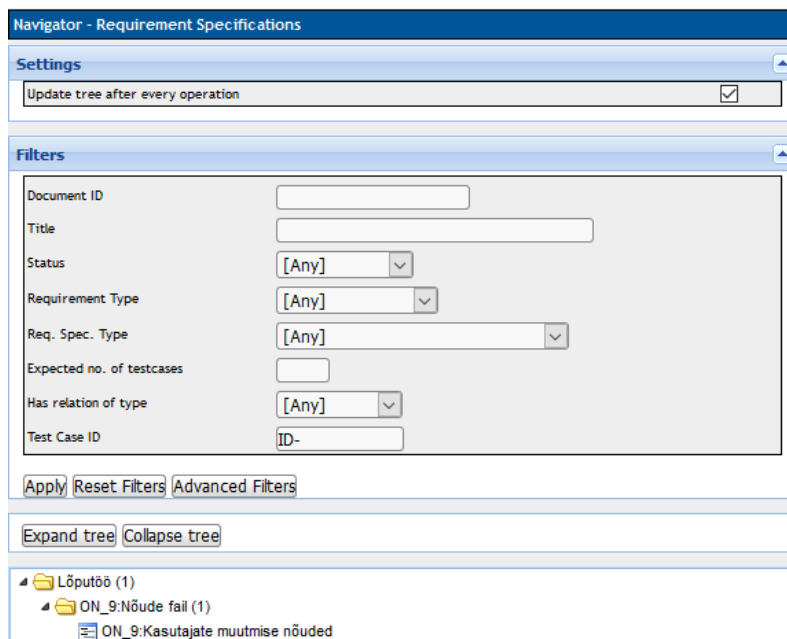


Joonis 15. Testide käivitamine erinevate testitsükklite poolt.

5.4 TestLink

TestLink'i esialgse kasutamise puhul on tegemist keerulisema testihaldusvahendiga kui eelkirjeldatud, sest kasutusjuhendit ei ole ja funktsionaalsuste olemasolu tuleb ise süsteemist üles leida. Kuna aga töö autor oli varasemalt TestLink'i erinevate versioonidega kokku puutunud, siis lähtus ta programmi kasutama õppimisel oma eelnevatest kogemustest. Selle testihaldusvahendi puhul on võrreldes varasema versiooniga jäänud samaks see, et erinevaid samme tuleb testilugude loomiseks ja

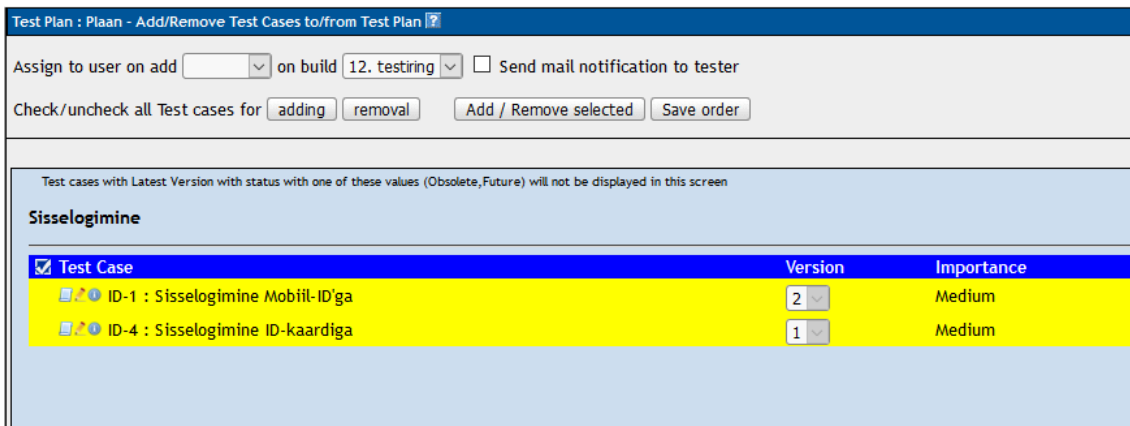
läbimiseks endiselt palju teha ning süsteemi pole kõige lihtsam kasutama õppida, kuid antud testihaldusvahendit on aja jooksul kasutajasõbralikumaks muudetud. Üheks suurimaks muudatuseks on see, et testilugu saab läbida sammhaaval. Vahendil on olemas eraldi koht nõuete haldamiseks (joonis 16) ja nõuete sidumise võimalus testilooga.



Joonis 16. Nõuete haldamine.

Testilugude läbimisega seotud nõuete täitmine

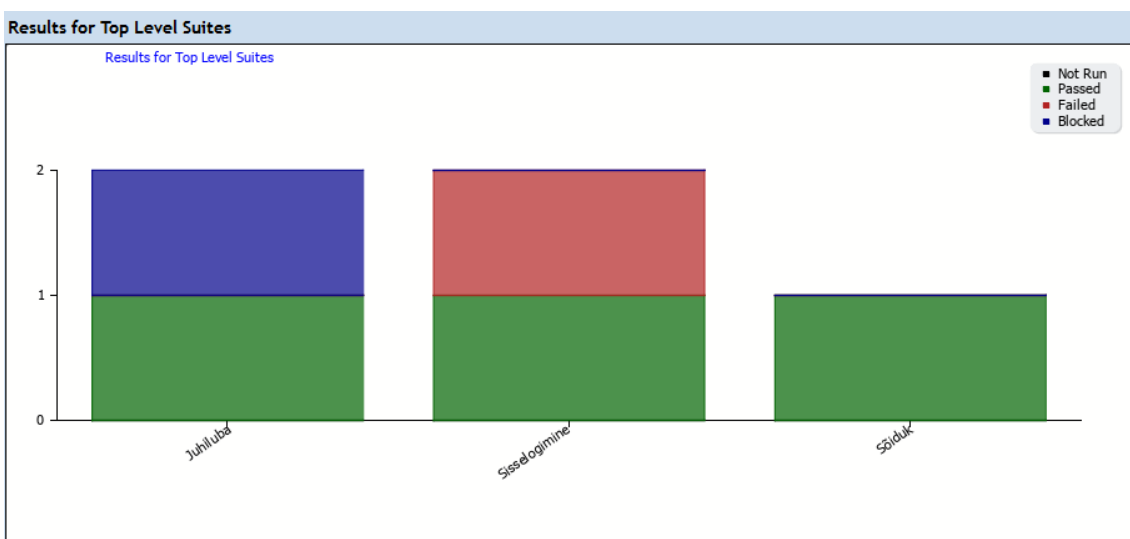
Ülesannete lahendamise tulemusena selgus, et ülesannete lahendamisel on võimalus märkida tulemusi nii sammhaaval kui ka kogu testiloo kohta korraga. Kui testitulemused märkida sammhaaval, siis tuleb kindlasti ära märkida ka kogu testi tulemuse staatus, sest muidu käivitamise tulemust ei näe. Testilugusid saab korduvalt kasutada ning ühe testiplaani raames mitmesse erinevasse testitavasse versiooni (*build to execute*) lisada. Testandmed saab lisada kokkuvõtte alla ning manuste lisamise võimalus on nii nõuete kui ka testilugude all. Joonisel 17 kuvatakse testilugude lisamine/eemaldamine testiplaani, kus kollast värvi read tähistavad seda, et antud testilood on juba testiplaani lisatud.



Joonis 17. Testilugude lisamine/eemaldamine testiplaanis.

Muude nõuete täitmine

Testihaldusvahendil puudub sisseehitatud vigade raporteerimise ja haldamise võimalus. Selleks on loodud integratsioonivõimalus erinevate veahaldusvahenditega, mille ühendamine töö autoril kahjuks ei õnnestunud. Nõuete sidumine testilooga on võimalik, kuid selleks tuleb nõuded enne nõuete haldamise lehele sisestada. Testiplaani on testide juurde lisamisel alati võimalus täiendada ning samamoodi saab testilugusid sealt alati eemaldada. Testiplaani haldamise lehel on iga testiloo juures märged, kas antud testilugu on varem käivitatud või mitte. Sarnaste testilugude grupeerimiseks on võimalus teha uued kaustad ning lisada sarnased testilood ühte kausta. Raporti koostamiseks on olemas mitmeid erinevaid valikuid. Näiteks on võimalus testitulemusi kuvada eraldi dokumendifailina kui ka graafiliselt kujutades (joonis 18).



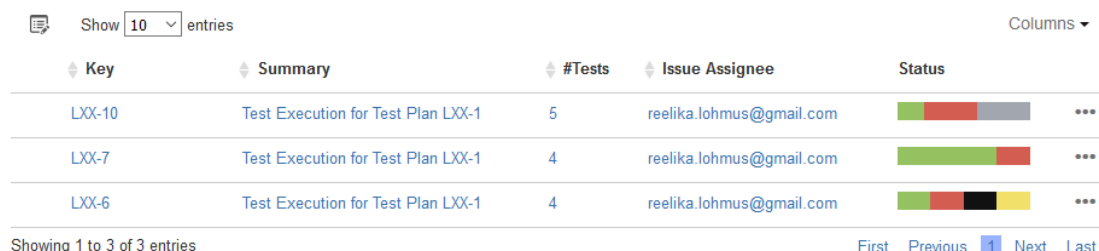
Joonis 18. Testitulemuste kuvamine gruppide kaupa.

5.5 Xray for JIRA

Kodulehe andmete põhjal selgus, et Xray vastab kõikidele nõuetele, mis töö kolmandas peatükis kirja pandud. Tal puudub küll liidestamine veahaldusvahenditega, kuid kuna JIRA enda veahaldus on väga põhjalik, siis ei ole töö autori arvates eraldi testihaldusvahendi jaoks vajadust. Ülesannete täitmise käigus selgus, et prooviversiooni kasutamise tõttu pole võimalik ühte ülesannet täita, sest prooviversioonis antud funktsionaalsus puudub. Kuna võimalusi on Xray juures palju, siis võib kasutamine esialgu keeruline tunduda, sest lisaks paljudele võimalustele tuleb ka osa seadistusest ise paika panna. Näiteks oli vaikimisi keelatud manuste lisamine, kuid manuse lisamise korral tuli selle kohta ka vastav teavitus. Antud testihaldusvahend sobib kasutamiseks nii klassikalises arendusprotsessis kui ka agiilses arendusprotsessis.

Testilugude läbimisega seotud nõuete täitmine

Testilugude loomiseks tuleb esmalt teha projekt, kuhu testilugusid panna. Sarnaselt Zephyr'ile on ka Xray kasutamisel lisaks tegevuse sammude ja oodatud tulemuste lahtritele olemas ka koht testandmete sisestamiseks. Ühte testilugu on võimalik korduvalt kasutada ning seda ka ühe testiplaani raames. Testiplaani juures tuleb testide korduvkasutamisel luua uus käivitamise protsess (*execution*). Käivitamiste ajalugu on nähtav iga testiloo all. Testitulemusi saab märkida nii sammude kaupa kui ka kogu testiloole korraga. Tulemuste märkimine on võimalik ka testiplaani väliselt. Manuste lisamise võimalus on olemas, kuid selle funktsionaalsuse kasutamiseks tuleb seadete alt see võimalus kõigepealt sisse lülitada. Järgneval joonisel x on kujutatud testide käivitamise komplektid ühe testiplaani raames ning sellega koos on nähtavad nende testide tulemused.



Key	Summary	#Tests	Issue Assignee	Status
LXX-10	Test Execution for Test Plan LXX-1	5	reelika.lohmus@gmail.com	<div style="width: 100%;"><div style="width: 20%; background-color: green;"></div><div style="width: 30%; background-color: red;"></div><div style="width: 50%; background-color: grey;"></div></div> ...
LXX-7	Test Execution for Test Plan LXX-1	4	reelika.lohmus@gmail.com	<div style="width: 100%;"><div style="width: 80%; background-color: green;"></div><div style="width: 20%; background-color: red;"></div></div> ...
LXX-6	Test Execution for Test Plan LXX-1	4	reelika.lohmus@gmail.com	<div style="width: 100%;"><div style="width: 20%; background-color: green;"></div><div style="width: 20%; background-color: red;"></div><div style="width: 20%; background-color: black;"></div><div style="width: 40%; background-color: yellow;"></div></div> ...

Showing 1 to 3 of 3 entries

First Previous **1** Next Last

Joonis 19. Testide tulemuste kuvamine ühe testiplaani raames.

Muude nõuete täitmine

Vigade raporteerimist on võimalik teha mitut moodi. Üks variant on viga kirja panna testilugudest sõltumata. Teine võimalus on aga testitulemuste märkimise juures, kus saab kirja pandud vea konkreetse testilooga siduda. Nõuete lisamise protsessi ei olnud prooviversiooni tõttu võimalik läbi proovida, kuid kodulehe andmetel on selline võimalus olemas. Nõuete põhjal on võimalik koostada ka eraldi testiraportit, samamoodi saab nõuete sidumisel testilugudega kontrollida kui suur osa nõuetest on testidega kaetud. Erinevalt Zephyr'ist on Xray eeltingimuste lisamine eraldi protsess ning need tuleb enne testilooga sidumist eraldi defineerida. Testilugude jagamine gruppidesse toimub siltide (*labels*) lisamise kaudu ehk igale testiloole saab lisada teemaga haakuva sildi ning selle nimetuse järgi on võimalik sarnaseid testilugusid hiljem otsida. Raportite loomine on seotud suures osas nõuetega ning seetõttu polnud paljude raportite koostamine võimalik. Küll aga saab koostada tabeli erinevate testide käivitamiste tulemuste kaupa, kus on täpselt näha millise staatuse ja tüübiga antud testid olid. Lisaks näitab tabel ka testide läbimise edukuse protsenti. Graafiliste tulemuste kuvamine on võimalik erinevate näitajate põhjal: testiplaan, tulemused, nõuded.

5.6 Testihaldusvahendite kasutamise analüüs

Testihaldusvahendiga qTest õnnestus edukalt läbida kõik 9 ülesannet. Kasutamiskogemuse põhjal on tegu neljast testihaldusvahendist kõige lihtsama variandiga. Peale projekti loomist sai koheselt hakata testilugusid kirja panema ning ei olnud vaja eraldi midagi seadistada. See, et testilugude käivitamine on võimalik ilma testiplaanita teeb vahendi teistega võrreldes paindlikumaks. Plussiks on veel see, et veahaldus on võimalik ka programmi siseselt ning integreerimine spetsiaalse veahaldusvahendiga pole tingimata vajalik.

Testihaldusvahendiga Zephyr for JIRA õnnestus täielikult läbida 8 ülesannet ning osaliselt 1 ülesanne. Kuigi testihaldusvahendi omaduste juures oli kirjas, et nõudeid on võimalik testilooga siduda, siis töö autor seda võimalust ei leidnud. Esmakordsele kasutajale, kes pole JIRA süsteemidega kokku puutunud, on tegemist üpris keerulise vahendiga, sest erinevaid võimalusi on vahendil palju. Iga testiloo sammu juures on Zephyr'il võimalus sisestada ka testandmed. Testilugude käivitamine on võimalik ilma testiplaani lisamiseta.

Testihaldusvahendiga TestLink õnnestus samuti läbida 8 ülesannet. Neljast testihaldusvahendist on tegemist ainsaga, kus ei ole sisseehitatud vigade raporteerimise võimalust. Integratsioonivõimalus veahaldusvahenditega on olemas, kuid kahjuks ühendamine sellega ei õnnestunud. TestLink'i kasutamine ei ole tegelikult keeruline, kuid raskeks teeb selle kasutamise korraliku kasutusjuhendi puudumine.

Testihaldusvahendi Xray'ga õnnestus läbida samuti 8 ülesannet. Kõigist neljast testihaldusvahendist on tegemist kõige keerulisema vahendiga ning selle kasutama õppimine võttis töö autoril kõige rohkem aega. Testilugude käivitamine oli võimalik ilma testiplaanita. Vigade raporteerimine ja testide käivitamine oli väga mugav. Selle vahendi kasutamise juures meeldis töö autorile kõige rohkem testide ajaloo nägemine.

5.7 Järelduste tegemine

Testihaldusvahendeid võrdlevat tabelit vaadates selgub, et kõiki nõudeid rahuldava ja lihtsasti mõistetava testihaldusvahendi leidmine on raske. Vahendeid on küll palju, kuid nende võimalused ja keerukus on väga erinevad. Testihaldusvahendite küsimustiku vastuseid uurides selgus, et kõige rohkem kasutatakse Eestis TestLink'i, kuid paljud ei ole sellega rahul. Tasuta saadaolevaid vahendeid on vähe ning tasulistesse ei taheta investeerida. Küsimustikule vastanute enda pakutud testihaldusvahendite seas oli kõige rohkem pakutud Xray'd, mistõttu antud tööriist võrdlusesse võeti.

Testitulemusi vaadates selgub, et kõikidel vahenditel on omad puudused, kuid ainsana sai kõikide ülesannete lahendamiseга hakkama qTest. Eelnevalt koostatud võrdlust, küsimustiku tulemusi ning ülesannete lahendusi arvesse võttes on kõige parem qTest testihaldusvahend. Kui Xray puhul ka nõuete haldamise osa kasutada saaks, siis sobiks hästi ka see testihaldusvahend, kuigi alguses võib selle kasutama õppimine kauem aega võtta.

6 Kokkuvõte

Käesoleva magistritöö eesmärgiks oli leida tarkvara testimisega tegelevale ettevõttele ASA Quality Services OÜ uuem ja sobivam testihaldusvahend, sest praegune programm on juba vana ning vajaks oma puuduste tõttu väljavahetamist. Eesmärgi saavutamiseks pandi kirja nõuded, millele otsitav vahend vastama peab ning see oli aluseks kõikidele järgnevatele sammudele. Peale nõuete koostamist valiti nelja välismaise testihaldusvahendeid võrdleva edetabeli põhjal välja 8 erinevat testihaldusvahendit mida võrdlema hakata. Testihaldusvahendite võrdluse tegemiseks koostati nõuete põhjal tabel kuhu märgiti kas testihaldusvahend täidab seda nõuet või ei. Lisaks sellele loodi testihaldusvahendite kasutamise kogemuse küsitlus. Küsitluse eesmärgiks oli uurida kas ja kui palju Eesti ettevõtetes testihaldusvahendeid kasutatakse.

Töö teises pooles mõeldi välja 9 erinevat ülesannet, mis olid samuti tehtud nõuete täitmise kontrolliks. Vahendid, millega testülesanded lahendati, valiti välja võrdleva tabeli ning testihaldusvahendi küsimuste vastuste põhjal. Kokku viidi testid läbi 4 erineva testihaldusvahendiga. Testihaldusvahendid, millega ülesanded läbi viidi olid:

- qTest
- Zephyr for JIRA
- TestLink 1.9.16
- Xray for JIRA

Testihaldusvahendite kasutamise tulemused ei erinenud üksteisest väga palju ning kõik testihaldusvahendid said suurema osa ülesannetega edukalt hakkama.

Magistritöö tulemusena selgus, et kõige sobivamaks testihaldusvahendiks oleks ettevõttele qTest. Kui testihaldusvahendil Xray saaks tööle ka nõuete osa, siis sobiks ka see testihaldusvahend kasutamiseks, sest kuigi tegemist oli keerulisema

testihaldusvahendiga, täitis see kõik ülejäänud nõuded. Töö autor soovib ettevõttel esmase valikuna kasutusele võtta qTesti testihaldusvahendi.

Kasutatud kirjandus

- [1] Agile Methodology. [WWW] <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban> (03.03.2017)
- [2] 7 Changes Scrum Made To The Tester's Role. [WWW] <https://www.ministryoftesting.com/2011/11/7-changes-scrum-made-to-the-tester-s-role/> (03.03.2017)
- [3] Rubin, K.S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Michigan : Pearson Education, Inc.
- [4] Kanban for QA Teams. [WWW] <https://kanbanize.com/kanban-resources/case-studies/kanban-for-qa-teams/> (06.03.2017)
- [5] Kanban. [WWW] <https://www.atlassian.com/agile/kanban> (03.03.2017)
- [6] Kanban for Software Testing Teams. [WWW] <https://dzone.com/articles/kanban-for-software-testing-teams-1> (03.03.2017)
- [7] Agile Task Management. [WWW] <http://www.continuousagile.com/unblock/kanban.html> (03.03.2017)
- [8] What is V-model- advantages, disadvantages and when to use it? [WWW] <http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/> (03.03.2017)
- [9] Black, R., Veenendaal, E.V., Graham, D. (2012). Foundations of Software Testing ISTQB Certification. 3rd ed. Hampshire : Cengage Learning EMEA.
- [10] About Test Management. [WWW] <http://www.testmanagement.com/about-test-management.html> (25.03.2017)
- [11] Unit Testing. [WWW] https://www.tutorialspoint.com/software_testing_dictionary/unit_testing.htm (26.03.2017)
- [12] What is Unite testing? [WWW] <http://istqbexamcertification.com/what-is-unit-testing/> (26.03.2017)
- [13] What is Integration testing? [WWW] <http://istqbexamcertification.com/what-is-integration-testing/> (26.03.2017)
- [14] What is System testing? [WWW] <http://istqbexamcertification.com/what-is-system-testing/> (27.03.2017)
- [15] What is Acceptance testing or User Acceptance Testing (UAT)? [WWW] <http://istqbexamcertification.com/what-is-acceptance-testing/> (27.03.2017)
- [16] The Importance of Test Management Tools. [WWW] <http://www.te52.com/testtalk/2015/06/02/importance-of-test-management-tools/> (03.03.2017)
- [17] What is Test management tools? [WWW] <http://istqbexamcertification.com/what-is-test-management-tools/> (29.03.2017)
- [18] Why is software testing necessary? [WWW] <http://istqbexamcertification.com/why-is-testing-necessary/> (30.03.2017)

- [19] What is Software Quality? [WWW] <http://istqbexamcertification.com/what-is-software-quality/> (30.03.2017)
- [20] What are Agile Software Development approaches? [WWW] <http://istqbexamcertification.com/what-are-agile-software-development-approaches-scrum-kanban-xp-explained/> (30.03.2017)
- [21] What is a kanban board? [WWW] <https://leankit.com/learn/kanban/kanban-board/> (07.04.2017)
- [22] What is V model in software testing and what are advantages and disadvantages of V model? [WWW] <http://testingfreak.com/v-model-software-testing-advantages-disadvantages-v-model/> (07.04.2017)
- [23] The International Software Testing Standard. [WWW] <http://www.softwaretestingstandard.org/> (01.05.2017)
- [24] Top 15 Best Test Management Tools of 2017. [WWW] <http://www.softwaretestinghelp.com/15-best-test-management-tools-for-software-testers/> (07.03.2017)
- [25] Top 20 Test Management Tools In 2017. [WWW] <http://www.guru99.com/top-20-test-management-tools.html> (07.03.2017)
- [26] Top 10 Test Management Tools for Smarter Testing. [WWW] <http://www.coderewind.com/2014/06/top-10-test-management-tools/> (07.03.2017)
- [27] Best Test Management Tools in 2017. [WWW] <https://geteasyqa.com/blog/best-test-management-tools/> (07.03.2017)
- [28] Comparison of Test Management Tools. [WWW] <http://blogs.perficient.com/delivery/blog/2015/08/17/comparison-of-test-management-tools/> (07.03.2017)
- [29] Transform Test Case Management. [WWW] <https://www.qasymphony.com/software-testing-tools/ctest-manager/test-case-management/> (10.03.2017)
- [30] About Xray for JIRA. [WWW] <http://confluence.xpand-addons.com/display/XRAY/About+Xray+for+JIRA> (10.03.2017)
- [31] TestLink. [WWW] <https://sourceforge.net/projects/testlink/> (10.03.2017)
- [32] QA & Test Case Management for Professional Testing. [WWW] <https://www.practitest.com/product/> (10.03.2017)
- [33] Zephyr for JIRA. [WWW] <https://www.getzephyr.com/products/zephyr-for-jira> (10.03.2017)
- [34] Take control of everything important from main dashboard. [WWW] <https://testcollab.com/tour/> (10.03.2017)
- [35] Modern test management for your team. [WWW] <http://www.gurock.com/testrail/tour/modern-test-management/> (10.03.2017)
- [36] A better way to manage tests. [WWW] <https://www.testlodge.com/tour> (10.03.2017)
- [37] Quality makes the difference! [WWW] <https://testcaselab.com/> (10.03.2017)
- [38] For teams of all sizes, in the cloud. [WWW] <https://www.atlassian.com/software/jira/pricing?tab=cloud> (04.05.2017)
- [39] Pricing. [WWW] <https://marketplace.atlassian.com/plugins/com.thed.zephyr.je/cloud/pricing> (04.05.2017)
- [40] For teams of all sizes, hosted on your server. [WWW] <https://www.atlassian.com/software/jira/pricing?tab=self-hosted> (04.05.2017)

- [41] Pricing. [WWW]
<https://marketplace.atlassian.com/plugins/com.thed.zephyr.je/server/pricing> (04.05.2017)
- [42] Pay-as-you-go pricing. [WWW] <https://www.practitest.com/pricing/> (04.05.2017)

Lisa 1 – Testihaldusvahendi küsitluse küsimused

* 1. Kui pikk on sinu kogemus tarkvara testijana?

- 0-2 aastat
- 3-5 aastat
- 6-10 aastat
- Rohkem kui 10 aastat

2. Kui kaua oled töötanud testijuhina?

- 0-2 aastat
- 3-5 aastat
- 6-10 aastat
- Rohkem kui 10 aastat
- Ei olegi

* 3. Olen olnud..

- Arendusmeeskonna testija
- Äritestija
- Vastuvõtutestija
- Other (please specify)

* 4. Milliseid testihaldusvahendeid oled oma töös kasutanud?

- qTest
- PractiTest
- Zephyr
- TestCollab
- TestRail
- TestLodge
- TestLink
- TestCaseLab
- Other (please specify)

Joonis 20. Testihaldusvahendi küsitluse küsimused.

* 5. Millised funktsionaalsused sinu arvates ühel testihaldusvahendil kindlasti olema peavad?

- Testilugude grupeerimine/jagamine gruppidesse
- Lisainformatsiooni/failide lisamine
- Testiraporti koostamine
- IT või ärinõuete testimine
- Rollihaldus
- Vigade raporteerimise võimalus
- Testitulemuste märkimise võimalus
- Other (please specify)

* 6. Kui tihti kasutad järgnevaid testihaldusvahendi funktsionaalsusi oma igapäevatöös?

	Mitte kunagi	Harva (1-2x aastas)	Vahepeal (1-2x kuus)	Igapäevaselt
Testilugude grupeerimine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lisainformatsiooni/failide lisamine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testiraporti koostamine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
IT või ärinõuete lisamine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testitulemuste märkimine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

* 7. Märgi kuni 3 testihaldusvahendit mida soovitaksid teistelegi?

- qTest
- PractiTest
- Zephyr
- TestCollab
- TestRail
- TestLodge
- TestLink
- TestCaseLab
- Other (please specify)

Joonis 21. Testihaldusvahendi küsitluse küsimused.

* 8. Vali testihaldusvahend, mille kasutamisega sa rahule ei jäänud?

- qTest
- PractiTest
- Zephyr
- TestCollab
- TestRail
- TestLodge
- TestLink
- TestCaseLab
- Other (please specify)

* 9. Millised on sinu kogemused testihaldusvahenditega agiilses arenduses? Milliseid vahendeid oled kasutanud ning kuidas need agiilsesse arendusprotsessi sobinud on?

Joonis 22. Testihaldusvahendi küstiluse küsimused.