

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Hanna Liivia Teller 193897 IABB

Webmasterite töö testimise ja kvaliteedi tagamise rakenduse prototüüp

Bakalaureusetöö

Juhendaja: Jelena Vendelin

PhD

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: Hanna Liivia Teller

30.12.2022

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on sisuhaldusprojekti tehtavate vigade määra vähendamine. Selle saavutamiseks kavandati veebirakenduse prototüüp, mis kontrollib veebilehte. Peamiseks vaadeldavaks probleemiks on veebilehe halduse käigus tekkinud liigne vigade arv, mis jõuab kliendini. Eesmärk on viia kliendini jõudvate vigade määr 5% peale ja tiimisiseste vigade määr 15% peale.

Vigade minimeerimiseks koostatakse *checklist*, kus on kirjas vastavalt veebilehe kategoriale list punktidest, mida kontrollida veebilehel. *Checklist* on valideeritud koos kliendiga. *Checklisti* eesmärk on aidata *webmasteritel* meeles hoida kõike vajalikku veebilehte kontrollides. Lisaks antakse töös teoreetiline ülevaade testimisest ja prototüübi loomise headest tavadest.

Bakalaureusetöö tulemina valmib *checklisti* põhjal veebilehe vigade kontrollimiseks kavandatud prototüüp, mis aitab *webmasteritel* oma tööd automaatselt kontrollida. See aitab omakorda vähendada tiimisiseste ja kliendini jõudvate vigade arvu. Kogu protsessi analüüsiti vigade aruande põhjal. Töös püstitatud eesmärgid said täidetud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 3 peatükki, 8 joonist, 6 tabelit.

Abstract

Prototype application to test the work of webmasters and ensure quality

This bachelor's thesis aimed to reduce the rate of errors made in the content management project. To achieve this, a web application prototype was designed to check the components on the web page. The main problem was that there were too many errors reaching the client. The acceptable error rate agreed upon with the client remained above 5%. The rate of errors within the team remained at the agreed 15%.

To achieve the goals, the content management project was prepared with an analysis of the AS-IS and TO-BE processes. To minimize errors, a checklist was prepared containing a list of points to be checked on the website according to the category of the website. The checklist was validated with the client. The purpose of the checklist is to help webmasters remember everything necessary when checking a website. In addition, the thesis provides a theoretical overview of testing and good prototyping practices.

As a result of the bachelor's thesis, a prototype designed for checking website errors was created based on the checklist, which helps webmasters check their work automatically. This, in turn, helps reduce team errors and reach the customer. The entire process was analyzed based on the results of a weekly error report. The results of the error report revealed that no more than 5% of errors from all completed works reached the customer per week. Mistakes made within the team during the 12 weeks were 10,1% for Lexus and 14,6% for Toyota, but there were weeks where the error rate was higher. Based on the results, it can be concluded that only using a checklist is not enough to reduce errors within the team and it is necessary to automate the checking. It is possible to lower the rate of errors within the team even further. The further development of the finished work includes the development process itself, where the finished prototype is realized.

The thesis is in Estonian and contains 29 pages of text, 3 chapters, 8 figures, 6 tables.

Lühendite ja mõistete sõnastik

<i>Backlog</i>	<i>Backlogi</i> kogutakse kõik kliendi ja meeskonna poolt välja pakutud tööd, mida tulevikus teostada. [1]
Figma	Figma on vektorgraafika redigeerija ja tööriist, mida kasutatakse prototüüpimiseks. [2]
Jira	Jira on arendusprojektide haldamise tarkvara. [3]
<i>Kanban board</i>	Kanban tahvel on paindlik projektihaldustööriist, mis on loodud töö visualiseerimiseks. [1]
QA	Kvaliteedi tagamine (<i>quality assurance</i>) on protsess, mille käigus tehakse kindlaks, kas toode või teenus vastab kindlaksmääratud nõuetele. [4]
SEO	<i>Search engine optimization</i> . SEO olemasolul on võimalik veebisaidi nähtavust tõsta otsingumootorites nagu Google, Bing jt. [5]
<i>Stand-up</i>	Püstikohtumine on igapäevaselt korraldatav kindla struktuuriga lühike koosolek. [6]
<i>Webmaster</i>	<i>Webmaster</i> ehk veebimeister loob kasutades sisuhaldusplatovormi uusi lehti või redigeerib olemasolevaid.
UAT	Kasutajate aktsepteerimise testimine, testimise viivad läbi lõppkasutajad. [7]

Sisukord

1 Sissejuhatus	9
1.1 Taust	9
1.2 Probleem	10
1.3 Eesmärk	11
1.4 Töö struktuur	11
2 Metoodika	12
2.1 Ettevõtte lühitutvustus	12
2.2 Tööprotsessi lühitutvustus	12
2.3 Sisuhaldusplatvormi tutvustus	14
2.4 Jira protsessid	14
2.5 Manuaaltestimise plussid ja miinused	16
2.6 Teavitustöö ja iganädalane vigade aruanne	18
2.7 QA <i>checklist</i>	19
3 Veebilehe vigade kontrollimise prototüüpi loomine	20
3.1 Prototüübi ja <i>wireframe</i> definitsioon ja kasutusosalad	20
3.2 Prototüüpimise võimalused	21
3.3 Nõuded prototüübile	22
3.4 Prototüübi eesmärk ja funktsionaalsused	22
4 Tulemuste analüüs	25
4.1 Vigade aruande tulemused ja analüüs	25
4.2 Tööde topeltkontroll	29
4.3 Jira protsesside parendused	30
4.4 <i>Checklisti</i> kasutuselevõtu analüüs	31
4.4.1 <i>Checklisti</i> kasutuselevõtu tulemused praktikas	32
4.4.2 Nelja <i>ticketi</i> ülesandepüstitus ja vigade leidmise arv lehelt	33
4.5 Prototüübi edasiarendused ja kasutusvaldkonnad	36
5 Kokkuvõte	38
Kasutatud kirjandus	39

Jooniste loetelu

Joonis 1. Sisuhaldusprojekti tööprotsess.	14
Joonis 2. Jira <i>Kanban board</i>	15
Joonis 3. Tööprotsessi AS IS joonis.	16
Joonis 4. Tööprotsessi TO BE joonis.	16
Joonis 5. Veebilehe kontrollimise prototüübi algvaade.	23
Joonis 6. Vaade prototüüpist peale kontrolli teostamist.	24
Joonis 7. Veebilehe kontrollimiseks loodud prototüüp täisvaates.	24
Joonis 8. Ekraanipilt Lexuse pistikhübriidi lehelt, kus tekst ei mahu komponenti.	33

Tabelite loetelu

Tabel 1. Lexuse sisuhaldusprojekti vigade aruanne.	27
Tabel 2. Toyota sisuhaldusprojekti vigade aruanne.	28
Tabel 3. <i>Checklisti</i> kasutuselevõtu tulemused esimese <i>ticketi</i> puhul.	33
Tabel 4. <i>Checklisti</i> kasutuselevõtu tulemused teise <i>ticketi</i> puhul.	34
Tabel 5. <i>Checklisti</i> kasutuselevõtu tulemused kolmanda <i>ticketi</i> puhul.	34
Tabel 6. <i>Checklisti</i> kasutuselevõtu tulemused kolmanda <i>ticketi</i> puhul.	35

1 Sissejuhatus

Järjest suuremal määral liigub ettevõtete teenuste ja toodete müük veebikeskkonda. See tähendab omakorda, et ettevõtte veebileht peab pakkuma järjest suuremat funktsionaalsust, mis aitab kliendil oma ostuotsust teha. Selle kõige jaoks on oluline omada terviklikku tarkvara, milles ei esine vigu. 2022. aasta Ühendkuningriigi e-kaubanduse aruande põhjal on veebilehel esinevate probleemide tõttu kaotanud ettevõtted viimase kahe aasta jooksul 10% oma tuludest. Lisaks näitab Amazon Web Services'i statistika, et 88% kasutajatest ei pöördu pärast halba kasutajakogemust veebisaidile tagasi. [8]

Vigade vältimiseks on tarvis tarkvara testida. Üheks testimise võimaluseks on manuaaltestimine, mida ka bakalaureusetöös lähemalt vaatame. Manuaaltestimist teostatakse käsitsi. Selle käigus on võimalik leida nähtavaid ja peidetud probleeme tarkvaras. [9]

1.1 Taust

Bakalaureusetöö autor töötab alates 2020. aasta oktoobrikuust ettevõttes ADM Interactive. Ettevõtte põhitegevusala on digiturundus. Autor alustas ettevõttes *webmasteri* rollis, kuid alates 2022. aastast on ta Lexuse Baltikumi veebi projektijuht. Autori tööülesannete hulka kuulub Lexuse veebilehtede sisu ja toimimise eest vastutamine. Lisaks sellele tegeleb lõputöö autor ettevõttes ka Toyota projekti uudiskirjade koostamise ning *Online Service Bookingu* haldamisega.

Käesolevas projektis vaatleme ADM Interactive ühte suurimat klienti Toyota Baltic AS, kes on autotööstuse üks suurtootjaid. Ettevõtte kaubamärkide alla kuuluvad nii Toyota kui ka Lexuse autobrändid. ADM Interactive pakub kliendile sisuhaldusteenust, mis sisaldab uute veebilehtede tegemist, olemasolevate muutmist ning parema funktsionaalsuse loomist.

Antud Toyota ja Lexuse sisuhaldusprojektis on kasutusel agiilsed arenduspõhimõtted. Agiilne arendus tähendab pidevat muutmist ja kohanemist vastava keskkonna muutustele. Agiilsete arendusmeetoditega saame muutusi kiiresti luua ja neile reageerida, saavutades edu ka kõige muutuvamas keskkonnas. [10]

Lisaks agiilsetele printsiipidele on võetud probleemi lahendamiseks kasutusele ka PDCA (*Plan-Do-Check-Act*) tsükkel.

Tsükli neli etappi on:

- *plan* – praeguse olukorra ja probleemi kirjeldus, soovitud tulemuse välja töötamine;
- *do* – lahenduse testimine, kõige pealt testitakse lahendust ühe tiimiga ja kui see toimib võetakse lahendus kasutusele laiemalt;
- *check* – tulemuste kontrollimise ja analüüsimise faas, kus vaadatakse kas midagi on läinud protsessis paremaks. Kui ei ole, minnakse tagasi planeerimisefaasi;
- *act* – kui lahendus oli edukas hakatakse seda laialdasemalt kasutama (näiteks võetakse kasutusele see ka teistes organisatsiooni tiimides/harudes), kui lahendus ei olnud edukas liigutakse tagasi algusfaasi. [11]

Vaadeldavasse projekti kuulub kümme veebilehte, viis lehte brändi kohta. Nendeks on: Eesti, Eesti-Vene, Läti, Läti-Vene ja Leedu. Kasutatav sisuhaldusplatvorm on Adobe Experience Manager. [12]

1.2 Probleem

2022 aasta suve lõpus hakkas ADMi Toyota ja Lexuse tiim vigade tekkimisele tähelepanu pöörama. Saadi aru, et viimasel ajal on vigu liiga palju tekkinud ja ka kliendi poolt tulid esimesed kommentaarid, et antud teema tuleks fookusesse võtta. Peamised vead tekkisid *webmasterite* töös, kus kliendi poolt soovitud ülesanded saadeti vigaselt tagasi.

Selle aasta augustikuu põhjal oli Toyotal 6% vigu ja Lexusel 15% vigu, mis jõudsid kõikidest töödest kliendini. Vigadeks olid näiteks: katkised pildid, vale tekst, katkine komponent lehel või vale URL'i aadress. Kliendi poolne tunnetus antud olukorrale oli, et Toyotal on 5% ja Lexusel 50% vigu kõikide tööde peale kokku. Vead, mis jõudsid *webmasterilt* projektijuhini oli ligikaudu 30% mõlema brändi puhul.

Vigadeks loetakse ühes Jira *ticketis* tehtud vigu. *Ticket* on Jira töö, mida tuleb tegeleda. *Ticket* on tavaliselt 1 lehekülg ja kõik vead, mis ühes *ticketis* tehakse võrduvad ühe veaga. [13]

1.3 Eesmärk

Lõputöö eesmärk on viia sisuhaldusprojekti tehtavate vigade arv, mis jõuab kliendini 5% peale. Antud protsendimäär on kliendiga kokkulepitud ning kliendi poolt aktsepteeritava määr. Lisaks sellele on eesmärk viia tiimisiseste aktsepteeritavate vigade määr 15% peale. Selle saavutamiseks on koostanud kolm meetet.

1. Teavitustöö tegemine, miks teema on oluline, millist finantsilist kahju võivad vead veebilehel kliendile põhjustada. Tuua välja head testimise praktikad, millele tugineda.
2. *Webmasterid* võtavad kasutusele QA (*quality assurance*) *checklisti* töö tegemiseks.
3. Veebirakenduse prototüübi kavandamine, mis kontrollib *checklistis* olevaid komponente.

1.4 Töö struktuur

Autor kasutab bakalaureusetöös kvalitatiivse analüüsi meetodikat, kus uuritakse saadud tulemusi ja pakutakse välja uusi lahendusi. Töö koosneb sissejuhatusest ja kolmest suuremast peatükist.

Esimene peatükk tutvustab lühidalt ettevõtet, tööprotsesse, sisuhaldusplatvormi, Jira protsesse, vigade aruannet ja vigade vähendamiseks loodavat *checklisti*. Lisaks tutvustatakse manuaaltestimise plusse ja miinuseid.

Teises peatükis vaadeldakse autori poolt loodud QA rakenduse prototüüpi. Samuti uuritakse, mis on olulised aspektid prototüübi loomisel ning mida tuleb seda tehes jälgida.

Kolmandas peatükis analüüsitakse kõiki lõputöös saadud tulemusi. Nendeks on Jira protsesside parendamine, topeltkontrolli meetod, vigade aruanne ja *checklisti* kasutusele võtt.

2 Metoodika

Käesolevas peatükis käsitletakse kõiki bakalaureusetöös kasutatavaid metoodikaid. Sealhulgas tutvustatakse ettevõtet, vaadeldavat projekti ja selle tööprotsessi, projektis kasutatavat sisuhaldusplatvormi, Jira protsesse, vigade aruannet ja vigade vähendamiseks loodavat *checklisti*.

2.1 Ettevõtte lühitutvustus

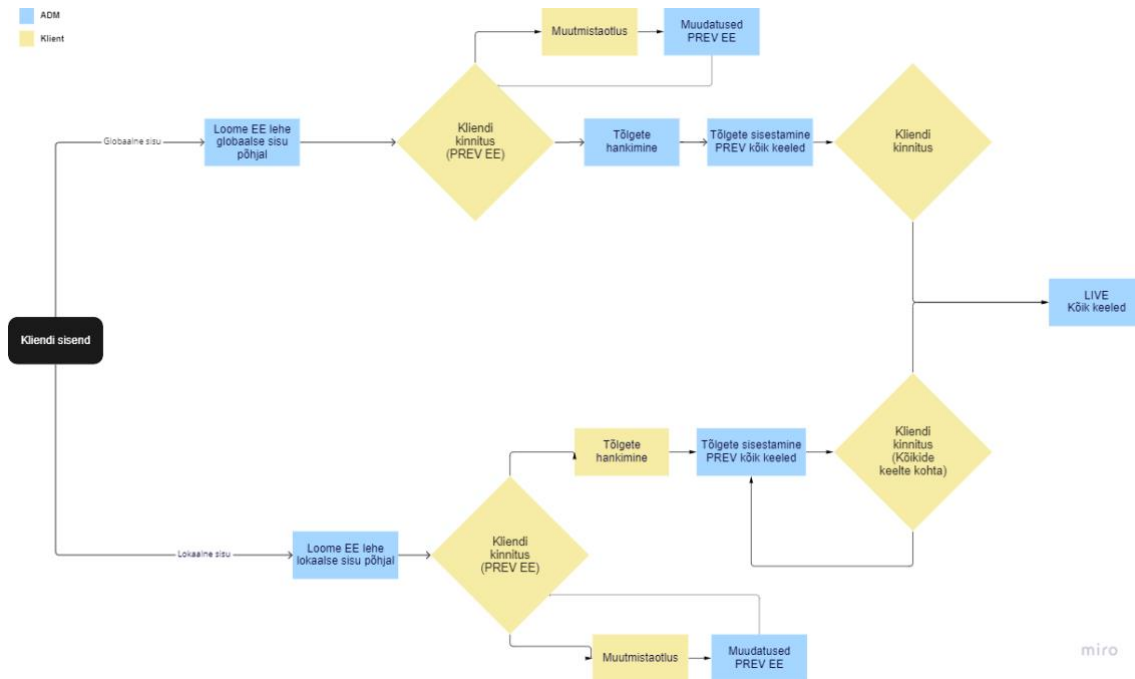
ADM Interactive on digitaalse teeninduse, müügi- ja turunduslahendusi loov digiagentuur. Ettevõtte tegutseb aastast 1997. Praegu töötab Eestis 100 + töötajat, kuid kogu ADM gruppi kuulub üle 215 töötaja. ADM gruppi kuulub järgmised brändid: ADM unit X, ADM interactive, ADM integrations, ADM cloudtech, ADM media, ADM workflows ja Elevator Startups. Ettevõtte tegutseb erasektoris, mille organisatsioonitüübiks on divisionaalne organisatsiooni struktuur. ADM Interactive peakontor asub Tallinnas Kultuurikatlas (Põhja pst 27a). Lisaks on kontorid veel Londonis (20 Birchin Lane) ja Berliinis (Schulstr. 13). Tegemist on keskmise suurusega ettevõttega, kogu gruppi peale oli 2021 aastal käive ligikaudu 20 miljonit. Ettevõtte klientideks on näiteks Eesti suurimad pangad, poeketid ja teised brändid. [14]

2.2 Tööprotsessi lühitutvustus

Käesolevates projektides annab projektijuht *webmasteritele* tööülesannete juhised Jira platvormis. Tööülesanneteks võib olla näiteks uute funktsionaalsuste loomine, kasutades HTML/CSS koodi kirjutamist või sisuhaldusplatvormis lehtede loomine ja muutmine. Projektijuhi ülesannete hulka kuulub kliendiga suhtlemine, tähtaegade paika panemine ja nendest kinnipidamine, kliendile tööde edastamine ja tiimi tööprotsesside toimimise eest vastutamine.

Järgnevalt antakse ülevaade sisuhaldusprojekti tööprotsessist.

1. Klient annab projektijuhile sisendi, kas emaili teel või suuliselt. Kui tegemist on uue lehega (jaguneb lokaalseks ja globaalseks sisuks), siis on tööprotsess erinev (vt Joonis 1).
2. Projektijuht koostab Jiras töö kohta pileti (*ticket*), milles kirjeldab tööülesande ja tähtaja.
3. *Webmaster* alustab enda tööülesannete täitmist.
4. Kui *webmaster* lõpetab enda poolsed ülesanded, kirjutab ta *ticket*isse kommentaari, mis muudatusi ta läbi viis ja suunab töö tagasi projektijuhile, kes töö üle kontrollib.
5. Projektijuht vaatab töö üle ning kui töös esinevad vead, suunab ta Jiras töö tagasi *webmaster*ile ja lisab kommentaarid, mida tuleb parandada.
6. Kui töö on valmis, edastab projektijuht töö kliendile. Kui tegemist on uue lehega, saab klient eelvaate eesti keelsest lehest, mille ta üle vaatab. Kui kõik sobib ja täiendavaid parandusi pole tarvis teha, tehakse eesti keele põhjal ka teised keeled.
7. Klient saab projektijuhilt valmis töö. Joonis 1 kirjeldab täpsemalt kogu tööprotsessi algusest lõpuni, millised on vaheetapid ja seda, kus toimub lehe ülevaatus kliendi poolt.



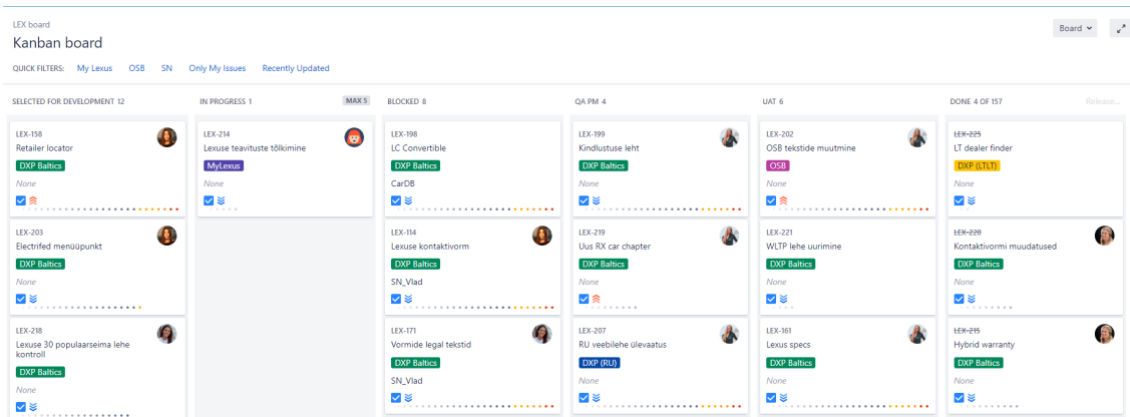
Joonis 1. Sisuhaldusprojekti tööprotsess.

2.3 Sisuhaldusplatvormi tutvustus

Adobe Experience Manager, lühendiga AEM on sisu- ja digitaalvara haldusrakendus, mis võimaldab luua, hallata ja edastada sisu igas digitaalses puutepunktis. See aitab sisuhaldusprotsessi muuta sujuvamaks ja lihtsamaks. AEMiga on võimalik hallata dokumente, PDF faile, pilte, videoid, veebivorme ja palju muud. [12]

2.4 Jira protsessid

Ettevõtte kasutab oma tööde haldamiseks Jira platvormi, läbi mille hallatakse kõikide projektide tööülesandeid. Lisaks kasutatakse osaliselt Scrumi protsessiraamistikku ja *Kanban boardi*. *Kanban boardiga* saab projektijuht prioritseerida *webmasteri* jaoks tähtsuse järjekorras erinevad ülesanded.



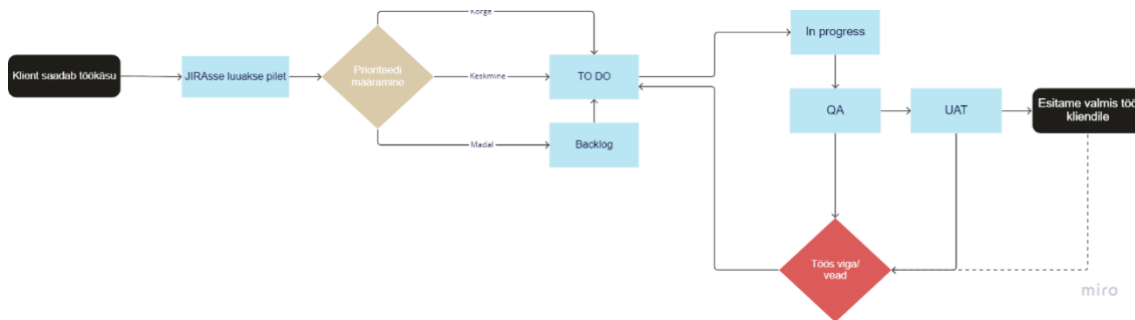
Joonis 2. Jira Kanban board.

- *Product backlog* on Scrumi üks osa. See on list töödest, mis soovitakse veebilehe täiustamiseks teha. *Backlogis* on tööd, mida soovitakse teha tulevikus või siis kui jääb aega üle *Selected for development* tööde kõrvalt.
- *Selected for developmentis* on tööd, mida soovitakse teha kindla ajaperioodi sees. Selles tulbas olevate *ticketite* tähtaeg on varasem kui *backlogis* olevatel töodel. Tööd, mis lähevad *Selected for development* tulpa, otsustab klient.
- *In progress* tulp tähistab praegu *webmasterite* poolt tehtavaid töid.
- QA tulbas on tööd, mida projektijuht kontrollib. Tema roll on kontrollida kõiki ülesande püstituses läbivaid punkte ning vaadata üle kas kõik on õigesti tehtud.
- UAT tulpa jõuavad tööd, mis on projektijuhi poolt kontrollitud ja lähevad kliendile edastamiseks.
- *Done* tulbas on kõik tehtud tööd.

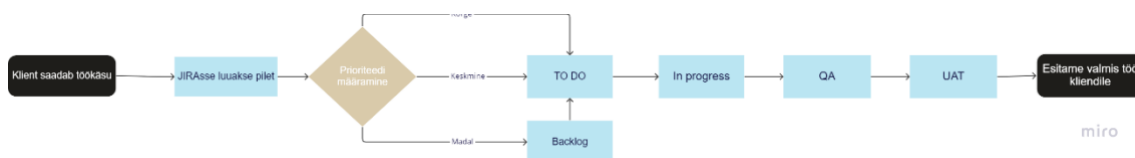
Kanban boardi tööde liikumine:

AS-IS: *Backlog* → *Selected for development* → *In progress* → QA → *Selected for development* → *In progress* → QA → UAT → Done (vt Joonis 3);

TO-BE: *Backlog* → *Selected for development* → *In progress* → QA → UAT → Done (vt Joonis 4).



Joonis 3. Tööprotsessi AS IS joonis.



Joonis 4. Tööprotsessi TO BE joonis.

Eesmärk on saavutada sisuhaldusprojekti Jira liikumine TO BE (vt Joonis 4). Selle saavutamise hoiaks aega kokku, mis kulub iga *ticketi* peale, sest edasi-tagasi liikumine jääks ära.

2.5 Manuaaltestimise plussid ja miinused

Tarkvara testimise eesmärk on enne lõppkasutajale tarkvara andmist testida kõik funktsionaalsused ja võimalused läbi. See aitab veenduda, et tarkvaras ei esineks vigu ja, et see vastaks nõuetele. Testimise tüübid jagunevad kaheks: manuaaltestimine ja automaattestimine. Lähemalt hakkame vaatama manuaaltestimist.

Manuaaltestimise eesmärk on kontrollida rakendust, veebilehte, näiteks klõpsata läbi kõik väljad, lisades sinna mingeid väärtuseid ja kontrollida, kas kõik töötab plaanipäraselt. Kui testija leiab vea, dokumenteerib ta selle ja annab informatsiooni edasi tarkvaraarendajale, kes saab juba vea eemaldada.

Testimist saab jagada mustaks ja valgeks. Kui teostatakse testimist kasutaja vaatenurgast ehk valdavalt kasutades programmi kasutajaliidest, siis on tegemist musta kasti testimisega. Sellisel juhul puudub testijal ülevaade programmi sisust ja struktuurist. Musta kasti testimist saab teostada ilma programmeerimise või koodi lugemise oskuseta.

Valget testimist ehk klaaskasti testimist nimetatakse vastupidi aga testimiseks, kus testija tegeleb programmeerimise või vähemalt programmi koodi lugemisega, et kindlaks teha millised reeglid, piirangud ja tegevused on koodis kirjas. Valge testimine toimub süsteemi looja vaatenurgast. Testimise protsessi on võimalik alustada enne kui on olemas kasutajaliides ehk valget testimist on võimalik alustada juba arendusprotsessis. [15]

Järgnevalt on loetelu testimise liigid sihtmärkidest.

- Uurimuslik testimine, tähendab seda, et testijal on konkreetne kava kuidas ta programmi kontrollib. Mida keerulisemad on testjuhtumid, seda tõenäolisem on leida vigu.
- *Ad-hoc* testimine on improvisatsioon, spontaanne vigade otsimine, mida tehakse ilma planeerimise ja dokumentatsioonita ning seda saab rakendada nii teadus- kui ka äriprojektide jaoks.
- Kasutatavuse testimine, selle käigus kontrollitakse toodete kasutajakogemust. Selle testimine on oluline nägemaks, kuidas lõppkasutaja rakendust kasutaks ning milline on programmi kasutajamugavus.
- Kasutajate aktsepteerimise testimine (UAT), testimise viivad läbi lõppkasutajad. Nad teevad otsuse programmi või rakenduse valmisoleku kohta ja annavad oma hinnangu, kas see on valmis turule toomiseks. See on ühtlasi ka testimise viimane etapp.

Manuaaltestimise kasutamise plussid on:

- Lühiajaliselt kuluefektiivsem, inimesi on lihtsam ja odavam palgata manuaaltestjaks, sest see ei nõua suuri oskusi ja varasemat pikaajalist töökogemust.
- Mitmekülgne ja paindlik, inimene võib testida mitmekülgsemalt kui arvuti.
- Lõppkasutajate tagasiside on vajadusel kättesaadav. Kasutaja annab oma hinnangu, milline oli tema kogemus programmi kasutamisel. Juhul kui kasutaja on rahul, võetakse programm kasutusele, kui mitte, läheb programm veel muutmisele.
- Ei nõua lisainvesteeringuid arendusse.

Manuaaltestimise kasutamise miinused on:

- Inimliku vea tekke võimalus. Suurte tarkvarade testimiste puhul võib jääda sisse märkamata vigu, mida arvuti oleks võinud märganud.
- Suurte tarkvarade testimisel ei ole manuaaltestimine kuluefektiivne ega korduvkasutatav.
- Teste pole võimalik täielikult uuesti korrata samm-sammult.
- Võib võtta kauem aega (eelkõige suurte tarkvarade testimise puhul). [9], [7]

2.6 Teavitustöö ja iganädalane vigade aruanne

Järgnevalt soovib lõputöö autor katsetada iganädalast vigade raporteerimist ja antud teema olulisuse tõstmist meeskonnas. Esimene samm, millega autor soovis vigade vähendamiseks alustada, oli teavitustöö. Selle eesmärk on anda tiimile mõista, millist finantsilist kahju võib vigade esinemine põhjustada ettevõtte brändile tervikuna. Näiteks, valed hinnad veebilehel, võivad ettevõttele põhjustada suurt finantsilist kahju. Mingil määral on vigade tekkimine loomulik, kuid kui neid hakkab tekkima palju, võib see kliendis tekitada usaldamatust.

Teema olulisusest tuleb pidevalt rääkida ja meelde tuletada, miks on oluline jälgida oma töö kvaliteeti. Sellise arutelu käigus on võimalik *webmasteritel* hakata mõtlema, et mis on need tüüpilised vead, mida nad teevad ja miks need juhtuvad. Tavaliselt võib põhjus olla ebapiisavas ajas, ebaselges ülesandepüstituses või milleski muus. Teemast rääkimise mõju on keeruline hinnata, sest me ei tea täpselt, mis ühte või teist viga võis põhjustada. See tähendab, et on keeruline hinnata, kas teema olulisusest rääkimine on üheks põhjuseks, miks veamäär langeb või mitte.

Iganädalaselt vaatab projektijuht üle kõik eelmisel nädalal tehtud tööd ja analüüsib, mitmes töös esines vigu (vead, mis jõudsid projektijuhini või kliendini). Samuti proovivad projektijuhid leida vigades ühisosasid ja vigu, mis olid oma loomult sarnased. Igal teisipäeval *stand-up* kõnes võetakse ette eelmise nädala raport, kus vaadatakse ühiselt tulemusi ja eesmärgis püsimist. Lisaks räägitakse tehtud vigadest, kas tehtud vead on kuidagi sarnased ning kas projektijuhid saavad omalt poolt olla *webmasteritele* rohkem toeks, et vigu ei tekiks. Näiteks, kui vea põhjuseks oli kiirustamine ajapuudusest

tingituna, siis saaks projektijuht mõelda, kuidas töid planeerida selliselt, et ajasurvet ei peaks tekkima.

2.7 QA checklist

2022. aastal migreeriti Toyota ja Lexuse veebileht üle SDL Tridioni versioonist 13 [16] AEMi. Migreerimist viis läbi kaks agentuuri ja selle käigus sai kogu vana veebilehe sisu toodud üle uuele sisuhaldusplatvormile. Kuna migreerimise käigus oli arvestatav ajasurve, ei olnud selle käigus palju aega testimiseks ja seetõttu võib tagantjärele esineda veebilehel „vanadest aegadest“ vigu.

Vigade vähendamiseks lõi autor *checklisti*, kus on kirjas kõik punktid, mida valmis töö puhul peaks kontrollima. Alustades auto mudelilehtedega, lõpetades artiklilehtedega. *Checklisti* eesmärk on, et *webmaster* võtab peale oma töö lõpetamist ette *checklisti* ning käib läbi kõik seal olevad olulised punktid. Selline kontroll aitaks ennetada suuremaid ja väikesemaid vigu, mis võivad töö käigus tekkida. *Checklist* aitab meeles pidada seda, et on oluline kontrollida veebilehte tervikuna.

3 Veebilehe vigade kontrollimise prototüüpi loomine

Autor loob Figma platvormi kasutades automaatseks vigade kontrollimiseks prototüübi *checklisti* põhjal. Selle abil on võimalik *webmasteril* kontrollida kas veebilehel olevad komponendid on töökorras. Järgnevates alampeatükides kirjeldab autor, mis on prototüüp ja paneb paika prototüübi nõuded ja funktsionaalsused. Figma on vektorgraafika redigeerija ja tööriist, mida kasutatakse prototüüpimiseks [2].

3.1 Prototüübi ja *wireframe* definitsioon ja kasutusala

Prototüüpi saab defineerida väga mitmel erineval viisil. Nii nagu nimigi ütleb, tähendab see millegi algversiooni, algmudelit või esialgset teostust [17].

Prototüübiks võib veel nimetada:

- lihtsat *šketsti*, visandeid, *wireframei*, millele võib olla lisatud funktsioonid. Näiteks saab kasutaja näpuvajutusega liikuda ühelt vaatele teisele, mis annab reaalse rakendamise kasutamise kogemuse;
- täielikult interaktiivne teostus, kus kasutatakse stiili, disaini ning mida rakendatakse hiljem ka programmi puhul, lisaks on sellisel prototüübil ka erinevad funktsionaalsused. [18]

Wireframe kujutab ette visuaalselt kasutajaliidest enne selle juurutamist ja kasutusele võttu, kuid see ei pea kujutama ette visuaalse kujunduse lõpptulemust. Lihtsalt öeldes, enne maja ehitamist vajame detailplaani lõpptulemusest ning *wireframe* annab meile ettekujutuse, milline näeb programm/rakendus välja kui see on valmis.

Wireframe loomiseks on meil tarvis viite elementi.

1. Struktuur: selles faasis pannakse paika kasutajaliidese struktuur, komponentide asukoht ning see, kuidas luuakse üks tervik.
2. Sisu: luuakse list komponentidest, elementidest ja tekstides, mida kasutatakse toote disainiks.

3. Teave: selles faasis vaadatakse üle kõik kasutajatele kuvatav info, teave ja proovitakse seda teha parimal võimalikul viisil.
 4. Funktsionaalsus: vaadatakse ja kontrollitakse üle, kuidas kasutajaliides töötab ning kuidas komponendid omavahel suhtlevad.
 5. Käitumine: selles osas kontrollitakse, kuidas kasutajaliides ja kasutajad käituvad.
- [19]

Lähemalt hakkame vaatama veebisaitide prototüüpe ja seda, milleks neid kasutatakse.

- Prototüüpi luuakse aja ja raha säästmiks. Prototüüpi tehes on võimalik seda testida mitmete kasutajate peal, mille tulemusena võivad tulla esile esimesed murekohad, mis vajaksid parandamist. Kui alustada kohe teostusega (tehes rakendus valmis), siis võib olla hiljem tunduvalt kulukam ja ajamahukam vigu parandada.
- Tuues kasutajad disainiprotsessi on võimalus näha, kuidas näeb lõppkasutaja loodud rakendust/programmi. Kasutaja saab tagasisidestada oma kasutajakogemuse. Prototüübi puhul jääb detailsete funktsionaalsuste testimise võimekus küll ära, kuid sellegi poolest on võimalik anda kliendile võimalikult lähedane kasutajakogemus antud rakenduse puhul. Vastavalt kliendi tagasisidele on võimalik teha prototüübis parendusi, muuta nuppude ja tekstide asukohta jm.
- Disainiprotsessi on võimalik kaasata asjatundjaid, kes tunnevad prototüübitava rakenduse valdkonda ja kelle tagasiside prototüübile võib olla kasulik.
- Disain erinevates seadmetes ja reaalse sisu ja andmetega prototüübi testimine. See annab hea ülevaate, milline näeb rakendus välja erinevates seadmetes. [18]

3.2 Prototüüpimise võimalused

Prototüübi loomiseks on disaineritele saadaval palju erinevaid tööriistu. Alustades staatiliste kuni täielikult interaktiivsete disainitööriistadeni. Enamike prototüübi loomise tööriistadega on võimalik luua algselt *wireframe* ja hiljem prototüüp. Nii prototüübi kui ka *wireframe* loomiseks saab kasutada tööriistu nimedega: InVision, UXPin ja Axure.

[19]

Veebilehe vigade kontrollimise prototüübi loomiseks hakkab autor kasutama Figmat. Tegemist on tasuta tööriistaga, mis on saadaval nii internetis kui ka rakenduse kujul. Lisaks on selle tööriista abil võimalik teha reaalsajas tööd mitmel disaineril korraga. [20, 17]

Figma on disainerite seas väga populaarne töövahend. 2020.aasta uuringus, kus osales neli tuhat disainerit, valis 57% vastanutest, et Figma on nende lemmik disainitööriist. Figma kasutajaliides on lihtne ja sobilik algajale disainerile. [2]

3.3 Nõuded prototüübile

Lõputöö eesmärk on viia sisuhaldusprojekti tehtavate vigade arv, mis jõuab kliendini 5% peale ning tiimisest aktsepteeritavate vigade määr 15% peale. Prototüüp peab lõputöö suurt eesmärki toetama. See annab *webmaster*ile tööriista, millega nad saavad oma tööd kontrollida ning mis omakorda peaks vigade arvu vähendama. Peamised vead, mis *webmaster*ite töös tekivad on hooletusvead. Hooletusvigadeks on: kirjavead tekstides, katkised videod või pildid veebilehel, valed hinnad või auto spetsifikatsioonid (tekkinud valesti kopeeritud algfailist/algandmetest), komponendi element on veebilehelt puudu või katki. Lisaks võib olla probleemiks sama veebilehe ülesehitus erinevates keeltes. Näiteks, kui on tarvis muuta pildigaleriid kõikides keeltes, on mõnes keeles galeriid muudetud probleemset (pilte pole lisatud või neid ei kuva). Eelmainitud probleemide vähendamiseks loob autor prototüübi veebilehe kontrollijast, mille eesmärk on vigade vähenemine selle kasutuse korral.

Veebilehe kontrollija aitab *webmaster*il veebilehte kontrollida, leides veebilehelt vigu, mida ilma põhjaliku kontrollita ei leia. Sellisteks vigadeks võivad olla: pealkirjade stiilid ja suurused ning SEO tähemärkide arv.

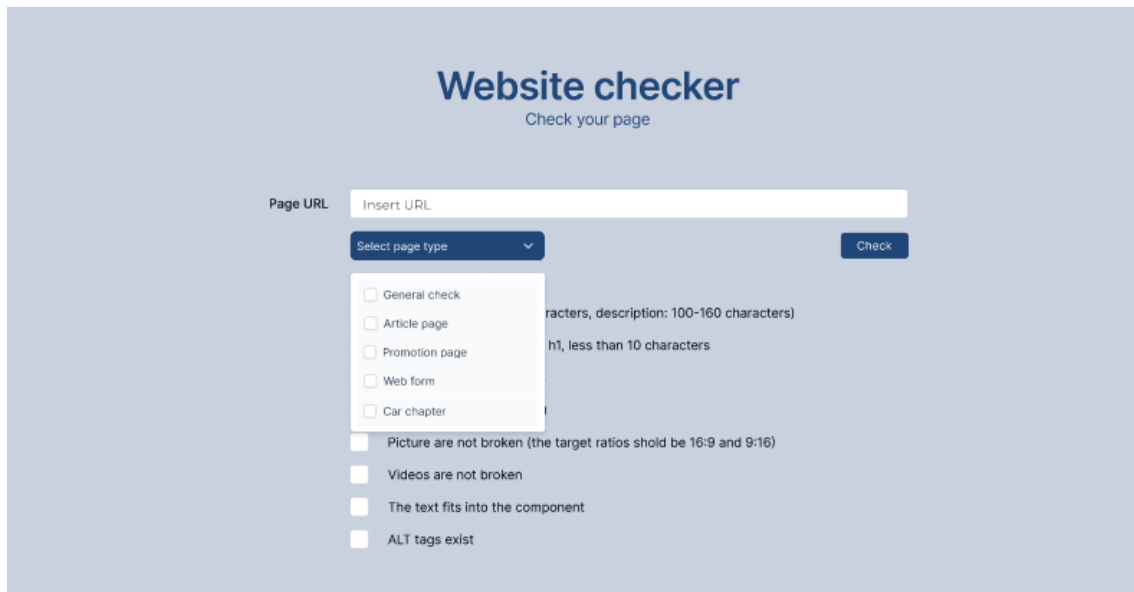
3.4 Prototüübi eesmärk ja funktsionaalsused

Veebilehe vigade kontrollimise prototüübi eesmärk on aidata *webmaster*itel oma tööd kontrollida, mille tulemusel vigade arv nende töös langeb. Kui *webmaster*id lõpetavad oma töö, saavad nad kontrollida kõiki veebilehti, milles on nad teinud muudatusi. Selleks tuleb neil sisestada rakendusse veebilehe URL, mis annab tagasiside sellele, kas veebileht

vastab nõuetele või mitte. Rakendus kontrollib veebilehti loodud *checklisti* alusel (vt Lisa 2).

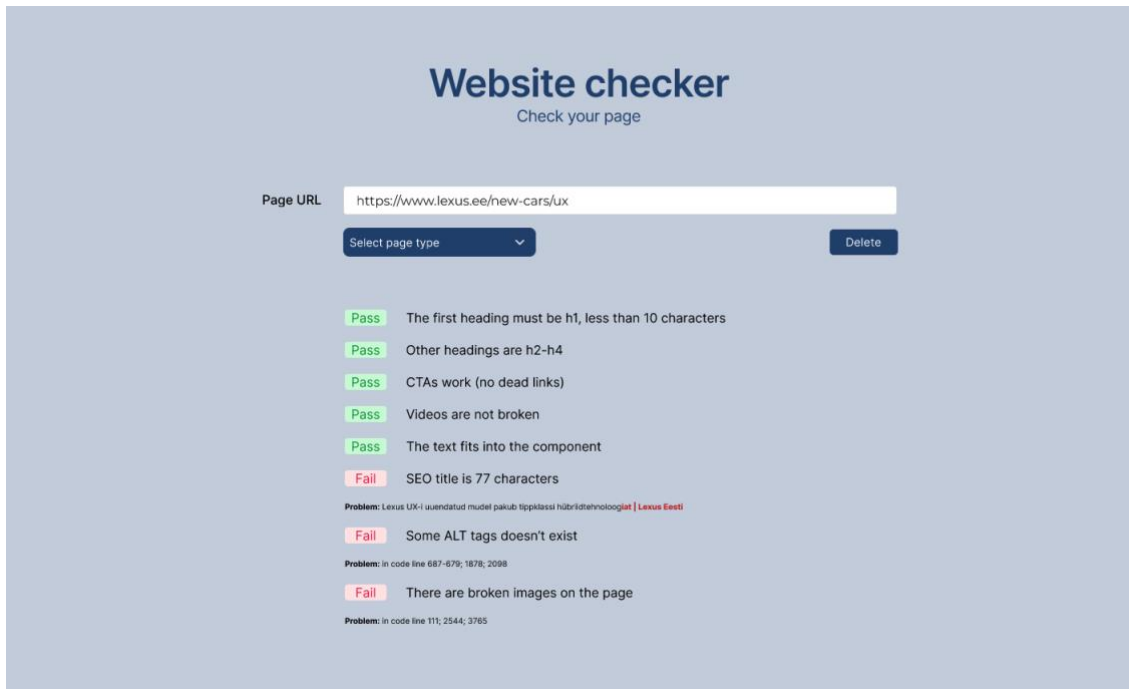
Veebilehe kontrollimise prototüübil on kaks peamist funktsionaalsust.

- Veebilehe URL'i lisamisel prototüüpi, annab veebilehe kontrollija vastuse, kas leht on kasutusele võtmiseks sobilik või vajab see täiendavad parandusi.
- Kasutaja saab nelja erineva veebilehe tüübi põhjal lehte kontrollida. Veebilehe tüüpideks on: artiklilehed, mudelilehed, pakkumiste lehed ja veebivormide lehed.

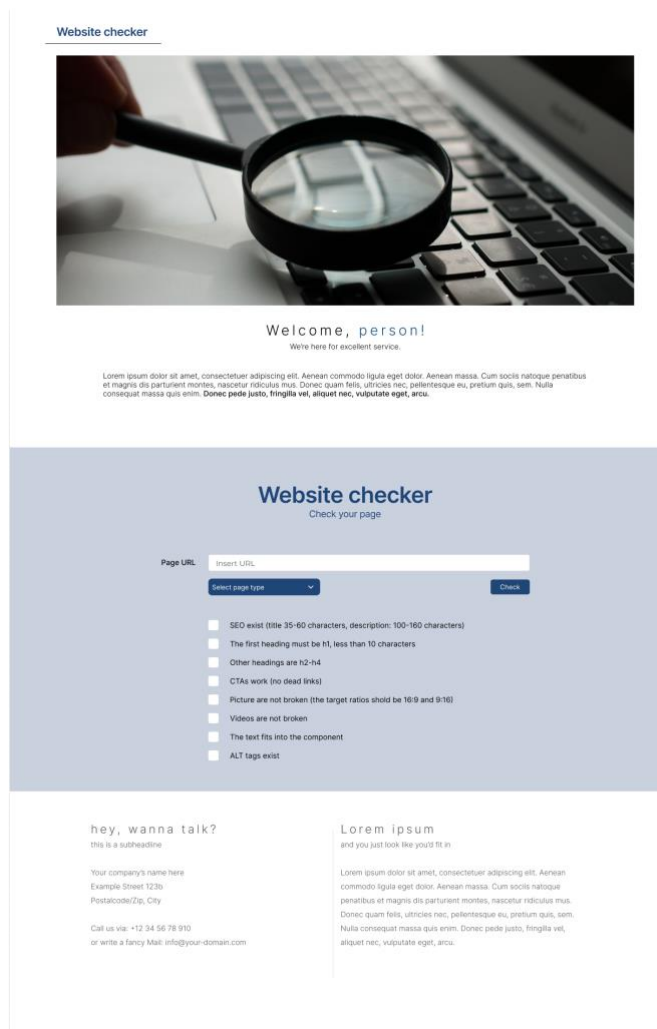


Joonis 5. Veebilehe kontrollimise prototüübi algvaade.

Kasutaja saab lisada veebilehe URL'i ja valida, millise veebilehe tüübiga tegu on. Kui kasutaja on need valikud ära teinud, saab ta kätte tulemuse ning tagasiside sellele, mis on veebilehe hetkes seis.



Joonis 6. Vaade prototüüpest peale kontrolli teostamist.



Joonis 7. Veebilehe kontrollimiseks loodud prototüüp täisvaates.

4 Tulemuste analüüs

Järgnevas peatükis analüüsitakse bakalaureuse töös saadud tulemusi vigade aruannetest ning seda, millist mõju avaldas *checklisti* kasutusele võtmine Toyota ja Lexuse sisuhaldusprojektis.

4.1 Vigade aruande tulemused ja analüüs

Augustikuus korraldas autor, koos ADMi projektide tiimijuhi ja strateegiajuhiga koosoleku, kus arutati Toyota ja Lexuse projektide hetkeolekut.

Koos tõdeti, et sisuhaldusprojektis esinevate vigade arv, mis jõuab kliendini on liiga suur. Lahenduste väljatöötamiseks ja plaani paika panemiseks kasutati PDCA (*Plan-Do-Check-Act*) tsükli.

Esmalt tuli teha erinevad tegurid.

- Mida arvab klient hetkeolukorrast ning kas teema oleks tarvis fookusesse võtta?
- Kas või milline on kliendi aktsepteeritav veamäär?
- Kas mingit tüüpi vead on kliendi jaoks olulisemad ja kriitilisemad, kui teised?
- Mis tuleks fookusesse võtta esmalt?
- Käia kliendiga läbi reaalsed faktid seoses sellega, kui palju on temani jõudnud vigu augustikuus kokku.

Kliendiga kokkulepitud aktsepteeritavaks veamääraks jäi 5% kõikidest töödest. Klient pidas vigu hinnakirjas või varustustasemes kõige kriitilisemaks. Lisaks mainis ta ka erinevaid nähtavaid vigu, milleks on katkised pildid, videod, kirjavead ja mittetöötavad veebilingid CTA'del (*call to action*) [21].

Vaadates tagasi tulemustele, tuli välja, et augustikuus oli Lexuse projektis kokku 15% vigu (24 *ticketist* olid 4 vigased) ja Toyotal 6% vigu (32 *ticketist* oli 2 vigased), mis jõudsid kliendini.

Lisaks lepiti kokku ka tiimisisene aktsepteeritav veamäär, milleks jäi 15%. Seda numbrit soovitakse saavutada tiimi teadlikkuse tõstmisega, probleemi olulisest rääkides.

Vigade mõõtmise ja esinemise eesmärgiks oli näha reaalselt tulemust, mis annaks ülevaate, kuidas projektis praegused QA protsessid töötavad ning kas need vajavad parandamist, võttes arvesse tulemusi. Lisaks andsid iganädalaselt projektijuhid *webmasteritele* ülevaate, kuidas läks eelmine nädal vigade tekkimisega, kus oleks saanud midagi paremini teha, kas esinenud vigades oli ühisosasid. Lisaks sellele, oli kõigil võimalus jagada oma mõtteid antud teema kohta. Selleks, et rõhutada teema olulisust, räägiti terve meeskonnaga igal nädalal ühel *stand-upil* vigade aruandest, analüüsides seda.

Alateist 2022. aasta septembrikuust on autor koos Toyota projektijuhiga mõõtnud ADMi Toyota ja Lexuse sisuhaldusprojekti tiimis vigade esinemist kõikides töödes. Üheks tööks loetakse Jirasse loodud *ticketit* ja üheks veaks loetakse Jira *ticketis* tehtud vigu. Tööülesanne on üldjuhul üheleheküljeline Jira *ticket*. Kõik vead, mis ühes *ticketis* tehakse võrduvad ühe veaga (vt Lisa 3).

Alljärgnev tabel 1 on ülevaade vigade tekkimisest Lexuse projektis ajavahemikus 12.09-04.12.22. Vigu ja nende esinemist tiimisiselt mõõdeti nädalate kaupa, lisaks sellele mõõdeti ka kliendini jõutavate vigade esinemist.

Tabel 1. Lexuse sisuhaldusprojekti vigade aruanne.

Kuupäev	Lexus				
	Tiimisiseste vigade %	Kliendini jõudnud vigade %	Kokku <i>ticket</i> eid	Vigased <i>ticket</i> id	Millest kliendini jõudis
12.09-18.09.22	16,7%	8,3%	12	2	1
19.09-25.09.22	14,3%	0,0%	14	2	0
26.09-2.10.22	7,1%	0,0%	14	1	0
3.10-9.10.22	0,0%	0,0%	8	0	0
10.10-16.10.22	8,3%	0,0%	12	1	0
17.10-23.10.22	11,1%	0,0%	9	1	0
24.10-30.10.22	0,0%	0,0%	8	0	0
31.10-6.11.22	14,3%	0,0%	7	1	0
7.11-13.11.22	14,3%	0,0%	7	1	0
14.11-20.11.22	0,0%	0,0%	4	0	0
21.11-27.11.22	16,7%	0,0%	6	1	0
28.11-04.12.22	12,5%	0,0%	8	1	0

Lexuse vigade mõõtmise tulemusi analüüsid selgub, et vigade määr, mis on kliendini jõudnud alates 19.09st, on püsinud 0% juures. Tiimisiseste vigade määr on kõikunud sõltuvalt nädalast 0-16,7% juures. Esimese nelja nädalaga 12.09-9.10.22 oli tiimisiseseid vigu kokku 10,4% (48 *ticket*ist, 5 esines viga), millest 1 viga ehk 2,1% jõudis kliendini. Järgneva nelja nädalaga 10.10-6.11.22 oli vigu 8,3% (36 *ticket*ist, 3 esines viga) ning 7.11-04.11.22 oli vigu 12% (25 *ticket*ist, 3 esines viga).

Järgnev tabel sisaldab Toyota projekti iganädalast vigade ülevaadet, alates 12.09-04.12.22 (vt Tabel 2).

Tabel 2. Toyota sisuhaldusprojekti vigade aruanne.

Kuupäev	Toyota				
	Tiimisiseste vigade %	Kliendini jõudnud vigade %	Kokku <i>ticket</i> eid	Vigased <i>ticket</i> id	Millest kliendini jõudis
12.09-18.09.22	27,3%	0,0%	22	6	0
19.09-25.09.22	30,0%	0,0%	20	6	0
26.09-2.10.22	0,0%	0,0%	8	0	0
3.10-9.10.22	0,0%	0,0%	9	0	0
10.10-16.10.22	16,7%	0,0%	6	1	0
17.10-23.10.22	0,0%	0,0%	7	0	0
24.10-30.10.22	14,3%	0,0%	7	1	0
31.10-6.11.22	13,3%	0,0%	15	2	0
7.11-13.11.22	12,0%	4,0%	25	3	1
14.11-20.11.22	6,3%	3,1%	32	2	1
21.11-27.11.22	15,0%	0,0%	20	3	0
28.11-04.12.22	21,4%	0,0%	14	3	0

Toyota vigade mõõtmise tulemusi analüüses selgub, et vigade määr, mis on kliendini jõudnud 12.09-06.11.22, on püsinud 0% juures. Alates 07.11-04.12 on vigade määr püsinud 0-4% vahel. Tiimisiseste vigade määr on kõikunud sõltuvalt nädalast 0-30%. Esimese nelja nädalaga 12.09-9.10.22 oli kokku tiimisiseseid vigu 20,3% (59 *ticket*ist, 12 esines viga) ja järgneva nelja nädalaga 10.10-6.11.22 oli vigu 11,4% (35 *ticket*ist, 4 esines viga). Alates 7.11-04.11.22 oli vigu 12,1% (91 *ticket*ist, 11 esines viga), millest 2 viga ehk 2,2% jõudis kliendini.

Kokkuvõtteks on näha, et tiimisiseste vigade määr kaheteistkümneme nädala jooksul oli Lexusel 10,1% ja Toyotal 14,6%, millest Lexusel jõudis kliendini viga 0,9% ja Toyotal 2,2%. Seega üks eesmärk, et kliendini jõuab kuni 5% vigu sai täidetud. Tiimisiseste

vigade määr kaheteistkümne nädala jooksul oli alla 15%, kuid nädala või nelja nädala lõikes oli ka tulemusi, mis olid üle 15% (vt tabel 1 ja 2). Nädalatel, kus veamäär oli suurem kui 15%, arutles autor koosoleku raames seda, kas nende vigade ära hoidmiseks oleks saanud midagi teistmoodi teha.

Peamised tekkinud veatüübid oli hooletusvead, mis tekkisid kas kiirustamisest, ebapiisavast testimisest või inimlikust faktorist, et kõiki vigu alati pole võimalik märgata.

4.2 Tööde topeltkontroll

Üheks võimaluseks kuidas vigade arvu vähendada on teostada topeltkontroll. Kui varasemalt vaatas *webmasteri* töö enne kliendini jõudmist üle ainult projektijuht, siis topeltkontrolliga kontrolliks *webmaster* üks *webmaster* kahe tööd veel lisaks. Ristkontrolli rakendamine võib olla edukas, kuid selle negatiivne pool on lisa-aeg, mis kulub ühele ülesande tegemisele.

Lõputöö autor rakendas topeltkontrolli meetodit enda meeskonnas ühe nädala. Eesmärgiks seati, et iga *webmaster* kontrollib iga päev vähemalt ühe või kahe teise *webmasteri* tööd. Hiljem vaatab töö üle veel ka projektijuht ning alles siis edastab selle kliendile.

Tööprotsess nägi välja järgmine: *Backlog* → *Selected for development* → *In progress* → QA *webmaster* → QA projektijuht → UAT → *Done*.

Enne antud lahenduse kasutusele võttu oli olemas kliendi nõusolek, kes oli teadlik, tundide mahu suurenemisest, ristkontrollimise kasutusele võtmisel.

Antud lahendust katsetati nädal aega, vahemikus 3.10-9.10.22. Selle nädala jooksul oli nii Toyota kui ka Lexuse vigade määr, mis jõudis kliendini ja projektijuhini 0% ehk Toyota projekti 8 *ticketis* ja Lexuse projekti 8 *ticketis* ei esinenud vigu. Seevastu leidsid *webmasterid* QA etapis ühes *ticketis* ühe vea, mis oleks muidu projektijuhini või kliendini jõudnud.

Nädal aega ristkontrolli katsetades, tõdes autor, et selline lahendus võib küll vigu vähendada ja olla edukas, kuid meeskonna tööaeg suureneb, mis tähendab omakorda

kliendile suuremaid arveid ja tähtaegade pikenemist. Seetõttu otsustati lahendusega mitte edasi liikuda ja proovida rakendada teisi võimalusi vigade vähendamiseks.

4.3 Jira protsesside parendused

Läbimõeldud ja töötavad Jira protsessid on hea projektijuhtimise aluseks. Jira töövoog tagab, et iga meeskonnaliige teaks, mis on tema ülesanded. Samuti annab Jira ülevaate kogu meeskonna töökoormusest. [22]

Üheks võimaluseks, kuidas töödes esinevaid vigu vähendada on tööprotsesside parendamine. Meeskonnas vaadati ühiselt üle kõik Jira protsessid ning veenduti selles, kas kõik tiimiliikmed mõistavad neid ühtemoodi. Lisaks loodi kokkuleppeid, mida tuleb jälgida ning arutleti, kas on kohti, kus oleks võimalik teha parendusi. Esmalt vaadati üle, kas kõik *Kanban boardil* olevad etapid on loogilised ja aitavad tööd lihtsustada. Nendeks etappideks olid: *Backlog*, *Selected for development*, *In progress*, *QA,UAT* ja *Done* (vt joonis 4). Koos tõdeti, et praegused etapid *ticketi* liikumisel on head ja neid ei ole tarvis muuta. Küll aga on tarvis muuta *ticketite* edasi-tagasi liikumist ühest etapist teise. Metoodikas mainitud TO-BE joonis (vt Joonis 4) sai aluseks, kuidas *ticket* peab ideaalis liikuma. Selle täitmiseks on aga tarvis vigu vähendada, et *ticket* ei peaks edasi-tagasi liikuma.

Autor leidis koos Toyota projektijuhiga, et kuigi Jira töövoog on hea, vajaks tähelepanu see, kuidas Jiras kommentaare kirjutada. Jira kommentaar peaks sisaldama: mida ja kuidas tehti, kus tehti muudatusi (veebilehe URL'id) ning seda, mida peaksid teised teadma antud ülesande kohta. [23]

Autor leppis tiimi ja kliendiga kokku, et viimane Jira kommentaar peab olema piisavalt informatiivne, et kui teine *webmaster* peaks töö üle võtma, saaks ta juba viimast kommentaari lugedes aru, millest on jutt ja mida ta tegema peab.

Kokkulepitud head tavad *ticketi* kommenteerimisel, peale töö lõpetamist:

- kirjutada lahti, mis sai tehtud ja kas midagi jäi pooleli, panna kirja miks see pooleli jäi ning kes sellega peaks edasi tegelema;
- lisada kõikide lehtede URL'id, mida muudeti;
- lisada kommentaari juurde probleemid, millele ei saadud lahendust;
- lisada mida veel vaja teha on kui ticket pooleli jääb.

4.4 Checklisti kasutuselevõtu analüüs

Lõputöö autor sai aru, et ainult teavitustööst ei piisa, et hoida tiimisiseste vigade määra 15% juures. Enesekontrolliks tuleb kasutada ka muid võimalusi. Üheks enesekontrollimise võimaluseks on kasutada *checklisti*, mis aitab *webmasteril* oma töö üle kontrollida. *Webmasterid* kasutavad töö tegemisel ja kontrollimisel erinevad tööriistu, näiteks: Web Developer, Change Case, Mate Translate, SEO Minion.

Loodud *checklist* on mõeldud kasutamiseks Toyota ja Lexuse veebilehtede kontrollimiseks. *Checklist* on loodud kõikide erinevate lehetüüpide jaoks. (vt Lisa 2)

Erinevad lehtede tüübid jagunevad neljaks.

- Autode mudelileht on leht, milles on kataloog, kus on võimalik näha autode omadusi, varustust, hinnakirja, brošüüre ning autot 360-kraadises vaates. Lisaks on võimalik konfigureerida endale sobiv auto.
- Artiklilehed võivad olla erinevate autode omaduste kohta või mõne konkreetse mudeli kohta, mis ei ole veel saadaval. Näiteks, elektriautode lehed, hübriidautode lehed, jt.
- Eripakkumiste lehed on mõne auto või teenuse pakkumised, mida soovitakse rõhutada.
- Veebivormide lehed sisaldavad kontaktivorme, proovisõiduvorme, auto- või hoolduse broneerimise vorme.

Autor tegi *checklisti* ning edastas selle tiimiliikmetele ja kliendile kooskõlastamiseks, et veenduda selle sobivuses Toyota ja Lexuse veebilehtedega. Ülevaatamise käigus tehti mõningaid parandusi ja soovitusi *checklisti* paremaks toimimiseks.

Järgnevalt kirjeldab autor *Checklisti* kasutamise protsessi. Kui *Webmaster* lõpetab töö, kontrollib ta oma töö *checklisti* alusel ära. Iga lehetüüp eri *checklisti* järgi. *Checklist* võeti kasutusele alates 2022. aasta novembrikuust. Selle aja jooksul on leitud, millal on *checklisti* kasutamine eriti vajalik ja edukas ning seda, millal saaks ka ilma selleta hakkama. Algatuseks prooviti *checklist* kasutusele võtta kõikides tööülesannetes, kuid hiljem mõisteti, et iga ülesande puhul on selle kasutamine ajamahukas ning see ei täida täielikult oma eesmärki.

Millal tuleks *checklisti* kasutada:

- lehe üldise auditi tegemise korral. Kui soovitakse käia kogu leht üle ja vaadata, kas lehel esineb vigu;
- uue lehe loomisel;
- *ticketites*, mille ülesandes tuleb muuta rohkem kui 5 veebilehte.

Millal ei ole vaja *checklisti* kasutada:

- pisikeste leheparanduste tegemisel (näiteks on tarvis muuta lehel ära üks pilt või tekst);
- kuni 5 veebilehe pisiparanduste tegemisel, ühetaoliste muudatuste tegemisel, näiteks pisikesed keelelised parandused;
- veebilehel olevate autode tehniliste andmete muutmisel, näiteks muudetakse auto kütusepaagi mahtuvust.

4.4.1 *Checklisti* kasutuselevõtu tulemused praktikas

Praktikas sai tiim *checklisti* kasutada *ticketite* puhul, mille ülesandepüstitus oli üldine lehe või lehtede ülevaatus. Autor lasi Toyota ja Lexuse *webmasteritel* *checklisti* kasutades valmistada neli mahukat *ticketit*. *Checklisti* kasutamise eesmärk antud tööde puhul oli leida probleemseid kohti lehtedel, mis ilma *checklistita* ülevaatamisel võiksid jääda

märkamata. See oli vajalik kuna veebilehe komponente, mida kontrollida on palju ning mõni neist võis kogemata jääda üle vaatamata.

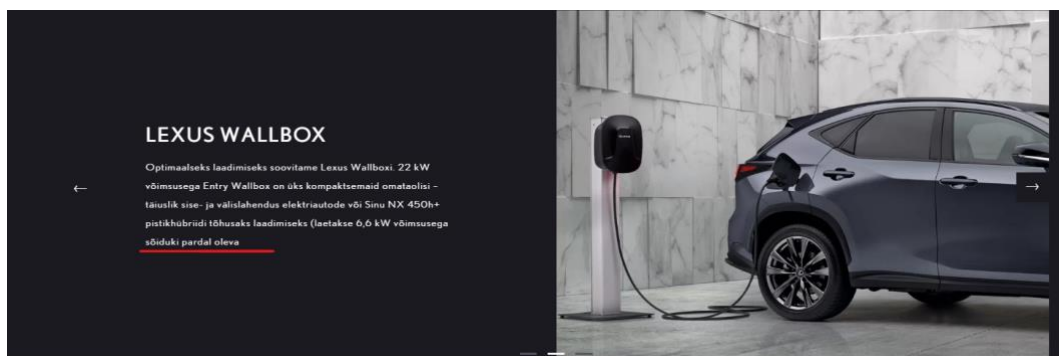
4.4.2 Nelja *ticketi* ülesandepüstitus ja vigade leidmise arv lehel

Tabel 3. *Checklisti* kasutuselevõtu tulemused esimese *ticketi* puhul.

Ülesandepüstitus	Veebilehede regioonid	Kontrollitavate lehtede arv	Leitud vigade arv
Teha 30'le populaarsemale veebilehele Google Analytics andmete põhjal Toyota veebilehele audit, kasutades <i>checklisti</i> .	EE/EERU	30	19
	LT	30	16
	LV/LVRU	30	26

Esinenud vead olid:

- katkised pildid galeriikomponendis;
- tekst ei mahu komponenti ära, komponenti on lisatud rohkem tähemärke, kui on võimalik ja tekst kuvab ära lõigatult;
- lehel olemasolevad pildid on kehva kvaliteediga või udused;
- kirjavead;
- CTA'de sõnastused erinevad kokkulepitust;
- mobiilivaates on pildid ära lõigatult.



Joonis 8. Ekraanipilt Lexuse pistikhübridi lehel, kus tekst ei mahu komponenti.

Tabel 4. *Checklisti* kasutuselevõtu tulemused teise *ticketi* puhul.

Ülesandepüstitus	Veebilehtede regioonid	Kontrollitavate lehtede arv	Leitud vigade arv
Teha 30'le populaarsemale veebilehele Google Analytics andmete põhjal Lexuse veebilehele audit, kasutades <i>checklisti</i> .	EE/EERU	30	14
	LT	30	12
	LV/LVRU	30	18

Esinenud vead olid:

- tekst ei mahu komponenti ära, komponenti on lisatud rohkem tähemärke kui on võimalik ja tekst kuvab ära lõigatult;
- tekstid ei ole kohalikus keeles (lehel esinevad inglise keelsed tekstid);
- kirjavead;
- lehel oleval pildil on kehv kvaliteet;
- mobiilivaates on pildid ära lõigatult.

Tabel 5. *Checklisti* kasutuselevõtu tulemused kolmanda *ticketi* puhul

Ülesandepüstitus	Veebilehtede regioonid	Kontrollitavate lehtede arv	Leitud vigade arv
Vene keele "Avasta" menüüpunkti ülevaatus.	EERU	45	18
	LVRU	45	21

Esinenud vead olid:

- katkised pildid/videod;
- tekst ei mahu komponenti ära, komponenti on lisatud rohkem tähemärke kui on võimalik ja tekst kuvab ära lõigatult;
- tekstid ei ole kohalikus keeles (lehel esinevad inglise keelsed tekstid);

- veebilehe struktuur ja ülesehitus erineb teisest regioonidest;
- mobiilivaates on pildid ära lõigatud.

Tabel 6. *Checklisti* kasutuselevõtu tulemused kolmanda *ticketi* puhul.

Ülesandepüsisus	Veebilehtede regioonid	Kontrollitavate lehtede arv	Leitud vigade arv
Vene keele elektrifitseeritud menüüpunkti ülevaatus.	EERU	12	7
	LVRU	12	8

Esinenud vead olid:

- katkised pildid ja videod;
- tekst ei mahu komponenti ära, komponenti on lisatud rohkem tähemärke kui on võimalik ja tekst kuvab ära lõigatud;
- tekstid ei ole kohalikus keeles (lehel esinevad inglise keelsed tekstid);
- veebilehe struktuur ja ülesehitus erineb teisest regioonidest;
- mobiilivaates on pildid ära lõigatud.

Saadud tulemuste kokkuvõttes on näha, et küllaltki palju vigu on leitud veebilehtedelt, kui on teostatud kontroll *checklisti* abiga. Paljud avastatud vead ei ole seotud veebilehe funktsionaalsustega, kuid võivad muuta kasutajakogemust.

Autor uuris *webmasterite* arvamust selle kohta, kuidas nad hindavad *checklisti* kasulikkust ning millist rolli mängis *checklisti* kasutamine nendes neljas *ticketis*. Oma hinnangu andsid kolm *webmasterit*, kes antud *ticketitega* tegelesid.

Küsitlusest selgus, et kõik kolm *webmasterit* hindavad *checklisti* kasulikuks eelmainitud *ticketite* puhul.

Küsitlusest selgus, et *webmasterite* hinnangul:

- aitab *checklisti* kasutamine meeles hoida kõike vajalikku;
- toodi välja, et *checklisti* on hea kasutada just üldist kontrolli vajavate ülesannete puhul;
- lisaks toodi välja, et *checklist* tuleks üle käia ka tavaliselt peale puhkust või siis kui alustada uue konkreetse lehetüübiga tööd;
- uutel töötajatel on tänu *checklistile* lihtsam, sest nad saavad tänu sellele kogu vajaliku informatsiooni kiiresti kätte.

Negatiivse poole pealt tõid kõik kolm *webmasterit* välja, et praegune *checklisti* kasutamine ei ole mugav ja tihti on selle kasutamine liialt ajamahukas. Lisaks toodi välja ka see, et kui oleks olemas rakendus, mis *checklistis* olevaid punkte üle vaatab, aitaks see kontrollimisele kulutavat aega kokku hoida.

Checklist on *webmasteritele* kättesaadav Toyota ja Lexuse projekti Wikis ja Microsoft Wordis. Uurides *webmasteritelt* kui mugav on *checklisti* kasutada, selgus, et Wikis ja Wordis ei ole *checklisti* kõige mugavam kasutada. Seetõttu viis autor *checklisti* üle Microsoft Excelisse, kus on võimalus vastavalt lehetüübile filtreerida *checklisti* punkte. Samuti on seal kohe võimalik ära märkida, kas *checklisti* punkt on üle kontrollitud.

Iga väiksemgi tõrge veebilehel mõjutab kasutajakogemust, mis võib omakorda avaldada mõju sellele, kuidas kliendid hindavad brändi usaldusväarsust. Kui veebileht ei ole kasutajasõbralik, siis 50% kasutajatest ei pöördu veebilehele enam kunagi tagasi. Lisaks sellele tasuks meeles pidada, et 96% veebilehtedest ei ole mõeldud mobiiltelefonis kasutamiseks. [24]

4.5 Prototüübi edasiarendused ja kasutusvaldkonnad

Luues prototüübist reaalse rakenduse, millega veebilehte kontrollida, võiks vigade määr Toyota ja Lexuse projektis jõuda alla 10%.

Valminud rakendust on võimalik edasi arendada, kohandades seda erinevate veebilehtede jaoks ning lisades juurde erinevaid kontrollelemente. Samuti on võimalik rakendust kohandada manuaaltestijate jaoks selliselt, et nad saaksid seda oma igapäeva töös kasutada. Lisaks sellele on võimalik lisada täiendavaid funktsioone loodud rakendusele.

Kui prototüüp luua universaalseks ning selliseks, et seda oleks võimalik kasutada erinevate projektide puhul, oleks võimalus rakenduse kasutamise eest küsida ka kasutajatelt tasu.

5 Kokkuvõte

Lõputöö peamine eesmärk oli viia sisuhaldusprojektis tehtavate vigade arv, mis jõuab kliendini 5% peale ja tiimisiseste vigade määr 15% peale. Selle jaoks kavandas autor nõuetele vastava veebirakenduse prototüübi, kontrollimaks veebilehel olevaid komponente.

Töö eesmärgi saavutamiseks loodi veebilehe kontrollimise *checklist*, kus pandi paika kõik olulised nõuded veebilehe kontrollimiseks. Loodud *checklist* on valideeritud kliendi poolt ja selle abil on võimalik *webmasteritel* oma tööd üle kontrollida. Veebilehe automaatseks kontrollimiseks alustati veebirakenduse prototüübi kavandamisega, mis kontrollib *checklistis* sisaldavaid punkte. Sisuhaldusprojektis tehtavate vigade arvu mõõdeti iga nädalaselt Jirasse loodud *ticketite* alusel.

Bakalaureusetöös püsitatud eesmärgid said täidetud. Vigade aruande tulemustest selgus, et kliendini ei jõudnud nädalas rohkem kui 5% vigu kõikidest valminud töödest. Tiimisiseste vigade määr 12 nädala lõikes oli Lexusel 10,1% ja Toyotal 14,6%, kuid esines ka nädalaid mil tiimisiseste vigade määr kõikus 0-30% vahel. Tulemuste põhjal sai järeldada, et ainult *checklisti* kasutamisest ei piisa tiimisiseste vigade vähendamiseks ja tarvis on kontrollimist automatiseerida. Seega loodud prototüübi realiseerimisel on võimalik viia tiimisiseste vigade määr veelgi madalamale.

Valminud töö edasiarendused hõlmavad endas arendusprotsessi, kus realiseeritakse valminud prototüüp. Kui prototüüp luua universaalseks on seda võimalik kasutada ka erinevate projektide puhul.

Kasutatud kirjandus

- [1] A. Jira, „What is a kanban board?“, 2022. [Võrgumaterjal]. Available: <https://www.atlassian.com/agile/kanban/boards>. [Kasutatud 20 12 2022].
- [2] E. Kuttichová, „How did Figma take over UI/UX market?“, 2022. [Võrgumaterjal]. Available: <https://uxplanet.org/how-did-figma-take-over-ui-ux-market-99c32352e947>.
- [3] A. Jira, „What is Jira used for? Atlassian Jira“, Atlassian Jira, 2022. [Võrgumaterjal]. Available: <https://www.atlassian.com/software/jira/guides/use-cases/what-is-jira-used-for#jira-for-agile-teams>. [Kasutatud 20 12 2022].
- [4] A. S. Gillis, „Quality assurance (QA)“, [Võrgumaterjal]. Available: <https://www.techtarget.com/searchsoftwarequality/definition/quality-assurance>. [Kasutatud 22 12 2022].
- [5] „What Is SEO – Search Engine Optimization?“, 5 12 22. [Võrgumaterjal]. Available: <https://searchengineland.com/guide/what-is-seo>.
- [6] Tehnopedia, „Stand-Up Meetings“, 2022. [Võrgumaterjal]. Available: tehnopedia.com/definition/25923/stand-up-meeting. [Kasutatud 28 12 2022].
- [7] U-TOR, „Manual vs Automation Testing: What to Choose For Your Project“, 15 6 2021. [Võrgumaterjal]. Available: <https://u-tor.com/topic/manual-vs-automation>.
- [8] Y. Morgenstern, „Small Mistakes Lose Big Money: How To Avoid Common Errors On Digital Platforms“, Forbes, 23 12 2022. [Võrgumaterjal]. Available: <https://www.forbes.com/sites/forbestechcouncil/2022/12/23/small-mistakes-lose-big-money-how-to-avoid-common-errors-on-digital-platforms/?sh=315170353e1b>. [Kasutatud 26 12 2022].
- [9] IEEE Explore, „Critical Analysis of Manual Versus Automation Testing“, %1 2021 *International Conference on Computational Performance Evaluation (ComPE)*, Shillong, India, 2021.

- [10] „Agile Alliance,“ Agile Alliance, 2022. [Võrgumaterjal]. Available: <https://www.agilealliance.org/agile101/>.
- [11] D. J. McGrath, *The Little Book of Big Decision Models*, Pearson Education Limited, 2015.
- [12] „Adobe Experience Manager (AEM): What It Is and How It Works,“ 8 12 2021. [Võrgumaterjal]. Available: <https://www.contentbloom.com/blog/adobe-experience-manager-aem-what-it-is-and-how-it-works/>.
- [13] Smartsheet, „What Is Jira?,“ 2018. [Võrgumaterjal]. Available: <https://www.smartsheet.com/jira-ticketing-management>. [Kasutatud 20 12 2022].
- [14] ADM Interactive, „ADM Interactive,“ [Võrgumaterjal]. Available: <https://www.adm.ee/kes-me-oleme/>.
- [15] V. V. j. J. Laanpere, „Tarkvara analüüs ja testimine,“ Mart Laanpere.
- [16] „For Tridion 2013 SP1 HR1 or SDL Tridion Sites 9.x how to perform an unattended install of Content Manager?,“ 8 6 2020. [Võrgumaterjal]. Available: <https://gateway.sdl.com/apex/communityknowledge?articleName=000003857>.
- [17] „Võõrsõnade leksikon,“ [Võrgumaterjal]. Available: <https://www.eki.ee/dict/vsl/index.cgi?Q=protot%C3%BC%C3%BCp>.
- [18] B. C. Dan Goodwin, *Designing UX: Prototyping*, SitePoint, 2017.
- [19] E. Canziba, *Hands-On UX Design for Developers*, Packt Publishing, 2018.
- [20] A. K. T. Vatsal Sharma, „A Study on User Interface and User Experience Designs and its Tools,“ *World Journal of Research and Review*, kd. 12, nr 6, pp. 41-44, 2021.
- [21] E. Varagouli, „What Is a Call-to-Action? CTA Meaning & Their Importance,“ Semrush Blog, 2021. [Võrgumaterjal]. Available: <https://www.semrush.com/blog/what-is-a-call-to-action/>. [Kasutatud 1 12 2022].
- [22] Atlassian, „Jira workflows power effective teamwork,“ [Võrgumaterjal]. Available: <https://www.atlassian.com/software/jira/features/workflows>. [Kasutatud 22 11 2022].
- [23] R. Wright, „Jira User Best Practices,“ 2020. [Võrgumaterjal]. Available: <https://www.jirastrategy.com/jira-user-best-practices/>. [Kasutatud 17 12 2022].

- [24] J. A., „Smallbizgenius,“ 21 2 2022. [Võrgumaterjal]. Available:
<https://www.smallbizgenius.net/by-the-numbers/ux-statistics/#gref>. [Kasutatud 26
12 2022].
- [25] L. S. Sterling, The Art of Agent-Oriented Modeling, London: The MIT Press,
2009.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Hanna Liivia Teller

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Webmasterite töö testimise ja kvaliteedi tagamise rakenduse prototüüp“, mille juhendaja on Jelena Vendelin.
 - 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.

31.12.2022

¹ 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 punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – QA checklist lehtede kontrollimiseks

Checklist Toyota/Lexuse veebilehtede jaoks	Lehe tüüp	Staatust
Tekst mahub komponentidesse	Üldine	In progress
Pildid ja videod on töökorras ja hea kvaliteediga	Üldine	ERROR
Lehe esimene pealkiri peab olema h1, vähem kui 10 sõna	Üldine	OK
Lehe alapealkirjad on h2-h4	Üldine	OK
CTAd töötavad (ei ole surnud linke) ja välislink avaneb uues aknas	Üldine	OK
Pildid, videod töötavad	Üldine	In progress
Tekst on kohalikus keeles	Üldine	
Alt tagid lisatud ja tõlgitud	Üldine	
Sina/Teie vastavalt riigile, kas suure või väikse tähega	Üldine	
Teksid ei ole UPPERCASEs	Üldine	
CO ₂ - väike number	Üldine	
Mobiilivaade korras (sh. pildid jne)	Üldine	
Inglise keelse lingi taga on (ENG)	Üldine	
Vaadata, et Model Key Specs'i komponendis oleks numbrite juures disclaimeri infonupud olemas.	Mudelileht	
Testida kõik lingid läbi, et avaneks õigesti ja avanev leht oleks õiges keeles	Mudelileht	
Hinna brošüüri/ hinnakirja CTAle on vaja lisada igale mudelile ja riigile erinev fail	Mudelileht	
Galerii all peab olema disclaimer	Mudelileht	
Galerii peab olema lehe ülaosas teine komponent, grades kõige all.	Mudelileht	
CO ₂ , kütusekulu disclaimerid peavad olema tõlgitud (, kui link on olemas, kontrollida, et avaneks õige leht)	Mudelileht	
"Vaata lisaks" komponent õiges keeles (Venel viivad õige riigi lehele)	Artiklileht	
Kontrollida vormidel kõik konteineris olevate elementide osad üle, et ei jääks ingliskeelseid tekste	Vormid	
Testida vormid läbi, kontrollida ka captcha täitmist/mitte täitmist, jällegi ingliskeelsete tekstide vältimiseks	Vormid	
"Saada" või "Edasi" nupu reeglite all võib mõnikord olla mõni error disclaimer inglise keeles (nt recall vorm)	Vormid	
Jälgime, et oleksid õiged kuupäevad	Pakkumiste leht	
Pealkirjades "Minu" mitte "Sinu"	Portaal	
Võimalusel testida funktsionaalsused päriselt läbi - minu autode alt saab autode VIN koodi millega testida	Portaal	

Joonis 9. QA checklist.

Lisa 3 – Vigade iganädalane raport

Tabel 7. Toyota ja Lexuse vigade aruanne.

Kuupäev	Toyota			Lexus		
	Tiimisiseste vigade %	Kliendini jõudnud vigade %	Kokku ticketeid	Tiimisiseste vigade %	Kliendini jõudnud vigade %	Kokku ticketeid
12.10-18.09.22	27,3%	0,0%	22	12,5%	8,3%	12
19.10-25.09.22	30,0%	0,0%	20	14,3%	0,0%	14
26.10-2.10.22	0,0%	0,0%	8	7,1%	0,0%	14
3.10-9.10.22	0,0%	0,0%	9	0,0%	0,0%	8
10.10-16.10.22	16,7%	0,0%	6	8,3%	0,0%	12
17.10-23.10.22	0,0%	0,0%	7	11,1%	0,0%	9
24.10-30.10.22	14,3%	0,0%	7	0,0%	0,0%	8
31.10-6.11.22	13,3%	0,0%	15	14,3%	0,0%	7
7.11-13.11.22	12,0%	4,0%	25	7,1%	0,0%	7
14.11-20.11.22	6,3%	1,6%	32	0,0%	0,0%	4
21.11-27.11.22	15,0%	0,0%	20	16,7%	0,0%	6
28.11-04.12.22	21,4%	0,0%	14	12,5%	0,0%	8