

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond  
Tarkvarateaduse instituut

Raido Roben 163066IABM

**KASUTATAVUSE HINDAMISE  
AUTOMATISEERIMINE RIIGI  
INFOSÜSTEEMIDE AMETI NÄITEL**

Magistritöö

Juhendaja: Ahto Kalja

Professor,  
filosoofiadoktor

Jevgeni Marenkov

Tehnikateaduse  
magister

Tallinn 2018

## **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: Raido Roben

07.05.2018

## **Annotatsioon**

Käesoleva töö eesmärgiks on täiendada avaliku sektori asutuse tarkvaraarendustsükli vahendiga, mis võimaldab veebikeskkondade automaatset kasutatavuse hindamist. Samuti on eesmärgiks tõestada, et vastava tarkvara rakendamine aitab vähendada inimressursi kaasamist kasutatavuse hindamisel ning kärpida üleüldisi arendusprotsessi kulusid.

Antud magistritöö raames paigaldati Riigi Infosüsteemide Ameti infrastruktuuri kasutatavuse automaatse hindamise tarkvara Guideliner, mille reeglibaas sisaldab nii kasutatavuse parimaid praktikaid kui ka asutuse spetsiifilisi ärireegleid. Antud tarkvara võimaldab veebirakenduse kasutatavuse testimist nii arendus-, testimis- kui ka paigaldusfaasis.

Toetudes varasematele allikatele esitatakse töö käigus tõestus, et automaattestimise tarkvara võimaldab vältida ekspertide rakendamist kasutatavuse hindamisel ning vähendada seeläbi hindamisprotsessi maksumust. Lisaks, kuna tarkvara võimaldab probleemide tuvastamist varajases arendusfaasis, aitab selle rakendamine kärpida arendusprotsessi kulusid 3-6 korda.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 64 leheküljel, 5 peatükki, 30 joonist, 3 tabelit.

## **Abstract**

### **Automation of Usability Evaluation Based on the Example of Republic of Estonia Information System Authority**

This thesis aims to improve a public sector organization software development process. This is done by introducing automated usability evaluation tool during web environment development. Another aim is making the process of evaluation less costly and requiring less human resources. This is achieved by researching usability evaluation during website development process.

The first goal was attained by deploying an automated usability evaluation software Guideliner into the infrastructure of Republic of Estonia Information System Authority. The tool combines guidelines extracted from best practices and rules based on specific business requirements. In addition, it enables usability evaluation during development, testing and deployment phases.

To accomplish the second goal a research backed by theoretical materials was conducted. The thesis reached to two important conclusions. Firstly, it was shown that usability evaluation software allows to cut out the involvement of human experts and reduce the evaluation process cost. Secondly, the findings demonstrate, that the given tool helps to discover usability issues in early stage and therefore reduces the overall costs of software development by 3-6 times.

The thesis is in estonian and contains 64 pages of text, 5 chapters, 30 figures, 3 tables.

## Lühendite ja mõistete sõnastik

BA	<i>ButtonAdapter</i>
CSS	<i>Cascading Style Sheets</i>
ETIS	Eesti Teadusinfosüsteemi
GR	<i>Guideline Repository</i>
HTML	<i>Hyper Text Markup Language</i>
ISO	<i>International Organization for Standardization</i>
JS	<i>JavaScript</i>
KB	Kilobait
NIST	<i>National Institute of Standards and Technology</i>
RIA	Riigi Infosüsteemide Amet
RIHA	Riigi Infosüsteemi haldussüsteemi
SEO	<i>Search Engine Optimization)</i>
UEE	<i>Usability Evaluation Engine</i>
URL	<i>Uniform Resource Locator</i>
VPN	<i>virtual private network</i>
WCAG	<i>Web Content Accessibility Guidelines</i>

## Sisukord

1 Sissejuhatus .....	11
2 Kasutatavus.....	13
2.1 Kasutatavus, kasutajakogemus ning juurdepääsetavus .....	14
2.2 Kasutatavuse olulisus .....	18
2.3 Kasutatavuse viis dimensiooni .....	20
2.3.1 Tulemuslikkus .....	20
2.3.2 Tõhusus.....	21
2.3.3 Veatuvastus.....	22
2.3.4 Kaasavus .....	24
2.3.5 Kergesti õpitavus .....	27
2.4 Vahekord kasutatavuse dimensioonide vahel.....	27
3 Kasutatavuse testimise meetodid erinevates arendusetappides .....	29
3.1 Ülevaatusmeetodid .....	29
3.1.1 Pluraalne ülevaatus .....	30
3.1.2 Heuristiline ülevaatus .....	30
3.1.3 Kognitiivne ülevaatus .....	30
3.2 Arendaja test (valge kasti testimine) .....	31
3.3 Lõppkasutajaga testimise meetodid.....	31
3.3.1 Konkurendiga testimine.....	32
3.3.2 Eemalt testimine .....	32
3.3.3 Võlur Oz .....	33
3.4 Live süsteemi testimise meetodid.....	33
3.4.1 Kriitiliste juhtumite tehnika.....	34
3.4.2 Veebianalüütika .....	34
3.4.3 Hindamiskaala.....	35
4 Kasutatavuse testimine RIA's .....	38
4.1 Eesti.ee kasutatavuse parandamise vajadus .....	38
4.2 Praegused meetodid kasutatavuse hindamiseks .....	44
4.3 Automaatse kasutatavuse hindamise tarkvara Guideliner .....	45

4.3.1 Guidelineeri kontseptsiooni tõestus.....	45
4.3.2 Guidelineeri arhitektuur ja tööpõhimõte .....	47
4.3.3 Guidelineeri reeglistik .....	49
4.4 Guidelineeri paigaldamine RIA süsteemidesse .....	51
4.5 Tarkvara muudatuse maksumus erinevates arendusetappides.....	56
4.5.1 Koodimuudatuse maksumuse eksponentsiaalne kõver.....	56
4.5.2 Koodimuudatuse maksumuse sirgjoon.....	57
4.5.3 Koodimuudatuse maksumuse tänapäevane statistika .....	59
4.6 Tulemused .....	60
5 Kokkuvõte .....	61
Kasutatud kirjandus .....	63
Lisa 1 –README.md fail .....	65

## Jooniste loetelu

Joonis 1. Kasutajakogemus, kasutatavus ja juurdepääsetavus .....	14
Joonis 2. iPhone X info.....	15
Joonis 3. Navigeerimisriba .....	16
Joonis 4. Vale atribuutide kasutus .....	17
Joonis 5. Õige atribuutide kasutus .....	17
Joonis 6. Eesti.ee juurdepääsetavuse valikud .....	18
Joonis 7. Dropboxi sisselogimise veateade .....	23
Joonis 8. Lego 404 leht.....	24
Joonis 9. FitBit'i koduleht .....	25
Joonis 10. PC World koduleht .....	26
Joonis 11. Apple avaleht.....	26
Joonis 12. Kasutatavuse testimise meetodid erinevates arendusetappides .....	29
Joonis 13. Likert'i skaala.....	36
Joonis 14. Semantiline diferentsiaalskaala .....	36
Joonis 15. Internetikasutajate osatähtsus 16–74-aastaste seas vanuserühma järgi.....	39
Joonis 16. Riigiportaali mobiilivaated.....	41
Joonis 17. Eesti.ee elukohateate avaldus .....	42
Joonis 18. Elukohateate kalendri avamise nupp.....	43
Joonis 19. Elukohateate kalendri kuupäeva valik mobiilis.....	43
Joonis 20. Guidelineri arhitektuur .....	48
Joonis 21. Lähtekoodi haldus .....	52
Joonis 22. Uue virtuaalmasina loomine.....	53
Joonis 23. Testide käivitamise käsk .....	53
Joonis 24. Jenkins'i projekti skript täiendavate funktsionaalsuste jaoks.....	54
Joonis 25. Jenkins'i testitulemuste trend .....	55
Joonis 26. Jenkins'i testitulemused kategooriatena .....	55
Joonis 27. Jenkins'i konkreetse kategooria testi tulemused .....	55
Joonis 28. RIA arendustükkel pärast Guidelinerit .....	56
Joonis 29. Boehmi hinna kasvu kõver .....	57



Joonis 30. Kent Becki muudatuste maksumuse sirgjoon .....	58
---	----

## **Tabelite loetelu**

Tabel 1. Eesti avaliku sektori veebilehtede vastavus kasutatavuse kriteeriumitele .....	46
Tabel 2. Guidelineri eeldefineeritud reeglid kategooriate kaupa .....	50
Tabel 3. NIST'i statistika muudatuste tegemise maksumusest .....	59

# 1 Sissejuhatus

Maailm on pidevas muutumises ning üha enam toimingutest, mida varasemalt tehti käsitsi, tehakse tänapäeval automaatselt IT-süsteemide poolt. Näiteks Eesti mootorsõiduki juhendaja tunnistus, mida 3 aastat tagasi väljastati Maanteeameti teenindusbüroos paberil, on võimalik tänasel päeval taotleda e-teeninduses<sup>1</sup>. Rääkides Eesti riigi veebiteenustest konkreetsemalt, siis üks maailma juhtivamaid tehnoloogia väljaandeid Wired peab digitaalse ühiskonna musternäiteks just eestlaste e-riiki<sup>2</sup>. Näide e-riigi võimekusest on tulude deklareerimine. Kui veel aastaid tagasi oli tegu aeganõudva protseduuriga, kus andmete sisestamine toimus käsitsi, siis tänu eeltäidetud andmetele saab alates 2015. aastast esitada deklaratsiooni ühe klikiga<sup>3</sup>. Olgugi, et digitaalne ühiskond on nii riigi kui kodanike jaoks mugav ning aega säästev, tuleb arvestada, et automatiseerimise käigus lisandub e-riigile üha uusi teenuseid ja vastutust. Kuivõrd riik peab kohtlema oma kodanikke võrdselt, ei saa järeleandmisi teha riigiportaali juurdepääsetavuses. Lisaks peab arvestama süsteemi disainil veebivaadete kohaldamist tahvelarvutite ja nutitelefonidega, mille osakaal<sup>4</sup> interneti toimingutes suureneb üha enam. Keerukust lisavad ka seadused ning direktiivid, mille eesmärk on ühtlustada avaliku sektori veebikeskkondi nii riigi kui rahvusvahelisel tasandil. Selleks, et Eesti e-riik oleks jätkusuutlik ka tulevikus, tuleb juba täna pöörata tähelepanu selle kasutatavuse edendamisele. Antud töös on vaatluse all Riigi Infosüsteemide Ameti (RIA) poolt hallatav veebikeskkond eesti.ee.

Kuna RIA soov on muuta senine kasutatavuse hindamine süsteemsemaks ning võimaldada kasutatavuse testimist paralleelselt arendustegevusega, on magistritöö

---

<sup>1</sup> <https://sakala.postimees.ee/3269227/mootorsoidukijuhi-juhendaja-tunnistust-saab-taotleda-e-teeninduses> (04.05.2018)

<sup>2</sup> <http://www.wired.co.uk/article/estonia-e-resident> (28.03.2018)

<sup>3</sup> <https://www.err.ee/529841/eeltaidetud-tuludeklaratsiooni-saab-nuud-esitada-vaid-uhe-klikiga> (28.03.2018)

<sup>4</sup> <http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide#monthly-200901-201804> (28.03.2018)

esimeseks eesmärgiks täiendada asutuse tänast arendustsüklit kasutatavuse automaattestimise tarkvaraga. Teine eesmärk on tõestada, et kasutatavuse hindamise automatiseerimine võimaldab vähendada arendusprotsessi kulusid ning inimtööjõu osalust.

Esimese eesmärgi saavutamiseks paigaldati RIA süsteemidesse kasutatavuse automaattestimise tööriist Guideliner. Teise eesmärgini jõudmiseks analüüsiti olemasolevaid materjale Guidelineri tulemuslikkusest ning vigade avastamise maksumusest erinevatel arendusetappidel.

Magistritöö jaguneb kolmeks peatükiks. Esimeses peatükis defineeritakse kasutatavus ning sellega seotud mõisted: kasutajakogemus ning juurdepääsetavus. Samuti räägitakse kasutatavuse olulisusest nii avaliku kui erasektori asutuste jaoks ning kirjeldatakse kasutatavuse tähtsamaid karakteristikuid. Teises peatükis tutvustatakse meetodeid, mida erinevates tarkavaraarenduse etappides saab kasutada. Kolmandas peatükis räägitakse RIA olemusest ning põhjustest, miks on vaja Eesti riigiportaali kasutatavust edendada. Samuti tutvustatakse RIA süsteemidesse lisatud tarkvara ning toiminguid, mida tarkvara paigaldamiseks tuli teha. Lisaks võrreldakse RIA arendusprotsessi enne ja pärast uue tarkvara olemasolu ning esitatakse tõestus kasutatavuse automaattestimise kasulikkusest.

Töö teema on aktuaalne, sest RIA poolt hallatav eesti.ee korraldab üha suuremas ulatuses riigi ning kodaniku vahelist suhtlust ning töötleb kõikide eestlaste delikaatseid isikuandmeid. Selleks, et kodaniku usaldus riigi vastu säiliks, peab riigiportaal olema töökindel, lihtsasti kasutatav ning kõikidele kasutajatele võrdsetel tingimustel kättesaadav.

Lõputöö on kirjutatud 2018. aasta kevadel Tallinna Tehnikaülikooli professori Ahto Kalja ning doktorandi Jevgeni Marenkovi juhendamisel.

## 2 Kasutatavus

Paarkümmend aastat tagasi oli interneti külastajate arv tuhandeid kordi väiksem kui tänapäeval<sup>1</sup>. Kui toona oli keskmine kasutaja ekspert, kes pärast avaliku veebi sündi 1993. aastal<sup>2</sup> pidi interneti kasutamiseks omama spetsiifilist riistvara ja spetsialisti teadmisi, siis praeguseks ajaks on internetist saanud laiatarbe teenus, mis on kergesti kättesaadav ning mille jaoks vajalikku seadet omab pea igäüks alustades lapsest ning lõpetades pensionäriaga. Kuna kasutatavuse hindamise meetodid tekkisid 1995. aastal<sup>3</sup>, mil erinevaid veebilehti oli kõigest 23 500<sup>4</sup> ning analoogseid veebikeskkondi võrdlemisi vähe, siis sel perioodil veebilehtede väljanägemisele erilist tähelepanu ei pööratud. Tänapäeval on veebilehtede koguarv 1,8 miljardit<sup>8</sup> ning alternatiivsete veebikeskkondade valik rikkalik. Võrreldes paarikümne aasta taguse ajaga on keskmine kasutaja erasektori veebilehtedel muutunud tunduvalt valivamaks ning seetõttu tuleb veebikeskkonna haldajal leida uusi mooduseid konkurentide edestamiseks ja turuosa suurendamiseks. Avaliku sektori lehtedele on see-eest juurde tekkinud palju uusi teenuseid ning mingil ajahetkel on oodata, et kogu suhtlus inimese ja riigi vahel on paberi pealt üle kolunud internetti. Seega avaliku sektori portaalidel, mis peegeldavad riigi nägu, lasub kasutajate ees üha suurem vastutus ning usaldusväarsuse säilitamiseks tuleb arendada kiireid ja töökindlaid teenuseid. Viis, kuidas era- ja avaliku sektori asutused saavad oma veebikeskkondi soovitud suunal täiustada, on läbi hea kasutatavuse.

---

<sup>1</sup> <http://www.internetlivestats.com/internet-users/> (21.03.2018)

<sup>2</sup> <https://home.cern/topics/birth-web> (21.03.2018)

<sup>3</sup> <https://www.nngroup.com/articles/summary-of-usability-inspection-methods/> (21.03.2018)

<sup>4</sup> <http://www.internetlivestats.com/total-number-of-websites/> (21.03.2018)

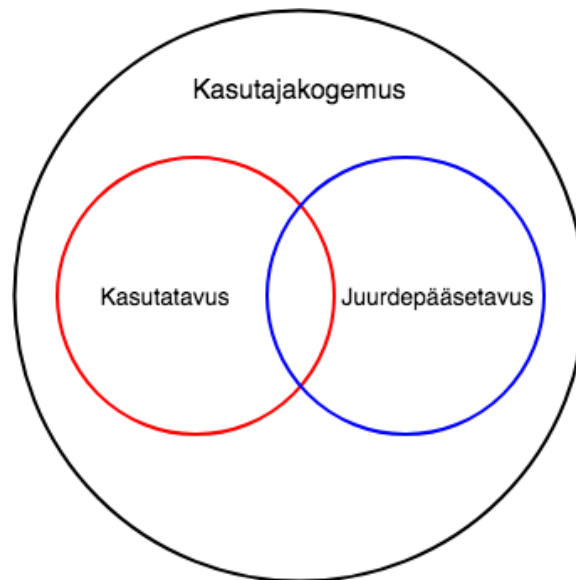
## 2.1 Kasutatavus, kasutajakogemus ning juurdepääsetavus

Selleks, et kasutatavusest paremini aru saada, võrreldakse seda kasutajakogemuse ja juurdepääsetavusega – mõistetega, mis on kasutatavusega tihedalt seotud ning mida tihtipeale ekslikult samastatakse.

Vastavalt ISO 9241-11 standardile [1] mõeldakse kasutatavuse all ulatust, mille raames saab konkreetne kasutaja konkreetses kontekstis toodet enda eesmärkide saavutamiseks tulemuslikult, tõhusalt ja rahuldust pakuvalt kasutada (ingl *The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*).

Kasutajakogemust defineerib ISO 9241-210 [2] kui tunnet, mida inimene tajub toote, teenuse või süsteemi kasutamisel (ingl *A person's perceptions and responses that result from the use and/or anticipated use of a product, system or service*).

Eelmainitud definitsioonidest järeldub, et kasutajakogemus on üldisem kui kasutatavus. Kui kasutajakogemus on terviktunne, mida kasutaja süsteemi kasutades tajub, siis kasutatavus on see osa tundest, mis seostub süsteemi kasutamise lihtsuse ja efektiivsusega. Seega kasutatavus on kasutajakogemuse alamosa (näidatud joonisel 1).



Joonis 1. Kasutajakogemus, kasutatavus ja juurdepääsetavus

Näide heast kasutatavusest ja kasutajakogemusest on Apple'i koduleht<sup>1</sup>. Joonisel 2 on illustreeritud uue iPhone X'i kohta käivat informatsiooni. Pika spetsifikatsiooni asemel toob Apple välja vaid primaarseima. iPhone X'i puhul on selleks uue lahendusega ekraan, mis on ehitatud servast servani. Kasutatavuse seisukohalt on tegu hea disainiga, sest kõige olulisem info hakkab kiiresti silma ning esmasel tutvumisel rohkemat polegi vaja (seotud tulemuslikkuse ja tõhususega). Samas, kui kasutaja soovib, on jäetud võimalus tutvuda tehniliste andmetega ka detailselt ("tehnilised andmed" nupp). Kasutajakogemuse tõstmiseks on Apple lisanud iPhone X'st kõrge kvaliteediga pildi ning "pakub silmailu" fraasi, mille eesmärk on pärast veebilehe külastust tekitada kliendis meelde jääv emotsioon ning tahe soetada endale uus iPhone.

iPhone X Ülevaade iOS Tehnilised andmed [Osta](#)

Super Retina ekraan

**iPhone X ise ongi ekraan. Uhiuus 5,8-tolline Super Retina ekraan täidab peopesa ja pakub silmailu.<sup>1</sup>**



Joonis 2. iPhone X info

---

<sup>1</sup> <https://www.apple.com/ee/> (31.03.2018)

Joonisel 3 on välja toodud Apple kodulehe navigeerimisriba. Disainilt on tegu lihtsa ja loogilise lahendusega, kus lisaks sõnadele kasutatakse ka pilte. Kasutajakogemusest pakub antud navigeerimine kliendile põnevust. Kasutatavuse poolest parandab see toodete eristamist ning väldib sõnade kokku sulamist.



Joonis 3. Navigeerimisriba

Ostu vormistamisel lisab Apple võimaluse avada suhtlusaken tehnilise toega, pakkudes võimaluse esitada täiendavaid küsimusi (seotud kasutatavuse ja rahuldusega) ning kasvatades kindlust tehingu sooritamiseks (seotud kasutuskogemusega).

Nagu näidatud joonisel 1, siis ka juurdepääsetavus on alamosa kasutajakogemusest. Vastavalt W3C'le peaks veebi disainima nii, et seda saavad kasutada kõik inimesed sõltumata riistvarast, tarkvarast, räägitavast keelest, asukohast või teovõimest. Kui antud tingimused on täidetud ning veeb on samadel tingimustel kättesaadav nii puudega kui puudeta inimesele, on saavutatud juurdepääsetavus<sup>1</sup>.

Ehk juurdepääsetavus moodustab selle osa kasutajakogemusest, millega tagatakse, et puudega inimene poleks puudeta inimesest ebasoodsamas olukorras. Viise juurdepääsetavuse saavutamiseks on mitmeid. Näiteks pimedate või vaegnägijate jaoks, kes navigeerivad veebilehel ilma hiireta, tuleks lehe disainimisel arvestada ekraanilugejaga. Kuna antud juhul kasutatakse toimingute tegemiseks klaviatuuri, siis veebileht peab toetama linkide vahel liikumist *tab*-nupuga ning soovitud lingi valimist *enter*-nupuga. Linkide vahel navigeerimisega esineb aga probleeme, kui vastav link ei juhata mitte kuskile (täidetud ei ole HTML elemendi *href*-atribuut) või kui ainus viis lingi valimiseks on liikuda hiirega selle kohale (kasutatud on HTML elemendi *onmouseover* atribuuti) [3].HTML elemendi probleemseid atribuute on illustreeritud joonisel 4.

---

<sup>1</sup> <https://www.w3.org/standards/webdesign/accessibility> (31.03.2018)



```
<a href="#" onmouseover="dropdownmenu()">Products</a>
```

Joonis 4. Vale atribuutide kasutus

Hea näide ekraanilugejast ja juurdepääsetavusest on kasutada *skip navigation* linki. Antud juhul on eesmärk suunata kasutaja koheselt veebilehe sisutekstini ilma, et ta peaks eelnevalt terve navigeerimisriba *tab* nupuga läbi käima. Eriti oluline on see inimestele, kel esineb lihashalvatus ning iga nupuvajutus on vaevaline. Vastava lahenduse jaoks tuleb lehe kõige esimeseks objektiks paigaldada HTML link, mis suunaks soovitud sisutekstini (HTML elemendi *href* atribuuti on lisatud soovitud sisuteksti id) [3]. Eelmainitud lahendus on kujutatud joonisel 5.

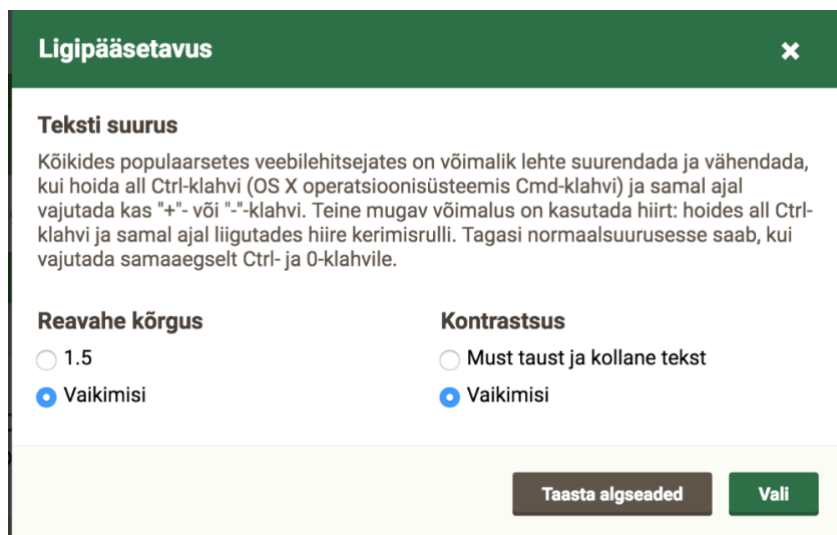
```
<body>
<a href="#maincontent">Skip to main content</a>
...
<main id="maincontent">
<h1>Heading</h1>
<p>This is the first paragraph</p>
```

Joonis 5. Õige atribuutide kasutus

Nagu joonisel 1 on välja toodud, siis mõningatel juhtudel võivad juurdepääsetavus ja kasutatavus kattuda. Näiteks veebileht, mis on disainitud nii, et seda saab kasutada ilma hiireta, võib puudeta inimese jaoks otseteede abil mingisuguste toimingute tegemist kiirendada. Lisaks on vastav funktsionaalsus juurdepääsetavuse seisukohast oluline inimesele, kellel esineb nägemispuue. Eesti.ee<sup>1</sup> pakub kasutajatele võimaluse vahetada reavahe kõrgust ja kontrastsust ning on ette kirjutanud selge juhise, kuidas veebilehitsejas teksti suurendada või vähendada (illustreeritud joonisel 6). Antud lahendus tuleb kasuks nii eakamatele kasutajatele, kelle silmanägemine on aastatega halvenenud kui ka nägemispuudega või värvipimedatele inimestele.

---

<sup>1</sup> <https://www.eesti.ee/et/index.html> (31.03.2018)



Joonis 6. Eesti.ee juurdepääsetavuse valikud

Kasutatavus ja juurdepääsetavus kattuvad ka navigeerimisriba disainimisel. Korrektselt kujundatud navigeerimisriba kiirendab soovitud lingi otsimist oluliselt nägemispuudega kasutaja puhul, kuid samas parandab lehekülje üleüldist loetavust ka puudeta inimese jaoks.

## 2.2 Kasutatavuse olulisus

Olles defineerinud kasutatavuse ja sellega seonduvad mõisted, on tähtis ka aru saada, miks tasub selle edendamisse aega ja raha investeerida. Definitsioonist tulenevalt on kasutatavuse kolm peamist ideed muuta veebikeskkond tulemuslikuks, tõhusaks ja rahuldust pakkuvaks. Arvestama peab aga asjaolu, et erinevad veebisaidid täidavad erinevaid eesmärke ning sellest tulenevalt on nende rõhuasetused ka erinevates kohtades.

Näiteks e-kaubandusega tegeleva ettevõtte (Amazon ja eBay) soov on kasutatavuse parendamisega kasvata külastajate arvu ning kasumit. Antud eesmärgi täitmiseks tuleb veebipoe külastus muuta nii ladusaks kui võimalik, sest kasutaja, kes e-poodi külastab, tahab soovitud toote minimaalse ajakuluga üles leida ning paari klikiga ära osta. Selle saavutamiseks on vaja esmalt optimeerida otsingut. Otsingu puhul on primaarseteks komponentideks otsingukasti silmapaistvus, laadimiskiirus, sisestatud tähtede abil sobiva toote ennustamine ning arusaadavad filtrid. Pärast toote leidmist on oluline, et selle eest saaks väheste sammudega ning turvaliselt maksta [4]. Amazoni lahendus ostuprotsessi kiirendamiseks on *1-click ordering*, mis tähendab, et kui kasutaja on oma andmed ühe korra sisestanud, siis edaspidi saab ta suvalise toote lehel viibides selle koheselt ka tellida.

Vahele jäetakse makseandmete kinnitamise faas<sup>1</sup>. eBay, kes toetab ka PayPal'i makseid, pakub sarnast funktsionaalsust PayPal'i *One Touch*'ga, mille puhul samas seadmes ja brauseris olles pole makseandmeid korduvalt üle kinnitada vaja<sup>2</sup>. Turvalisuse poolest on eBay tänu PayPal'le eelistatum. Kui Amazonist ostmiseks peab krediitkaardi andmed sisestama Amazoni profiili alla, siis eBay puhul saab makseviisina valida PayPal makse, mis lisaks e-poe enda turvalisusele tagab täiendava turvalisuse kihi<sup>3</sup>. Seega Amazon saaks parandada kasutatavusega seotud rahulolumomenti sellega, kui pakuks ka PayPal makseid.

Kui Amazon ja eBay on kasutatavuse poolest pigem positiivsed näited, siis negatiivne näide on ühest e-kaubandusega tegelevast ettevõttest, kes kaotas suurel hulgal potentsiaalseid kliente, sest ostuvormil olev nupp tekitas kasutajates segadust<sup>4</sup>. Probleem sündis erinevatest arusaamadest veebikeskkonna kasutatavuse osas. Kui disainerid olid arvamisel, et nende poolt välja töötatud lahendus muudab ostuprotsessi kasutaja jaoks lihtsamaks, siis reaalsus oli vastupidine. Nimelt nõudis veebileht ostu sooritamiseks registreerumist või sisse logimist. Disainerite nägemus oli, et kui klient omab veebilehel kontot, siis ei pea ta makseandmeid korduvalt sisestama. Kasutajad aga leidsid, et ettevõtte soovib koguda nende personaalandmeid. Olukord lõppes sellega, et "registreeri" nupp, mille tõttu paljud kliendid jätsid ostu sooritamise pooleli, asendati "jätk" nupuga, mis ei nõudnud kasutajakonto loomist ning mille tulemusena kasvas ettevõtte aastane kasum 300 miljoni dollari võtta.

Kuivõrd e-kaubandusega tegelevate ettevõtete jaoks on olulisel kohal külastuste arv ning kasum, siis avaliku sektori veebikeskkond rõhub teistele väärtustele. Esmatähtis on, et pakutav teenus oleks kasutajate jaoks ühtemoodi arusaadav<sup>5</sup>. Kui teenus on mõeldud kõigile vanuses 10-70, siis peab see oleme piisavalt lihtne, et nii koolilaps, keskealine kui ka pensionär oskaks sellega toime tulla. Võrdsuse tagamiseks peab veebileht olema

---

<sup>1</sup> <https://www.amazon.co.uk/gp/help/customer/display.html?nodeId=201889620> (31.03.2018)

<sup>2</sup> <https://www.paypal.com/us/webapps/mpp/one-touch-checkout> (31.03.2018)

<sup>3</sup> <http://www.washingtonpost.com/wp-dyn/content/article/2008/08/15/AR2008081503259.html> (31.03.2018)

<sup>4</sup> [https://articles.uie.com/three\\_hund\\_million\\_button/](https://articles.uie.com/three_hund_million_button/) (31.03.2018)

<sup>5</sup> <https://theknowledgeexchangeblog.com/2015/03/04/what-is-the-right-approach-to-public-service-website-design/> (31.03.2018)

ühthemoodi kättesaadav ka erinevates veebibrauserites ning seadmetes (arvuti, tahvelarvuti, nutitelefon)<sup>16</sup>. Lisaks on tähtis, et teenus ei oleks vigane<sup>16</sup>, sest tihtipeale töödeldakse avaliku sektori platvormidel delikaatseid isikuandmeid. Näiteks eesti.ee puhul, mis haldab Eesti kodaniku andmeid nii eluaseme, hariduse, perekonna, reisikindlustuse, sotsiaaltoetuste, õigusabi, juhilubade kui ka tervise kohta<sup>1</sup>, ei saa mitte mingil juhul lubada väärandmete salvestamist. Vastasel juhul kaob kodaniku usaldus nii riigiportaali kui ka terve riigi vastu ning langeb kasutaja rahulolu. Avaliku sektori veeb on erasektori veebist ka karmimalt reguleeritud<sup>16</sup>. Näiteks 2016. aasta Euroopa Parlamendi poolt koostatud juurdepääsetavuse direktiiv [5] on erasektori jaoks soovituslik, kuid avaliku sektori jaoks kohustuslik.

## **2.3 Kasutatavuse viis dimensiooni**

Nagu eelnevalt ilmnes, siis kasutatavusega tegelemine on oluline sõltumata sektorist, mille alla veebikeskkond kategoriseerub või ülesannetest, mis on veebilehele püstitatud. Järgnevalt tuuakse välja peamised dimensioonid hea kasutatavuse saavutamiseks.

### **2.3.1 Tulemuslikkus**

Esimene mõõde on tulemuslikkus, mille abil tuvastatakse kas ja millisel määral mõnda teenust on võimalik kasutada. Kui kasutajale antakse ette nimekiri tegevustest, mida on vaja veebikeskkonnas sooritada, siis tulemuslikkuse eesmärk on hinnata, kas tegevus viidi lõpuni ning kas sellega on jõutud soovitud tulemini. Kui kasutaja ei saanud planeeritud tegevust sooritada või oli sunnitud tegema seda poolikult, on tulemuslikkuse seisukohast tegu vigase süsteemiga [6]. Näiteks, kui Eesti eraisik, kel on mitu aktiivset pangakontot, ei saa Maksu- ja Tolliameti veebikeskkonnas tulu deklareerides süsteemi vea tõttu konto numbrit vahetada, siis tulemuslikkuse mõttes on tegu vigase teenusega. Olgugi, et maksuvaba miinimumi pealt kinni peetud summa jõudis deklaratsiooni kinnitamisel lõpptulemusena kasutaja pangaarvele, polnud tegu kontoga, kuhu kasutaja ülekannet reaalselt saada soovis. Ehk et teenus töötas poolikult.

---

<sup>1</sup> <https://www.eesti.ee/est/teenused/kodanik> (31.03.2018)

### 2.3.2 Tõhusus

Lisaks tulemuslikkusele on oluline ka tõhusus. Tõhusus erineb tulemuslikkusest selle poolest, et siinkohal hinnatakse pingutust ja aega, mis kulus konkreetse eesmärgi täitmiseks. Hea tava on, et süsteemi peamised protsessid on loogilised ning sisaldavad minimaalsel hulgal samme. Protsessi analüüsimiseks loendatakse tihtipeale hiireklikke ning nupuvajutusi, mis kulus kasutajal konkreetse ülesande täitmiseks [6].

Olulise panuse veebikeskkonna tõhususele annavad navigatsioonielemendid nagu menüüd, lingid või otseteed. Hästi disainitud navigatsiooniga süsteem peaks võimaldama toimingute tegemisel vähendada kasutaja pingutust. Õigete disainiotsuste tegemiseks on tähtis aru saada, kes moodustavad valdava osa süsteemi kasutajatest ning mis on nende eelistused. Näiteks eksperdi jaoks, kes tunneb süsteemi hästi, on otseteede olemasolu kindlasti hea täiendus, mis tõstab tööülesannete täitmise kiirust. Uute kasutajate jaoks võivad otseteed tekitada aga lisakeerukust ning nende seisukohalt rakenduse tõhusus langeb. Selleks, et välja selgitada, kumb kasutajagrupp on ülekaalus, tuleb mõõta aktiivsete ning uute kasutajate arvu. Kui uute kasutajate arv on aktiivsete kasutajate arvu kõrval marginaalne, võib eeldada, et domineeriv kasutaja on ekspert ning otseteede lisamine protsesside kiirendamiseks on õigustatud. Uute kasutajate ülekaalu korral tuleks aga peidetud funktsionaalsust vähendada ja panustada süsteemile, mille protsessid koosnevad küll rohkematest sammudest, kuid mille kasutamiseks ei pea omama lisateadmisi ning tarkvara saab hakata kohe kasutama [6].

Tõhususest rääkides on kriitilisel kohal ka aeg, mis kulub kasutajal rakenduse toimingute järgi ootamiseks. Ooteaeg on mõjutatud nii andmebaasipäringute kiirusest, äriloogika täitmiseks kuluvast ajast *back-end* koodis kui ka CSS või JS failide suurusest *front-end* koodis [7] [8]. Näiteks andmebaasipäringu optimeerimiseks tuleks sorteeritavatele väljadele lisada indeksid, mis võimaldavad antud väljade andmetele kiiret ligipääsu. Lisaks peaks vältima andmeliiasust. Ehk et optimaalne oleks pärida vaid andmeid, mida reaalselt ka kasutatakse [7]. Serveri kiirust läbi *back-end* koodi annab tõhustada, kui kasutada vahemälu. Näiteks andmebaasi päringud, mida rakendus kutsub sagedasti välja, kuid millega tagastatavad andmed ajas tihedasti ei muutu, oleks mõistlik tõsta vahemällu. Vahemälu võimaldab andmed kiiremini kätte saada ning samas vähendab ka andmebaasi koormust [7]. Rakenduse kiirust aitab parandada ka CSS ning JS failide minimeerimine. Antud protsessi puhul eemaldatakse koodist ebavajalikud tühikud, reavahed ja

kommentaariid ning vähendatakse seeläbi faili suurust. Kuivõrd failide suurus väheneb, väheneb ka veebilehe suurus ning kasvab lehe laadimiskiirus. Lisaks ooteaja vähenemisele mõjutab hea laadimiskiirus ka SEO'd (ingl *search engine optimization*) ehk otsingumootori optimeerimist. Näiteks Google'i otsing annab tulemuste järjekorras kõrgema koha veebilehtedele, millel on kiirem laadimiskiirus [8].

Kasutatavuse ekspert Jakob Nielsen on öelnud, et võttes arvesse inimese iseloomuomadusi, kehtivad ooteajale kolm reeglit. Kui see on kiirem kui 1/10 sekund, siis seda ei märgata. Kui see on 1/10 sekundi ning sekundi vahepeal, on tegu sujuva rakendusega, kuid tunda on, et tuleb oodata. 1 kuni 10 sekundit on ootamise piir [9].

Seega ideaalis peaks ooteaja optimeerima alla 1 sekundi. Kõik, mis kestab rohkem kui 10 sekundit, on liiga kaua ning paneb mõtlema, kas rakendus on kokku jooksnud. Kui taoline situatsioon esineb, on mõistlik täiendada süsteemi graafilise mõõdikuga, mis näitab kasutajale, mitu protsenti toimingule kuluvatest operatsioonidest on rakenduse poolt sooritatud. Taoline mõõdik annab kasutajale märku, et süsteem veel töötab ning näitab, kui kaua tuleb hinnanguliselt oodata [8].

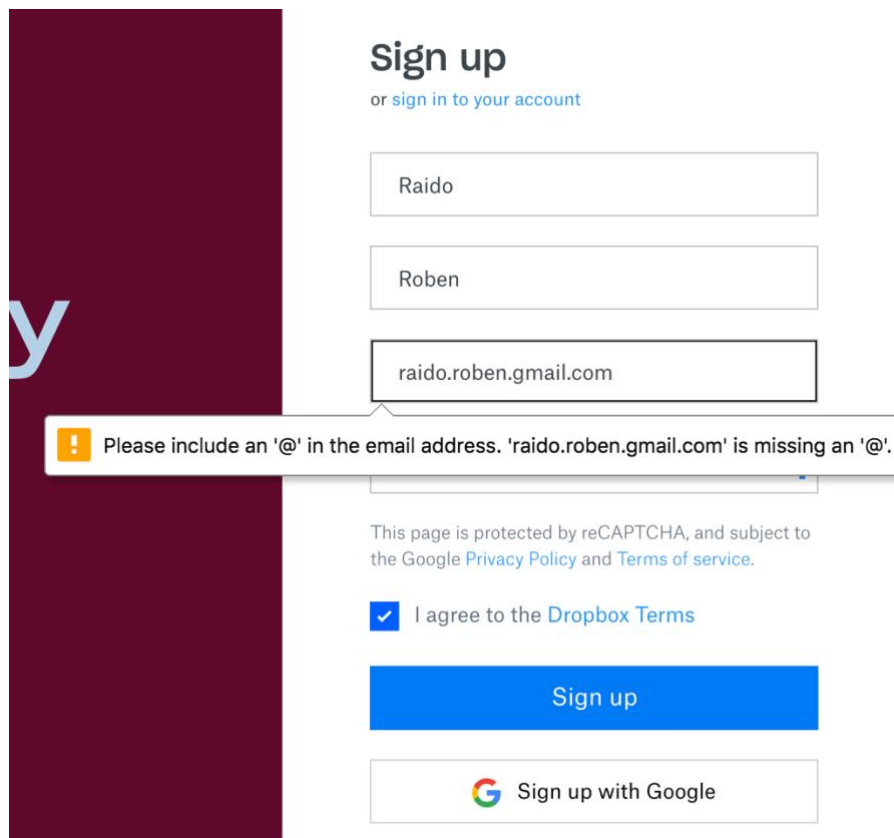
### **2.3.3 Veatuvastus**

Kahtlemata on iga ettevõtte või asutuse soov luua rakendus, mis töötab veatult. Reaalsuses on seda raske teostada, sest kõikide veaolukordade ette ennustamine on võimatu. Seega süsteemi arendamisel peab arvesse võtma, et vigu esineb ning töötada tuleb selle kallal, et süsteem edastaks veateateid, millest oleks kasu. Sellega seoses tuleb mängu veatuvastuse mõõdik, mis näitab rakenduse suutlikkust ennetada kasutaja poolt põhjustatud vigu ning vigade esinemisel kasutajat abistada [6]. Vastavalt Jakob Nielsen soovitustele<sup>1</sup> annab korrektne veateade selgelt märku, et midagi läks valesti. Seega kõige hullem on veateade, mida ei eksisteeri ning kasutaja ei saagi kunagi teada, mida ta tegi valesti. Näiteks kui sotsiaalmeedia rakenduses läheb kirjutatud sõnum süsteemi vea tõttu kaduma ning süsteem ei edasta selle kohta vastavat teadet, ei saa kasutaja kunagi teada, et sõnum on saatmata ning jääb ekslikult ootama vastaspoole vastust. Lisaks peaks veateade olema inimloetav ning viisakas. Sõnum, mis koosneb mittemidagiütlevast numbrilisest koodist või süüdistab kasutajat tema rumaluses, on halb praktika. Veateate

---

<sup>1</sup> <https://www.nngroup.com/articles/error-message-guidelines/> (14.04.2018)

koostamisel on olulisel kohal ka täpsus. Hea näide on Dropboxi veeb<sup>1</sup>, kus uue kasutaja registreerimisel antakse andmete sisestamisel emaili lahtri juures täpselt teada, et “@” märk on puudu (näidatud joonisel 7). Tunduvalt halvem lahendus oleks registreerimisvormi lõppu kirjutada, et tegu on süntaksi veaga. Ühest küljest ei viita taoline teade lahtrile, mille täitmine ebaõnnestus ning teisest küljest ei anna edasi infot, mis antud lahtri juures täpselt valesti on.



The image shows a screenshot of the Dropbox sign-up page. On the left, there is a dark red vertical bar with a white 'y' logo. The main content area is white and contains the following elements:

- Sign up** header, with a link "or sign in to your account" below it.
- Three input fields: the first contains "Raido", the second "Roben", and the third "raido.roben.gmail.com".
- A yellow error message box with an exclamation mark icon: "Please include an '@' in the email address. 'raido.roben.gmail.com' is missing an '@'."
- Text: "This page is protected by reCAPTCHA, and subject to the Google Privacy Policy and Terms of service."
- A checked checkbox: "I agree to the Dropbox Terms".
- A blue "Sign up" button.
- A "Sign up with Google" button with the Google logo.

Joonis 7. Dropboxi sisselogimise veateade

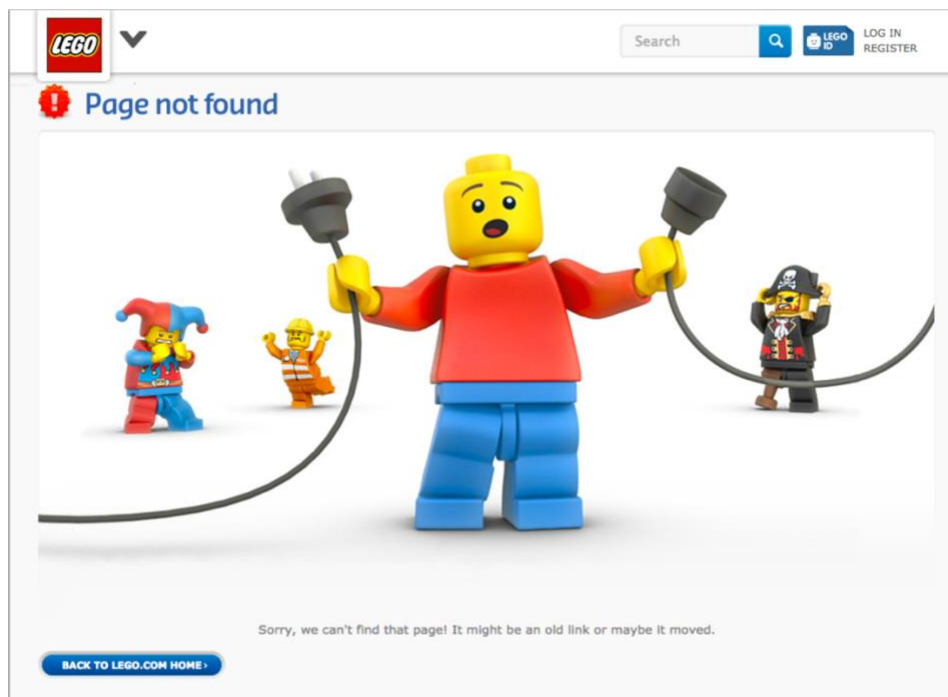
Samuti peaks veateade olema konstruktiivne. Kui näiteks e-kaupluses on mingisugune toode otsas, siis “kaup on läbi müüdud” kirja asemel võiks olla teade selle kohta, millal on kaup uuesti saadaval.

Veateadete alla kuulub ka veakood 404, mis suunab tühjale lehele ning viitab sellele, et otsitakse veebilehte, mida ei eksisteeri. Seetõttu oleks mõistlik kohandada antud leht selliselt, et see suunaks mõnele alternatiivsele lehele või veebikeskkonna kodulehele.

---

<sup>1</sup> <https://www.dropbox.com/login> (14.04.2018)

Ilusa 404 lehe lahenduse pakub välja Lego<sup>1</sup> (joonis 8), kes lisaks humoorikale pildile ning kodulehele suunavale nupule on lisanud ka otsingukasti täiendava abi pakkumiseks.



Joonis 8. Lego 404 leht

### 2.3.4 Kaasavus

Rakendus on kaasahaarav, kui see on ilusa väljanägemisega. Sellega seoses ei saa kaasavusest rääkides jätta vahele esteetika kasutatavuse efekti (ingl *aesthetics-usability effect*), mis tähendab, et kasutaja on valmis tolereerima kasutatavuse probleeme, kui toote visuaal on tema jaoks atraktiivne [6].

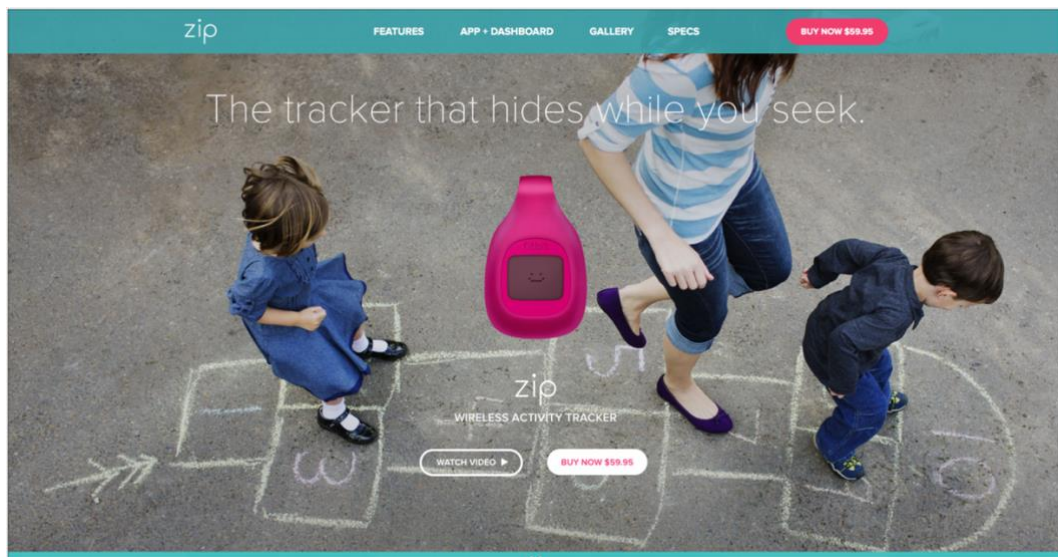
Kate Meyer Nielsen Norman Group'st kirjeldab kasutatavuse testi FitBit'i (aktiivsusmonitore müüv ettevõtte) veebilehel (näidatud joonisel 9), kus vaatamata olulistele kasutatavuse probleemidele andis testis osalenud kasutaja veebilehele kõrge hinde<sup>2</sup>. Täpsemalt esines veebilehel mitmeid vigu keskkonnaga suhtlemises ning navigeerimisel, mille tõttu kulus testis määratud ülesannete täitmisele rohkem aega ja vaeva, kui oleks pidanud. Testijärgses tagasisides hindas kasutaja veebilehte kõrge hindega ning lisas, et värvid ja pildid, mida kujunduses kasutati, on rahustavad ja

<sup>1</sup> <https://www.creativebloq.com/web-design/best-404-pages-812505> (14.04.2018)

<sup>2</sup> <https://www.nngroup.com/articles/aesthetic-usability-effect/> (14.04.2018)



meeldivad. Seega positiivne emotsioon, mida veebilehe välimus tekitab, aitas kasutatavuse probleemid testis osalenu jaoks ära peita.



Joonis 9. FitBit'i koduleht

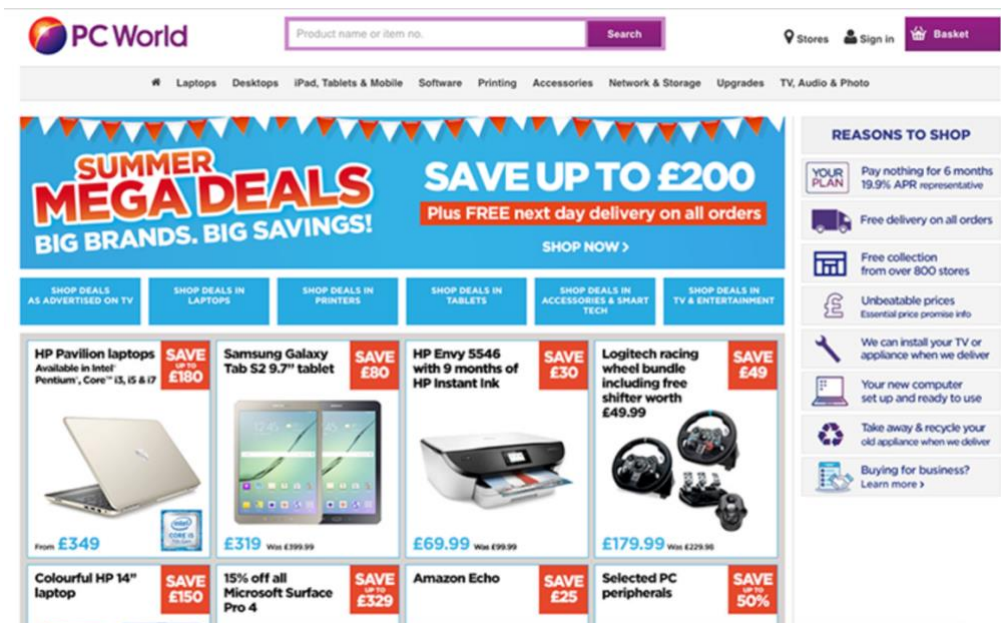
Esteetika kasutatavuse efekt on oluline e-kaubandusega tegelevate ettevõtete jaoks, sest ilus välimus tekitab kasutajas huvi veebilehel kauem aega veeta ning seda rohkem avastada. Iga lisanduv sekund, mille kasutaja veebipoe külastusele kulutab, suurendab aga tõenäosust, et külastus lõpeb ostuga [6].

Olgugi, et graafiline disain on kaasavuse puhul kõige olulisem faktor, kuuluvad siia alla ka mõned muud elemendid. Näiteks teksti loetavus ning moodus, kuidas informatsioon on liigendatud, võib muuta kasutaja hinnangut konkreetse rakenduse suhtes.

Halb näide liigendusest on PC World koduleht (näidatud joonisel 10), mille kohta läbi viidud küsitlusest ilmnas, et poolte kasutajate arvates on kujundus paigutatud liiga tihedalt<sup>1</sup>.

---

<sup>1</sup> <https://conversion.com/blog/aesthetic-usability-effect/> (14.04.2018)



Joonis 10. PC World koduleht

Võrreldes Apple'i kodulehega (joonis 11), mida võib disainilt lugeda üheks maailma parimaks<sup>21</sup>, on PC World koduleht kordades kirjum ning sealset informatsiooni on raske hoomata. Põhiprobleemideks on tühjade alade nappus ning kokkupressitud tekst**Error!**  
**Bookmark not defined..**



Joonis 11. Apple avaleht

Lisaks võib kliendi rahulolu mõjutada stiil, kuidas rakendus kasutajaga suhtleb. Süsteem, millele on lisatud animatsioone ning mis on stiili poolest mänguline, jätab kasutajale sügavama emotsiooni, kui mõni igava ja elutu väljanägemisega veebikeskkond [6].

### **2.3.5 Kergesti õpitavus**

Iga kord, kui kasutaja puutub kokku uue tootega, hakkab ta seda kasutama kas instinktiivselt, tuginedes kasutusmuustritele, mis on jäänud alateadvusesse teiste rakenduste kasutamisest, või uuesti õppides, tutvudes konkreetse toote kasutusjuhendiga. See, kui palju kasutaja peab juurde õppima, sõltub nii tootest kui ka kasutajast endast. Toote puhul mõjutab õpitavust see, kui keerulised on selle protsessid ning kas turul on ka teisi tooteid, millele konkreetse toote kasutusloogika sarnaneb. Kasutaja seisukohast sõltub õpitavus sellest, kas ta on millegi sarnasega juba varasemalt kokku puutunud ehk et mis on tema oskustase (algaja või ekspert). Kui Apple tuli 2007. aastal välja esimese iPhone'iga<sup>1</sup>, oli tegu tehnoloogiaga (puutetundlik ekraan ning äpid), millega valdav osa inimestest polnud varasemalt kokku puutunud ning telefoni kasutamiseks tuli juurde koguda uusi teadmisi. Samas kasutaja, kes soetas iPhone'i eelmisel aastal, ei pidanud põhifunktsionaalsuste kasutamiseks tõenäoliselt juurde õppima midagi, sest nutitelefonide baasloogika on olnud läbi aastate üldjoontes sama ning kasutusperiood pikk. Seega kergesti õpitavus põhineb teadmistel, mis on kasutajal juba eelnevalt olemas.

Veebilehe näitel tähendab kergesti õpitavus seda, et teatud komponendid peavad paiknema tuttavates kohtades. Näiteks otsingukast üleval paremas nurgas ning navigeerimisriba ülemises või vasakus ääres on mustrid, millega on harjunud igäüks. Lisaks on harjumuspäraseks saanud flopiketta ikoon tegevuste salvestamiseks või kolme paralleelse joone ikoon menüü avamiseks [10]. Üldine põhimõte on vältida ebaselgeid olukordi, kus kasutaja ei saa enam aru, kuhu hiirega liikuda ning mida klikkida. See aga ei tähenda, et arendama peaks igavaid ja standardseid veebilehti. Vastupidi, mõistlik on ehitada rakendus, mis eristub massidest. Teistsugused värvid, animatsioonid, pildid või kirjastiilid aitavad kaasa omanäolisele disainile [6].

## **2.4 Vahekord kasutatavuse dimensioonide vahel**

Olles tutvunud 5 kasutatavuse dimensiooniga, võib öelda, et iga dimensiooni olulisus on suhteline ning sõltub nii arendatava veebilehe eesmärkidest kui ka kasutajatest, kellele see on suunatud.

---

<sup>1</sup> [https://www.gsmarena.com/apple\\_iphone-1827.php](https://www.gsmarena.com/apple_iphone-1827.php) (14.04.2018)

Näiteks sotsiaalmeedia platvormi (Facebook, Instagram, Twitter) puhul on tähtsal kohal kaasavus ning õpitavus<sup>1</sup>. Ühest küljest peab rakendus olema atraktiivne ning äratama kasutajates esmase huvi sellel viibida, kuid teisest küljest peab see pakkuma tegevusi, millel on tähendust ning mis tekitavad vajadust külastada rakendust uuesti. Näiteks Instagrami peamine eesmärk on, et kasutajad saaksid end väljendada läbi fotode. Twitteri eesmärk on suhteliselt sarnane, kuid väljendusviisiks on kirjalikud postitused. Mõlemad, nii Instagram kui Twitter, võimaldavad lisaks enda poolt lisatud sisule jälgida ka sõprade või muude huvipakkuvate kasutajate postitusi. Samuti ei tohi sotsiaalmeedia rakenduste kasutamine olla ülemäära keeruline. Kuna väga paljud Instagrami ja Twitteri kasutajad on kokku puutunud ka Facebook'ga, mille kujundus on üles ehitatud sama loogika alusel, siis on neid suhteliselt lihtne hakata koheselt kasutama.

E-kaubanduse puhul on esmatähtsad tõhusus ja tulemuslikkus. Kuna 2/3 e-poe külastustest on eesmärgiga osta kindel toode<sup>2</sup>, on kriitiline roll veebilehe otsingul, mis võimaldab minimaalse vaevaga kiiresti tooteid otsida. Samuti tuleb tagada tõhus ostuprotsess, mis ei koosne liiga paljudest või keerulistest sammudest ning aitab kliendil jõuda soovitud tulemuseni – soovitud kauba tellimiseni.

Nii sotsiaalmeedia kui e-kaubanduse veebilehed sisaldavad registreerimisvormi. Registreerimisvormidel, mis nõuavad kasutaja ning rakenduse vahelist suhtlust, on esikohal veatuvastus. Oluline on kindlustada valiidne registreering ning vigade esinemisel näidata võimalikult täpseid veateateid (nt sisendvälja valideerimine). Tähtis on ka kergesti õpitavus, mis tähendab, et kasutaja ei pea rakenduse kasutamiseks läbima koolitust, lugema manuaali või helistama abiliinile. Tõhususe aspekt ilmneb sellega, et vormi täitmisele kuluv aeg ei tohiks olla pikem ajast, mis kuluks vormi käsitsi täitmisele [6].

Eelmainitud näited illustreerivad seda, kuidas viit karakteristikut hinnates saab leida suuna rakenduse disainile.

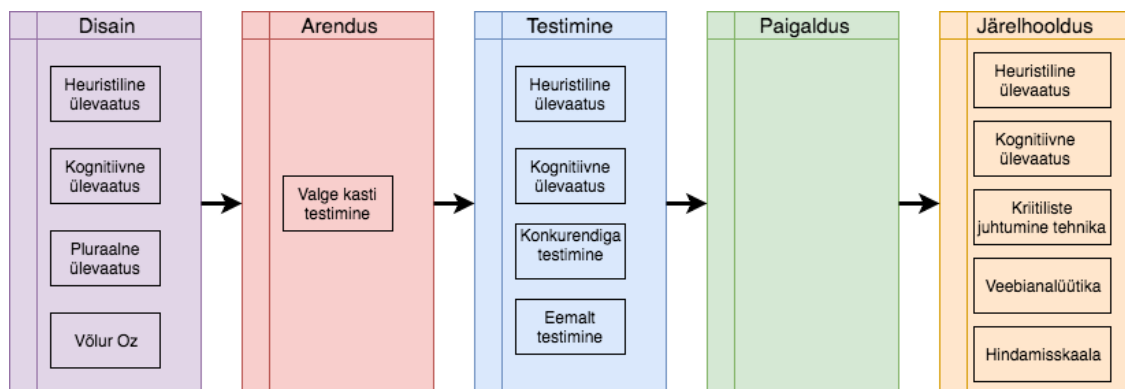
---

<sup>1</sup> <https://www.smashingmagazine.com/2009/07/social-network-design-examples-and-best-practices/> (21.04.2018)

<sup>2</sup> <https://www.nngroup.com/articles/ecommerce-improvements/> (21.04.2018)

### 3 Kasutatavuse testimise meetodid erinevates arendusetappides

Võttes aluseks klassikalise tarkvaraarenduse mudeli jaotatakse arendustsükkel 7 faasi vahel: planeerimine, analüüs, disain, arendus, testimine, paigaldus ja järelhooldus [11]. Samu faase sisaldavad ka kolm enimkasutatavat tarkvaraarenduse metodoloogiat: kose, iteratiivne ja agiilne mudel<sup>1</sup>. Antud peatüki eesmärk on välja selgitada, milline kasutatavuse testimise meetod millisesse faasi kuulub. Kuna kasutatavuse hindamise meetodeid eksisteerib palju ning osad neist on sisult väga sarnased, tuuakse käesolevas töös välja vaid põhilised. Testimise meetodid erinevates arendustsükli faasides on illustreeritud joonisel 12. Tabelis alustatakse disainifaasist, kuivõrd enne seda kasutatavuse testimist ei toimu.



Joonis 12. Kasutatavuse testimise meetodid erinevates arendusetappides

#### 3.1 Ülevaatusmeetodid

Kasutatavuse testimise meetodid on jagatud nelja kategooriasse. Neist esimene on ülevaatus. Antud kategooriasse kaasatakse eksperdid ning arendusmeeskonna liikmed, kes pakuvad kasutatavuse alast lisateavet süsteemi disainimisel. Ülevaatusel on kolme liiki: pluraalne, heuristiline ja kognitiivne.

---

<sup>1</sup> <http://www.agilistapm.com/differences-between-waterfall-iterative-waterfall-scrum-and-lean-software-development-in-pictures/> (21.04.2018)

### **3.1.1 Pluraalne ülevaatus**

Pluraalset ülevaatus kasutatakse esmase disaini hindamiseks. Antud ülevaatuses kasutatakse paberprototüüpe, mis kujutavad arendatava rakenduse kasutajaliidest. Testis osalevad arendusmeeskonna liikmed, kes lahendavad ettekirjutatud ülesandeid. Pluraalne ülevaatus võimaldab saada kohest tagasisidet ning pidada vahetult diskussioone kujundusprobleemide ning võimalike lahenduste üle. Samuti kogutakse sellega enne kulukate disainistrateegiate rakendamist esmaseid andmeid süsteemi suutlikkuse ja arusaadavuse kohta ning säästetakse seeläbi raha. Negatiivse poole pealt on pluraalne ülevaatus ebatäielik, sest nii varajases arendusetapis ei pruugi süsteemi disain olla lõplik ning kõikvõimalikke testsituatsioone ei saa läbi mängida. Pluraalne ülevaatus liigitub disainifaasi [12].

### **3.1.2 Heuristiline ülevaatus**

Heuristilises ülevaatuses osalevad üks või mitu kasutatavuse eksperti, kes lähtuvad süsteemi hindamisel parimatest praktikatest (heuristikatest) ning püüavad tuvastada olukordi, kus süsteem antud põhimõtetega ei ühti. Heuristiline hindamine on odavam kui teised ülevaatusmeetodid, sest osalejate arv on väike ning täpset testimisplaani pole vaja paika panna. Heuristilise ülevaatuses suur eelis on ka see, et seda saab rakendada paljudes arendustsükli faasides. Antud meetodi puuduseks on, et konkreetse või sarnase rakenduse hingeelu tundvat eksperti ei pruugi olla lihtne leida. Kuna ülevaatuses on vähe osalejaid, on avastatud vigade hulk tõenäoliselt väiksem kui suure testgrupiga testides. Lisaks, kuivõrd ekspert pole lõppkasutaja, võib heuristiline ülevaatus viia valede probleemide lahendamiseni, mis halvemal juhul muudab rakenduse kasutatavuse kehvemaks. Heuristilist ülevaatus saab kasutada nii disaini-, testimis- kui ka järelhooldusfaasis [12].

### **3.1.3 Kognitiivne ülevaatus**

Kognitiivses ülevaatuses osalevad üks või mitu hindajat, kes erinevaid stsenaariume läbi töötades püüavad jõuda samade küsimusteni, milleni jõuaksid lõppkasutajad. Ehk et hindamisel lähtutakse lõppkasutaja perspektiivist. Antud meetodi fookus on tuvastada süsteemi õpitavus uue või mitteaktiivse kasutaja jaoks. Kuivõrd kognitiivse ülevaatuses eesmärk on mõista uue kasutaja esmakogemust, on ajalooliselt rakendatud meetodit ka näiteks pangaautomaatide või interaktiivsete muuseumi eksponaatide peal, kus kasutajal puudub eelnev ettevalmistus [12].

Kognitiivse ülevaatus tugevuseks on, et üksikasjalikult lähtutakse ülesannetest, millega kasutaja reaalsuses kokku puutub ning seeläbi pakutakse soovitusi süsteemi õpitavuse edendamiseks. Samuti on meetodi kasutamine kiire ja odav ning seda saab rakendada mitmes arendustsükli faasis. Ülevaatus nõrkuseks on, et see sõltub hindaja oskustest ning ei pruugi peegeldada reaalseid tulemusi. Lisaks võib olla meetod kohati pinnapealne, kuivõrd testis keskendutakse enamasti ekraanilt nähtavatele sõnadele ja graafikale. Kognitiivset ülevaatus saab läbi viia disaini-, testimis- ja järelhooldusfaasis [12].

### **3.2 Arendaja test (valge kasti testimine)**

Olgugi, et arendajad võivad osaleda nii pluraalses kui heuristilises ülevaatuses, on olemas eraldi testimismeetod, mis on konkreetselt koodi kirjutaja spetsiifiline. Tegu on valge kasti testimisega, mille puhul testitava objekti struktuur ja disain on testis osalejale teada. Kuna testis osaleja määrab testi sisendid ja väljundid, peab ta tundma programmikoodi – seega tavaliselt on testija ka koodi autor. Valge kasti testimise eesmärk on detailselt hinnata koodi funktsioneerimist. Seos kasutatavusega ilmneb sellest, et funktsionaalsuse valideerimisega testitakse ka rakenduse kasutatavuse karakteristikuid - tõhusust ja tulemuslikkust. Näiteks veebilehe puhul on valge kasti testimise tööpõhimõtte määrata kõik lubatud ja mitte lubatud sisendid ning katsetada, kas väljundid vastavad oodatud väärtustele. Valge kasti testide alla kuuluvad ühiktestid, integratsioonitestid ning süsteemi testid [13].

Valge kasti testimise eeliseks on selle teostamine arendustsükli varajases faasis. Samuti on tugevuseks testimise põhjalikkus. Kuivõrd testi sooritaja peab süvitsi tundma koodiloogikat, oskab ta läbi katsetada paljusid testolukordi. Kuna valge kasti testimises osaleb inimene, kes koodi tunneb, võib antud meetodi peamiseks nõrkuseks pidada seda, et mingist hetkest ei oska testija olemasolevast koodist kaugemale mõelda ning ei pruugi ette näha harva esinevaid teststsenaariume. Valge kasti meetodit rakendatakse arendusfaasis [13].

### **3.3 Lõppkasutajaga testimise meetodid**

Järgmine kategooria on testimine lõppkasutajatega. Seega meetodid, mis siia kuuluvad, kaasavad inimesi, kes on potentsiaalsed või reaalsed süsteemi kasutajad. Testimist lõppkasutajatega viiakse läbi nii arendustsükli disainifaasis, leidmaks kitsaskohti juba

varakult kui ka testimisfaasis, et valideerida rakenduse toimimist mõne kindla ülesande sooritamisel. Vaatluse all olevasse kategooriasse kuuluvad konkurendiga testimine, eemalt testimine ja võlur Oz.

### **3.3.1 Konkurendiga testimine**

Konkurendiga testimise eesmärk on leida, kui hästi toimib arendatud süsteem võrreldes konkurendi omaga. Testimine võib olla holistiline, järjestades veebikeskkondi üldiste kasutatavuse mõõdikute alusel või spetsiifiline, keskendudes süvitsi mõnele funktsioonile, disainielemendile või muule võtmetähtsusega aspektile. Testimise ajal jälgitakse hoolikalt, kuidas kasutaja teostab toiminguid, mis erinevates veebikeskkondades sarnanevad ning tegevusi, mis sisu, funktsionaalsuse või disaini poolest varieeruvad. Pärast iga alternatiivse rakenduse testimist palutakse testis osalejalt kirjeldada, mis oli antud süsteemi plussid ja miinused ning võrrelda seda teiste rakendustega. Hea praktika on võtta skoopi kaks kuni neli süsteemi, sest muidu võib muutuda test liiga kulukaks või kasvada analüüsitavatelt andmetelt üle pea [14].

Konkurendiga testimine aitab maandada riske, sest mõne uudse elemendi toimimise edukust saab hinnata konkurendi rakenduselt enne, kui lisada see element enda rakendusele. Samas võib konkurendi süsteemist avastada elemente, mille peale poldud ise varem tulnud, kuid mis näitavad positiivseid testitulemusi. Lisades vastava elemendi ka enda keskkonda, aitab meetod kasvatada arendatava süsteemi väärtust. Konkurendiga testimist teostatakse arendustsükli testfaasis [14].

### **3.3.2 Eemalt testimine**

Eemalt testimise all mõistetakse meetodit, kus hindaja ning testis osaleja paiknevad erinevates asukohtades. Antud testi viiakse läbi kas modereeritud või modereerimata kujul [14].

Modereeritud variandi puhul osalevad testis kasutaja ja hindaja, kelle vahel toimub reaajas suhtlus ning ekraani jagamine. Hindaja ülesandeks on kätte jagada ülesanded, esitada küsimusi ning vajadusel suunata kasutaja õigele teele. Kasutaja ülesandeks on anda tagasisidet. Modereerimata meetodi puhul osaleb testis ainult kasutaja. Kasutajale antakse ette nimekiri eelnevalt kirja pandud ülesannetest, mille alusel tuleb tal need iseseisvalt lahendada. Kasutaja tegevused ning tagasiside salvestatakse. Eemalt testimise



eelisteks on, et see ei ole asukohast sõltuv ning seda saab kergesti rakendada suuremale inimgrupile .

Eemalt testimise hüved on peamiselt seotud testis osalevate osapoolte asukohtadega. Tänu võimalusele sooritada testi kodustes tingimustes, saab seda rakendada suurele inimgrupile ning kaasata testis osalejaid kõikjalt üle maakera. Kuna suhtlus ja jälgimine toimub interneti vahendusel, saavad ka testi hindajad jääda enda asukohta. Lisaks, kuivõrd testi sooritatakse isiklikest masinatest, saab kerge vaevaga kaasata ka puuetega inimesi, sest täiendavat riistvara või tarkvara pole vaja üles seadistada. Antud meetodi puuduseks on kõikvõimalikud tehnilised tõrked, mis testi sooritamise käigus võivad esineda. Samuti puudub võimalus hinnata kasutaja kehakeelt või muid kaudseid suhtlussignaale, sest tavapraktikas privaatsuse huvides veebikaameraid ei kasutata. Eemalt testimist kasutatakse arendustsükli testimisfaasis [14].

### **3.3.3 Võlur Oz**

Võlur Oz'i testimismetoodika puhul hinnatakse kasutaja toimetulekut rakenduse kasutamisega, kui süsteemi vastuseid simuleerib võõras inimene või inimeste grupp. Antud tehnikat rakendatakse tihtipeale süsteemides, kus kasutatakse häälkäsklusi ja -vastuseid [14].

Võlur Oz'i abil on võimalik testida tuleviku tehnoloogiaid ilma hinnalise prototüübi ehituseta. Positiivne on ka, et sisse viidud muudatusi saab kohe testida. Näiteks muudatused sõnastuses või toimingute järjekorras. Samuti aitab meetod koguda väärtuslikku teavet inimese ja masina vahelise kahekõne kohta. Meetodi peamised puudused tulenevad süsteemi kehastavast inimesest. Esiteks kulutatakse Oz'i simulatsioonides märkimisväärselt aega ja raha süsteemi esindava inimese treenimiseks, et tema vastused mõjuksid kasutajale usutavalt. Lisaks, kuivõrd inimene pole programm, millele kirjutatakse kindlad reeglid ette, on Oz'i puhul keeruline pakkuda järjepidevaid vastuseid terve testsessiooni vältel. Vastavat meetodit kasutatakse disainifaasis [14].

## **3.4 Live süsteemi testimise meetodid**

Neljandas kategoorias tuuakse välja kasutatavuse meetodid, mida rakendatakse pärast süsteemi paigaldamist *live* keskkonda reaalseste kasutajatega seotud andmete kogumiseks.

Siia kuuluvad kriitiliste juhtumite tehnika, veebianalüütika ning hindamisskaalad, mis kõik kuuluvad järelhoolduse faasi.

### **3.4.1 Kriitiliste juhtumite tehnika**

Kriitiliste juhtumite tehnika puhul kogutakse kasutatavusega seotud analüütikat läbi veebikeskkonnas registreeritud juhtumite. Juhtumeid registreerivad süsteemi kasutajad, kes teavitavad veebilehe haldajat süsteemi puudutavatest kitsaskohtadest või teevad ettepanekuid mingisuguste toimingute parendamiseks. Seega inimesed, kes hakkavad vastavaid kitsaskohti uurima, toetuvad dokumenteeritud sündmustele või telefonikõnedele, mis juhtumit kirjeldavad. Selleks, et saada probleemile jälile, vaadeldakse kas ühte konkreetset juhtumit või mitut juhtumit korraga. Praktika on näidanud, et tihtipeale viitavad väiksema ajaakna sees registreeritud juhtumid samale probleemile. Meetodi eesmärk on avastada võimalikult kiiresti veallikas ning säästa seeläbi potentsiaalset kaotust kasutajate arvus, rahas või andmetes [14].

Antud tehnika on kasulik, kuna võimaldab efektiivselt esile tuua suure kriitilisusega situatsioone, mille lahendamine aitab vältida mastaapseid kaotusi. Samuti võidakse sellega jälile jõuda haruldastele probleemidele, mida teised meetodid tuvastada ei suuda. Tänu süvauuringule pakub kriitiliste juhtumite tehnika ka head lahendust probleemidele, mille põhjus ja raskusaste on teadmata. Kuna esimesena tegeletakse prioriteetsemate juhtumitega, siis ei pruugi kõikide probleemide lahendamiseks ressursi jätkuda ning kasutaja seisukohalt on see pigem negatiivne. Kuivõrd kõikide vigade üle tavakasutajad ei kurda, on antud meetod toimingute algusest lõpuni analüüsimiseks ebapiisav. Lisaks ei pruugi vastav metoodika iga kord lahenduseni viia, sest uurimise käigus võib vaja minna andmeid, mis on kogutud ammu ajast, kuid on serveri mälu vabastamiseks juba kustutatud [14].

### **3.4.2 Veebianalüütika**

Veebianalüütika eesmärk on seadistada kasutaja käitumismustrite põhjal mõõdikud, millest võiks olla tulu veebilehe edendamiseks. Mõõdikud tuginevad andmetel, mis on kogutud automaatselt kõikidest veebikeskkonna külastustest. Näiteks kogutakse andmeid veebilehe küllastajate arvu, nende päritolu, populaarseimate rakenduste ning klikitavate linkide kohta [14].

Veebianalüütika jaguneb kaheks: *off-site* ja *on-site* analüüs. *Off-site* analüütika viitab andmetele, mis on konkreetse veebilehe välised. Siia kuuluvad uuringud süsteemi potentsiaalsete kasutajate, kommentaaride ning populaarsuse kohta. *Off-site* analüüs tugineb andmetel, mis on kogutud üle kogu interneti. *On-site* analüütika käsitleb andmeid, mis kogutakse konkreetse veebilehe külastuse ajal. Sealhulgas andmeid, mis ajendavad või tõrjuvad kasutajat mõnda soovitud toimingut tegema. Näiteks, kas veebireklaam e-kaubanduses toob kaasa tellimuste arvu suurenemise või mitte. Tüüpiliselt võrdleb *on-site* analüütika kasutaja käitumismustreid peamiste tulemuslikkuse näitajatega: tellimused, allalaadimised, registreeringud või muud tegevused, mis täidavad ettevõtte eesmärgi. Vastavad näitajad on sisendiks ettevõtte turundusstrateegia kujundamisele [14].

Veebianalüütika on oluline nii veebihaldajale kui ka kasutatavuse ekspertidele, saamaks laiaulatuslikku ülevaadet kasutajate käitumisest ning mõistmaks, mida tuleb ette võtta veebikeskkonna kasutatavuse parandamiseks. Samas ei oska see vastata küsimusele, mis kasutajaid motiveerib või mis on kasutaja põhivajadused. Näiteks kogutud analüütika alusel võib ilmned, millises tellimuse vormistamise faasis kasutaja süsteemi hülgab, kuid ei osata seletada, millest konkreetselt on see tingitud [14].

### **3.4.3 Hindamiskaala**

Veebilehe rahulolu ning lihtsuse hindamiseks rakendavad kasutatavuse eksperdid tihtipeale hindamiskaalasid. Hindamiskaala koosneb vastusevariantidest, mis eristuvad naabervalikust ning paiknevad kindlas järjestuses. Skaalade genereerimisel lähtutakse peamiselt selle suurusest, siltidest ning tüübist [14].

Skaala suuruse all peetakse silmas vastusevariantide hulka. Uuringud näitavad, et optimaalseim arv on 5-9, andes mõõdetavast objektist piisavalt hea ülevaate. Samas ei pea kasutaja valikutega tutvumiseks liiga palju aega kulutama. Skaala suuruse juures on veel oluline, kas sellel on paaris- või paaritu arv vastusevariante. Kui paaritu arv jätab võimaluse neutraalseks, keskmiseks vastuseks, siis paarisarv sunnib kasutaja tegema kas negatiivse või positiivse valiku [14].

Silte võib skaalal olla nii numbri kui tekstina. Samas võib mõne vastusevariandi puhul silt puududa. Sellisel juhul on sildid tavaliselt skaala otstes. Skaalal olevad numbrid võivad olla jaotatud regulaarse intervalli järgi (nt 1-5), kuid võivad paikneda ka

ebatraditsiooniliselt, mõjutades seeläbi vastajate tulemusi. Näiteks osad kasutajad väldivad negatiivsete arvude valimist [14].

Skaala tüüpidest on üks populaarsemaid Likert'i skaala, kus vastusevariandid paiknevad vasakult paremale alustades negatiivseimast ning lõpetades positiivseimaga (illustreeritud joonisel 13).

How would you rate the overall quality of xyz brand?				
1 - Poor	2 - Fair	3 - Good	4 - Very Good	5 - Excellent
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Joonis 13. Likert'i skaala

Iga vastusevariandi juures on number ning tekstiline silt, mis kirjeldab vastusevariandi tähendust. Likert'i skaalast mõnevõrra teistsugune on semantiline diferentsiaalskaala (illustreeritud joonisel 14), mis eristub selle poolest, et tekstisildid on vaid skaala otstes.

How satisfied are you overall with the quality of xyz brand?				
1 - Extremely Dissatisfied	2	3	4	5 - Extremely Satisfied
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Joonis 14. Semantiline diferentsiaalskaala

Kahe skaala võrdluses võib täpsemaks pidada Likert'i skaalat, mille puhul kasutaja teab, mida konkreetne vastusevariant tähendab [14].

Kuna käesolev töö on seotud kasutatavuse testimise automatiseerimisega, tehakse järgnevalt kokkuvõtte peatükis kirjeldatud meetoditest, mida oleks võimalik automatiseerida. Ülevaatusmeetoditest oleks kõige kergem automatiseerida heuristilist ülevaatus. Kuna veebielemendid, mida eksperdid visuaalselt hindavad, on veebilehtedel kujutatud ka HTML, CSS ning JS koodina, on nende andmed automaattarkvara jaoks kergesti kättesaadavad. Lisaks, kuivõrd koostatud on hulgaliselt kasutatavuse parimate praktikate kogumikke (nt WCAG ja Section 508), saaks neid hõlpsasti lisada reeglibaasi, mille vastu konkreetseid veebielemente valideerida. Pluraalse ja kognitiivse ülevaatus

automatiseerimine on tunduvalt keerulisem, sest hakkama ei saa inimfaktorita. Automatiseerimise üks lahendus oleks kasutada tehisintellekti, mis suudaks mõelda, liikuda ja suhelda nagu inimene. Valge kasti testimise puhul oleks lahenduseks samuti tehisintellekt, kes suudaks iseseisvalt teste välja mõelda ning neid käivitada. Kuna närvivõrkudega treenimine on viimastel aastatel jõudsalt arenenud, ei pruugi vastavate oskustega roboti loomine enam kaugel olla. Näiteks USA sõjaväe ja Google'i poolt rahastatud tehisintellekt Boyou oskab algelisel tasemel iseseisvalt Java koodi kirjutada<sup>1</sup>. Lõppkasutajaga testimise meetodeid on tänasel päeval keeruline automatiseerida, kuivõrd vaja on iseseisvat mõtlemist ning suhtlust. Ka siin tuleb ootama jääda tehisintellekti valmimist. *Live* süsteemide testimise meetoditest on veebianalüütika ning hindamiskaala andmete kogumise mõttes juba automatiseeritud. Keerulisem osa on aga veebianalüütika meetrikate ning hindamiskaala küsimuste välja mõtlemine, mille jaoks on taaskord vaja iseseisvat mõtlemist. Ka kriitiliste juhtumite lahendamise automatiseerimine on tänapäeval veel liiga keeruline, kuivõrd tarkvara peaks aru saama kliendi murest nii teksti kui kõne teel ning süübima rakenduse koodi ja logidesse, tuvastades selle põhjal vea.

---

<sup>1</sup> <https://tehnika.postimees.ee/4479651/teadlased-loid-tehisintellekti-mis-oskab-programmeerida> (04.05.2018)

## 4 Kasutatavuse testimine RIA's

Riigi Infosüsteemi Amet (RIA) on Majandus- ja Kommunikatsiooniministeeriumi haldusalas tegutsev riigiamet, mis koordineerib riigi infosüsteemi arendamist ja haldamist, korraldab infoturbega seotud tegevusi ja käsitleb Eesti arvutivõrkudes toimuvaid turvaintsidente. RIA konsulteerib avalike teenuste osutajaid, kuidas oma infosüsteeme nõuetekohaselt hallata ja teostab nende üle järelevalvet. Lisaks on RIA Euroopa Liidu struktuuritoetuste rakendusüksus<sup>1</sup>. Seega valdkondi, millega RIA tegeleb, on palju. Käesoleva lõputöö raames keskendutakse RIA haldusalas olevale riigiportaalile eesti.ee ja selle kasutatavusele. Kuna tegu on veebiga, mis peab võrdsetel tingimustel olema kättesaadav igale Eesti kodanikule, sõltumata vanusest, räägitavast keelest või puudest, siis ilmselgelt on nõudmised kasutatavuse osas kõrged. Lisaks vajab toonitamist, et riigiportali koosseisu kuulub sisuliselt kõik, mis seondub riigi või kohalike omavalitsustega. Arvuliselt tähendab see üle 500 e-teenuse, 500 artikli ja 2500 kontakti<sup>2</sup>. Arvestades, et iga e-teenus või artikkel koosneb vähemalt ühest veebilehest, siis veebilehtede koguarv, mis RIA vastutusalasse kuulub, on üle tuhande. Kõrgete nõudmiste ning ulatusliku skoobi tõttu on riigiportali kasutatavuse implementeerimine ja testimine keeruline ning aeganõudev, kuid kahtlemata vajalik.

### 4.1 Eesti.ee kasutatavuse parandamise vajadus

Eesti.ee 2016. aasta statistika järgi külastati riigiportali enam kui 40 miljonit korda ning veebisaidi kogu kasutajate arv oli 474 421<sup>3</sup>. See tähendab, et üle kolmandiku Eesti rahvastikust (1.jaanuar 2018 seisuga oli Eesti rahvaarv 1 318 700<sup>4</sup>) kasutab või on kasutanud eesti.ee poolt pakutavaid teenuseid. Suure kasutajate arvu tõttu võib väita, et

---

<sup>1</sup> <https://www.ria.ee/ee/ria.html> (22.04.2018)

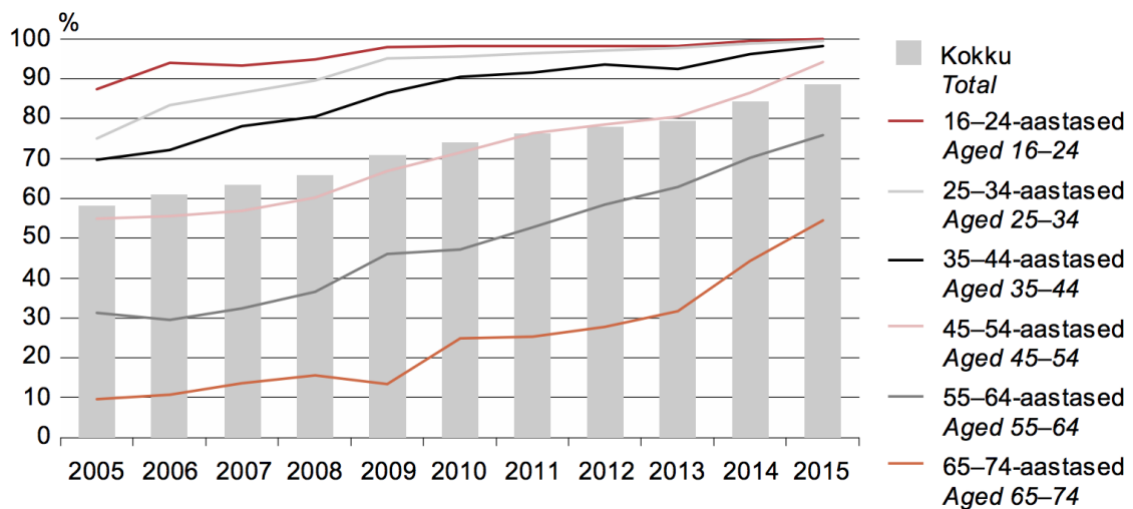
<sup>2</sup> <https://www.ria.ee/teejuht/riigi-infosusteemi-olemus-ja-komponendid/riigiportaal-eesti-ee> (22.04.2018)

<sup>3</sup>

[https://www.eesti.ee/est/teemad/ettevotja/riigiportali\\_abi/info\\_e\\_teenuse\\_tegijale/eesti\\_ee\\_2016\\_aasta\\_s\\_tatistika](https://www.eesti.ee/est/teemad/ettevotja/riigiportali_abi/info_e_teenuse_tegijale/eesti_ee_2016_aasta_s_tatistika) (22.04.2018)

<sup>4</sup> <https://www.stat.ee/pressiteade-2018-007> (22.04.2018)

eesti.ee kasutatavus on riigi seisukohalt äärmiselt oluline. Nagu eelnevalt mainitud, siis eesti.ee on veebisait, mis peab võrdsetel tingimustel olema kättesaadav kõigile Eesti kodanikele. See tähendab, et riigiteenuste kasutamisega peab toime tulema nii kooliõpilane, keskealine kui pensionär, kuid samas ka puudega inimene. Statistikaameti sotsiaaltrendide uuringust [15] on näha, et erinevatesse vanusegruppidesse kuuluvate internetikasutajate arv vahemikus 2005-2015 on tõusnud jõuliselt (illustreeritud joonisel 15). Sellest võib järeldada, et eesti.ee kasutajate arvu tõusu on oodata tulevikus veelgi ning veebikeskkonna kasutatavus muutub üha kriitilisemaks. “Eesti Statistika Kvartalikiri. 2/17” järgi [16] oli 2017. aasta alguses puuetega inimeste osakaal kogu Eesti rahvastikust 11,4%. Võttes aluseks Eesti rahvaarvu 2018. aasta jaanuari numbri, siis puuetega inimesi on Eestis ümmarguselt 130 000. Toetudes riigiportaali 2016. aasta statistika kasutajate arvule ning “Eesti Statistika Kvartalikiri. 2/17” puuetega inimeste osakaalule Eesti rahvastikus, siis lihtsa arvutusega saab leida, et ligi 43 000 puudega inimest kasutab eesti.ee’d.



Joonis 15. Internetikasutajate osatähtsus 16–74-aastaste seas vanuserühma järgi

Käesoleva töö peatükis 2.1 ilmnes, et kasutatavus ja juurdepääsetavus on omavahel seotud. Seega, kui tegeletakse süsteemi juurdepääsetavusega, parandatakse ühtlasi ka kasutatavust. Üks olulisemaid põhjuseid, miks RIA peab kasutatavusega tegelema, on Euroopa Parlamendi ja nõukogu poolt 2016. aastal koostatud direktiiv avaliku sektori asutuste veebilehtede ja mobiilirakenduste juurdepääsetavuse kohta [5]. Artikkel 12 järgi on direktiiv kohaldamiseks kõikidele Euroopa Liidu liikmesriikide avaliku sektori organite veebisaitidele (sealhulgas ka Eesti riigiportaalile) alates 23. septembrist 2020. Direktiivi preambula punkt 2 sätestab, et veebilehtede ja mobiilirakenduste disainimisel,

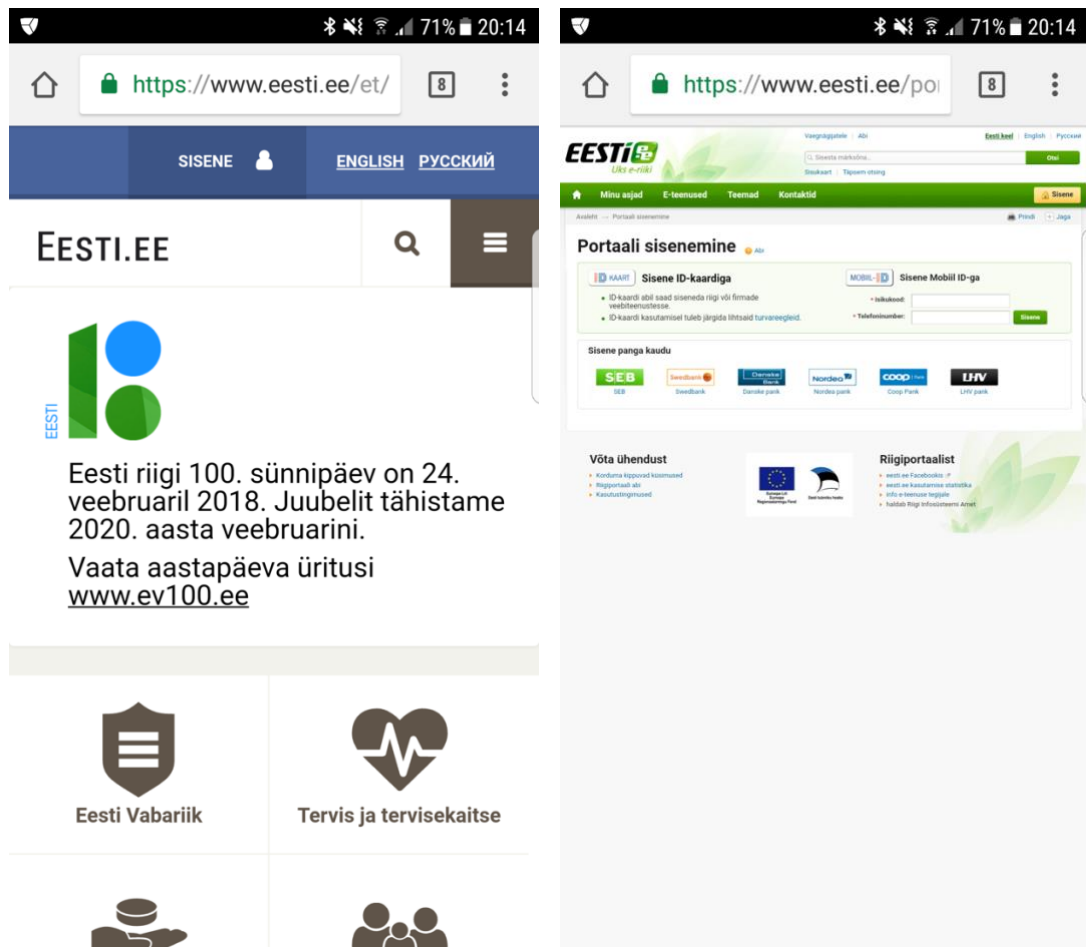
arendamisel, haldamisel ja ajakohastamisel tuleb järgida juurdepääsetavuse põhimõtteid ning tehnikaid, muutmaks need kättesaadavaks kõikidele kasutajatele, kuid eelkõige puuetega inimestele. Lisaks, olgugi et direktiiv õhutab avaliku sektori asutusi muutma oma veebi juurdepääsetavaks kõikidele kasutajatele, siis vastavalt direktiivi preambula punktile 28 ei ole eesmärk omaette keelata mitteligipääsetava sisu paigaldamist. Kui taoline olukord esineb, siis peaksid avaliku sektori asutused lisama oma veebisaitidele või mobiilirakendustesse juurdepääsetavaid alternatiive. Preambula punkt 39 täpsustab, et avaliku sektori organid peaksid rakendama juurdepääsetavuse põhimõtteid määral, mis ei tekita neile ebaproportsionaalset koormust. See tähendab, et mõningatel juhtudel (nt legitiimsetel põhjustel) ei pruugi sisu täielikult ligipääsetavaks muutmise olla mõistlikel tingimustel võimalik, mille tõttu on juurdepääsetavuse nõuete mittejärgmine teatud määral lubatud. Sellegipoolest ei tohiks õigustatud põhjusteks pidada juurdepääsetavuse tähtsustamata jätmist ning ajaliste ressursside või vajalike teadmiste puudumist.

Eesti.ee tänast versiooni<sup>1</sup> uurides leiab märkimisväärselt kasutatavuse kitsaskohti. Üks kriitilisemaid on mobiilivaatega mitteühildumine. Olgugi, et portaali avaleht on mobiili jaoks kohaldatud (näidatud joonise 16 vasakpoolsel pildil), pole sisselogimise vaates mobiilikasutajatega arvestatud (näidatud joonisel 16 parempoolsel pildil), mille tõttu kasutajaandmete sisestamiseks tuleb pilti suurendada. Mobiilile mittevastamine on riigiportaali üleüldine probleem, millega enamikel lehtedel pole hetkeseisuga tegeletud.

---

<sup>1</sup> <https://www.eesti.ee/et/index.html> (27.04.2018)





Joonis 16. Riigiportaali mobiilivaated

Rääkides konkreetselt e-teenustest, on töös välja toodud elukoha muutmise teenus, mis on autori hinnangul kasutatavuse mõttes riigiportaali üks halvemaid. Elukohateate avaldus (joonis 17), mida mobiili jaoks pole kohaldatud ning mida on seetõttu keeruline täita, jätab soovida ka veebivaates.

Minu avaldused Uus avaldus Abi

ELUKOHATEATE AVALDUS

Avalduse esitaja:

Eesnimi RAIDO  
Perekonnanimi ROBEN  
Isikukood 39301130824

Kas esitan enda kohta elukohta andmed

Jah   
Ei

Esitan elukohateate minuga koos elavate järgmiste isiku(te) ja, eestkostetava(te) kohta: (alkiri on nõutav vähemalt 18-aastaselt teovõimeliselt isikult)

JRK	Eesnimi	Perenimi	Isikukood	Alkiri nõutav
1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
6	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>

Eelmine elukoht (aadress välismaal, täidetakse saabumisel välismaalt):

Riik

Aadress

Välismaale lahkumise aeg

UUE ELUKOHA ADRRESSI ANDMED:

Uue elukohta asukoht:

Uue sideaadressi andmed: (sideaadressina esitatakse põhielukohast ERINEV aadress, juhul kui ajutiselt elatakse pikemat aega mõnes muus kohas)

Uue sideaadressi asukoht:

Sideaadress kehtib: (kehtivuse lõpp esitatakse juhul kui see on teada)

Alates

Kuni

Sidevahendite andmed: (palume täita, et elukohateate menetleja saaks vajadusel Teiega ühendust, samas kantakse esitatud andmed rahvastikuregistrisse sideandmeteks)

E-post

Telefon

Muu sidevahend

Sidevahendite andmed kehtivad: (kehtivuse lõpp esitatakse juhul kui see on teada)

Alates

Kuni

Omaniku või tema esindaja andmed (täidetakse, kui avalduse esitaja ei ole eluruumi omanik või kaasomanik):

JRK	Eesnimi	Perenimi	Isikukood	Alkiri nõutav
1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>

Salvesta avaldus

Joonis 17. Eesti.ee elukohateate avaldus

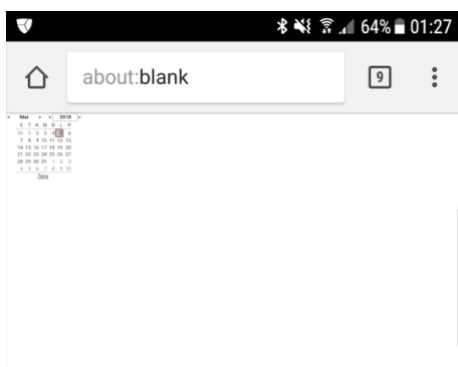
Antud teenuse juures on mitmeid aspekte, mida parandada. Esmalt jääb silma halb kujundus. Osades lausetes kasutatakse ainult suuri ja teistes väikeseid tähti, kuid loogika, mille alusel seda tehakse, pole arusaadav. Samuti jätavad lohaka mulje laused, mille taust on terve rea pikkuses markeeritud. Kahtlemata saaks olulise teksti toonitamiseks kasutada elegantsemat lahendust. Avalduse kõikide valikute viimased lahtrid on järgneva valiku selgitava tekstiga liiga lähestikku ning pärsivad loetavust. Et vältida elementide kokku sulamist, oleks mõistlik tekitada nende vahele lisaruumi.

Lisaks sellele, et vorm pole disainilt ilus, on see ka ebaloogiline. Näiteks “Esitan elukohateate minuga koos elavate järgmiste isiku(te) ja, eestkostetava(te) kohta” valikus on 6 rida. Esmalt tekib küsimus, miks on ridu nii palju ning teiseks, miks ei saa neid juurde lisada. Praktilisem lahendus oleks kuvada ühte rida ning nuppu uue rea lisamiseks. Valik “Eelmine elukoht (aadress välismaal, täidetakse saabumisel välismaalt)” on vabatahtlik ning seetõttu võiks kõiki lahtrid alguses mitte näidata. Esiteks võtab see ruumi, teiseks võib see ajada segadusse kasutaja, kes ei loe sulgudes olevat selgitust ning asub sisestama enda eelmise Eesti elukoha andmeid. Parem lahendus oleks lisada nupp või valikukast, millele klikkides avatakse lahtrid detailinfo sisestamiseks. Element, mis esineb nii antud valiku kui ka mõne järgneva valiku juures ning on terve avalduse puhul kõige häirivam, on kalendri avamise nupp (illustreeritud joonisel 18).



Joonis 18. Elukohateate kalendri avamise nupp

Selle asemel, et avada nupule vajutades hüpinkaken kalendriga, avatakse hoopis uus vahekaart, milles olev kalender on liiga pisike. Magistritöö autor proovis kuupäeva sisestamist ka mobiilivaates, mille puhul soovitud päeva valimine kalendri algsuuruses (näidatud joonisel 19) osutus väiksuse tõttu võimatuks.



Joonis 19. Elukohateate kalendri kuupäeva valik mobiilis

Samas ei parandanud olukorda ka ekraani suurendamine. Kuna pildi sisse suumimisel jääb ekraani keskpunkt sinna, kust sõrmed hakkavad liikuma ning ekraani horisontaal- ja vertikaalsuunas kerimist antud vaates ei võimaldata, siis üleval vasakus nurgas asuvat kalendrit on normaalmõõtudes keeruline tabada. Liikudes avaldusel edasi, siis jääb põhjendamata ka “Omaniku või tema esindaja andmed” valiku 3 rida. Sarnaselt eelmisele

näitele ei saa ka siin ridu juurde lisada. Pärast avalduse salvestamist tekib nupp “annulleeri avaldus”. Lähtudes sellest, et eesti.ee’ d kasutavad inimesed erinevatest vanusegruppidest, siis ei pruugi iga kasutaja “annulleeri” tähendust teada. Vastaval nupul olev tekst võiks olla “kustuta avaldus”.

Viimasena annaks antud avalduse juures parandada veateateid. Selle asemel, et andmete salvestamisel näidata hoiatusakent vigastest või täitmata jäänud andmetest, võiks konkreetse lahtri kõrval selgitada, milles täpsemalt probleem seisneb (sarnaselt ptk 2.3.3 Dropbox’i näitele). Samuti oleks mõistlik kuvada kõik probleemid ühe korraga. Hetkel toimib süsteem selliselt, et kui üks viga sai lahendatud, kuid vigu esines veel, siis salvestamispupu uuel vajutamisel ilmub hoiatusakent uue probleemiga. Selline lahendus on aga kasutaja seisukohalt aeganõudev ja tüütu.

Lisaks eelmainitule viitab riigiportaali kasutatavuse puudujääkidele ka Eesti Vabariigi president Kersti Kaljulaid, kes Eesti Vabariigi 100. aastapäeva kõnes tõi välja, et e-riigi kasutajaliidesed on tihti vanamoodsad, ebaloogilised ja pikatoimelised<sup>1</sup>. Seega põhjuseid eesti.ee kasutatavuse parandamiseks on palju.

## **4.2 Praegused meetodid kasutatavuse hindamiseks**

Selleks, et kaardistada praegused meetodid ning töövahendid, mida RIA kasutatavuse hindamiseks rakendab, intervjueriti lõputöö raames Eesti riigiportaali osakonnajuhatajat Raimo Reimanni. Intervjuu käigus ilmnes, et hetkeseisuga testitakse kasutatavust vajaduspõhiselt. Ehk et kindlat protokollit ei järgita ning testide arv ja varieeruvus on uute arenduste puhul erinev. Võttes aluseks kõik meetodid, mida RIA on varasemalt rakendanud, alustatakse kasutatavuse testimist disainifaasis pluraalse ülevaatusena. Antud testi eesmärk on koguda paberprototüüpide abil esmast hinnangut oma töötajatelt (disainerid, arendajad, testijad ning muud huvitatud osapooled). Enne funktsionaalsuse lisamist ning pärast paberprototüübi transformeerimist HTML kujule analüüsivad veebivaate disaini RIA kasutatavuse eksperdid. Vastav test toimub disainifaasis heuristilist ülevaatusena rakendades. Arendusfaasis tehakse tänasel päeval valge kasti teste, kus koodi autor sooritab süvatesti koodi funktsionaalsuse valideerimiseks. Pärast uue

---

<sup>1</sup> <https://www.president.ee/et/ametitegevus/koned/14153-2018-02-24-10-47-40/index.html> (27.04.2018)

arenduse paigaldamist testkeskkonda hinnatakse veebilehe kasutatavust Total Validatoriga<sup>1</sup>. Antud tarkvara kontrollib veebilehe vastavust juurdepääsetavussuunistele WCAG 2.0. Testifaasis viiakse läbi ka eemalt testimist suurema testgrupi peal. Testi sooritamiseks on RIA'l eraldi testkeskkond, mis on testis osalejatele VPN'i abil kättesaadav. Pärast arenduse paigaldamist *live*'sse monitooritakse järelhooldusfaasis lõppkasutajate tegevusi erinevate veebianalüütika kogumise tööriistadega: Google Analytics<sup>2</sup> ja Plumb<sup>3</sup>.

Nagu eelnevast näha, siis kõikides arendustsüklifaasides, kus hea tava kohaselt kasutatavuse testimine toimub (disain, arendus, testimine ja järelhooldus), tehakse teste ka RIA's. Sellegipoolest võiks kasutatavuse testimise intensiivsus ning erinevate meetodite hulk olla suurem.

### **4.3 Automaatse kasutatavuse hindamise tarkvara Guideliner**

Muutmaks kasutatavuse testimist süsteemsemaks ning tagamaks, et kindel arv teste saaks iga uue arenduse puhul minimaalselt sooritatud, otsustas RIA paigaldada oma süsteemidesse TTÜ doktorandi Jevgeni Marenkovi poolt loodud tarkvara Guidelineri.

#### **4.3.1 Guidelineri kontseptsiooni tõestus**

Aastal 2016 sooritati Guidelineriga beeta testimine 117 kasutatavuse reeglita. Ehk eeldefineeritud 98 reeglile lisati veel 19 reeglit. Testi käigus hinnati üheksat Eesti avaliku sektori veebirakendust, mille koosseisu kuulusid meditsiini, kinnisvara, keskkonna ja muude valdkondade saidid. Valiku koostamisel lähtuti sellest, et neid kasutaksid Eesti elanikud sõltumata nende sotsiaalsest staatusest või erinevate seadmete kasutamise oskusest. Täpsemalt hinnati Keskkonnaministeeriumi, e-valitsuse, Riigi Kinnisvara, Riigi Infosüsteemi haldussüsteemi (RIHA), Eesti Teadusinfosüsteemi (ETIS), Lõuna-Eesti haigla, Valga haigla, Rakvere haigla ning Narva haigla veebikeskkondi. Eksperimendi käigus grupeeriti kasutatavuse kriteeriumid viide kategooriasse [17].

---

<sup>1</sup> <https://www.totalvalidator.com/> (27.04.2018)

<sup>2</sup> <https://analytics.google.com/> (27.04.2018)

<sup>3</sup> <https://plumbr.io/> (27.04.2018)

Grupid jaotati järgnevalt:

A: lingi kontrastsus, mis näitab ebasobilike linkide hulka protsentides

B: teksti vastavus, mis näitab mõõdukalt kasutatud rasvases trükis tekstiridade hulka võrreldes veebilehel paiknevate kõikide rasvases trükis olevate tekstiridade arvuga

C: kirjeldamata pealkirjad, mis näitab pealkirja elementide protsenti, mille atribuudis pole pealkiri kirjeldatud

D: graafiliste elementide maksimaalne kogu lubatud suurus ning graafiliste elementide tegelik kogusuurus kilobaitides (KB)

E: navigatsiooni nähtavus, millega kuvatakse tõene/väär vastavalt sellele, kas navigatsioon mahub täies ulatuses ekraanile

Tabelis 1 on esitatud eksperimendis kasutatud valimi tulemused. Kõige kriitilisemad kõrvalekalded normidest on esitatud rasvaselt.

Tabel 1. Eesti avaliku sektori veebilehete vastavus kasutatavuse kriteeriumitele

Kasutatavuse kriteeriumi grupp	A	B	C	D	E
Keskkonnaministeerium	2%	0/1	1,5%	95/500	Jah
E-valitsus	1,5%	0/5	0%	109/520	Jah
Riigi Kinnisvara	14%	0/0	0%	68/245	Jah
Riigi Infosüsteemi haldussüsteem	0%	1/2	15%	56/315	Jah
Eesti Teadusinfosüsteemi	44%	6/21	2,5%	16/40	Jah
Lõuna-Eesti haigla	0%	2/10	3,5%	447/543	Jah
Valga haigla	9%	0/12	4,5%	<b>1951/2500</b>	Jah
Rakvere haigla	<b>85%</b>	1/10	7,5%	78/392	Jah
Narva haigla	39%	0/66	6%	154/432	Jah

Testi tulemustest ilmnes, et enamus kasutatavuse kriteeriumid on täidetud. Sellegipoolest esines ka mõningaid suuremaid kõrvalekaldeid. Näiteks Rakvere haigla veebilehe puhul

ollakse 85% juhtudest vastuolus WCAG2 AA<sup>1</sup> värvi kontrastsuse kriteeriumiga. Vastavalt kontrastsuse loogikale<sup>2</sup> tähendab 1:1, et esi- ja tagaplaani komponendi värvus on sama ning 1:21, et vastavate elementide värvused on maksimaalselt erinevad (must ja valge). Rakvere haigla puhul oli enamike linkide ja tausta värvuse suhe 1.91:1. WCAG 2 AA järgi võiks see olla vähemalt 4.5:1. Vastav näitaja on eriti oluline halvenenud nägemisega kasutaja jaoks. Sama näitaja polnud kiita ka ETIS'e veebilehel, kus 44% juhtudest oli linkide nähtavus alla normi. Kriitilisemate kõrvalekalletega paistis silma ka Valga haigla veeb, mille piltide kogumaht on umbes 2 megabaiti. Antud näitaja viitab sellele, et kasutatud pilte pole optimeeritud [17]. 2 megabaiti vähendab aga lehe laadimise kiirust, mis soovituslikult võiks olla alla 1 sekundi [9]. Laadimiskiirus on eriti oluline veebilehtede külastamisel mobiiliga.

Antud eksperiment näitas, et Guideliner on piisavalt küps teostamiseks kasutajaliidese hindamist nii eeldefineeritud kui ka täiendavate reeglite baasil. Samuti tõestati, et tööriist on sobilik igasuguste veebikeskkondade hindamiseks [17].

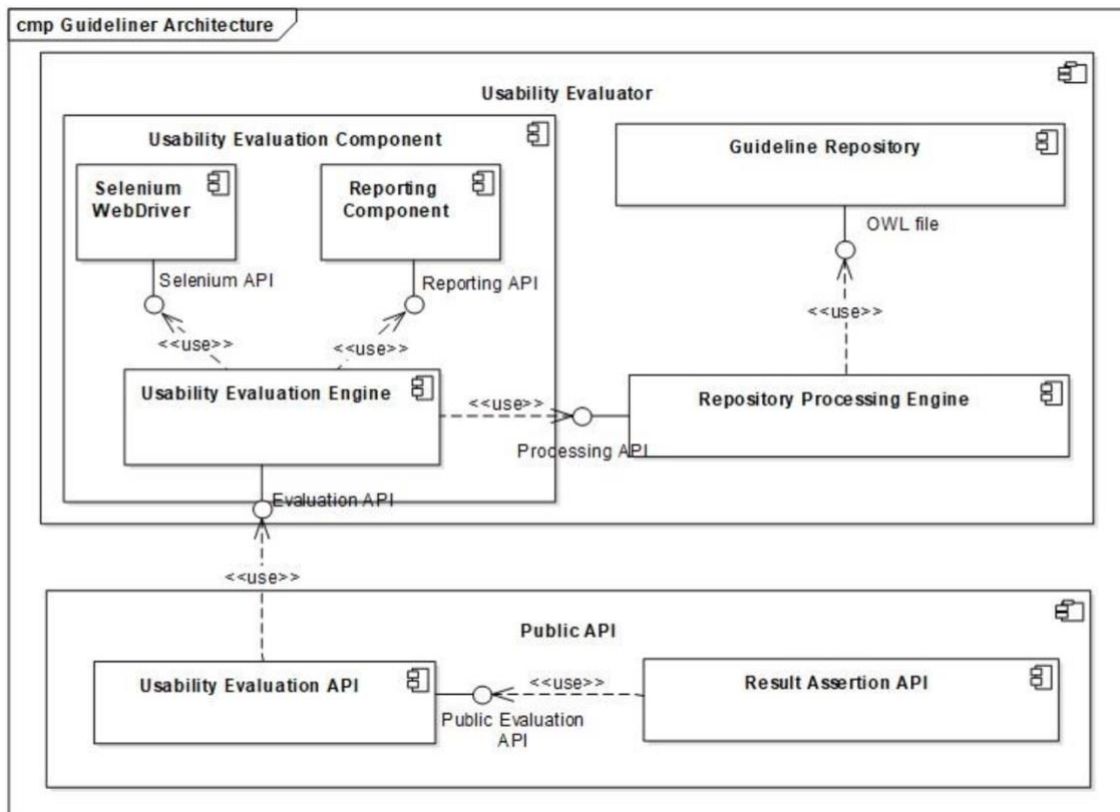
#### **4.3.2 Guidelineri arhitektuur ja tööpõhimõte**

Guideliner on Javal põhinev rakendus, mis koosneb väikestest iseseisvatest moodulitest. Antud muster tuleb kasuks kliendi süsteemiga liidestamisel. Kuivõrd iga kliendi süsteem on erinev, tuleb luua liidestus, mis ühildub just tema infrastruktuuriga. Moodulite isoleerimine võimaldab konkreetse kliendiga seotud äri loogika või protsesside lisamist ja muutmist viisil, et vigade esinemisel oleks mõjutatud vaid spetsiifiline liidestusega seotud komponent ning Guidelineri ülejäänud tükke viga ei mõjuta. Joonisel 20 on välja toodud Guidelineri arhitektuur, millel näidatakse kõiki tarkvara komponente ning nende vahelisi seoseid.

---

<sup>1</sup> <https://www.w3.org/Translations/WCAG20-et/> (27.04.2018)

<sup>2</sup> <https://decadecity.net/blog/2014/02/18/what-is-contrast-ratio> (27.04.2018)



Joonis 20. Guidelineri arhitektuur

Arhitektuuriliselt on Guideliner jaotatud kaheks suuremaks osaks: *Public API* ning *Usability Evaluator*. *Public API* ehk avalik API on komponent, millega luuakse ühendus kliendi süsteemi ja Guidelineri vahel. Selle sisendiks on kliendi poolt etteantav rakenduse URL ning väljundiks Guidelineri poolt tagastatavad testi tulemused vastava veebilehe kohta [18].

*Usability Evaluator* ehk kasutatavuse hindaja on kliendile mittenähtav komponent, milles peitub Guidelineri peamine tööloogika. Pärast hindamisprotsessi käivitamist laetakse *Guideline Repository*'st (GR) ehk kriteeriumite kogumikust kõik selles sisalduvad kasutatavuse kriteeriumid. Vastavaks toiminguks kasutatakse *Repository Processing Engine*'t ehk kogumiku töötlemise mootorit. Tagamaks täpseid tulemusi sisaldab GR vaid selliseid kriteeriume, mis on sobilikud just automaatseks kasutatavuse hindamiseks. Näiteks kriteerium, mis sätestab, et mobiilivaate nuppude laius peab olema vähemalt 48 pikslit, on automaatseks hindamiseks sobilik, sest nuppude laiust saab tarkvara abil lihtsasti arvutada. Seevastu tunduvat keerulisem on automaatselt hinnata kriteeriumit, milles öeldakse, et graafilisi elemente ei tohi kasutada veebilehe kaunistamise eesmärgil.



Kuna antud kriteeriumi hindamine ei annaks täpseid tulemusi, on see kriteeriumite kogumikust välja jäetud [18].

*Usability Evaluation Engine* (UEE) ehk kasutatavuse hindamise mootor teisendab veebilehe elemendid Guidelinerile arusaadavale kujule ning hindab nende vastavust GR'st sisse laetud kasutatavuse kriteeriumitele. UEE põhineb Seleniumi veebidraiveril. Antud draiver võimaldab testitava veebilehe operatsioonide ja käskude käivitamist. See on vajalik näiteks lehe laadimiseks, kasutajaliidese elementide asukoha määramiseks, elementide klikkimiseks, sisendväljade täitmiseks või veebivaadete vahetamiseks. Samuti pakub draiver vahendeid kasutatavuse kriteeriumite teisendamiseks. Teisenduse tööpõhimõte on esmalt sisse laadida kasutatavuse kriteeriumid ning seejärel kutsuda välja konkreetsele kriteeriumile vastav adapter. Näiteks nupu hindamisel kutsutakse välja *ButtonAdapter* (BA) ehk nupu adapter, millele antakse sisendina veebilehe elemendi parameetrid. Vastav adapter teeb parameetritele tuginedes vajalikud arvutused ning valideerib, kas element on vastavuses ette antud kasutatavuse kriteeriumiga. Kasutades eelpool mainitud nupunäidet, siis nupu laiuse hindamisel käsib BA Seleniumi draiveril üles leida kõik veebilehel olevad nupud ning arvutab nende laiused. Pärast seda kontrollib BA, kas veebidraiveri poolt sisse antud elemendi laiuse parameeter vastab kasutatavuse kriteeriumis defineeritud laiusele (nt kas laius  $\geq 48$ ) ning sellest sõltuvalt tagastab kas positiivse või negatiivse vastuse [18].

### 4.3.3 Guidelineri reeglistik

Üks olulisemaid Guidelineri komponente on reeglibaas, milles sisalduvatele reeglitele tuginetakse kasutatavuse hindamisel. Lähtudes Guidelineri arhitektuurist asuvad reeglid GR's. Algse seisuga on reeglibaasis vaid eeldefineeritud reegleid, kuid samuti saab sinna lisada ka uusi reegleid (nt kliendi ärireegleid). Eeldefineeritud reeglid on kombineeritud erinevate teadusväljaannete, WCAG, Section 508 ning HSS poolt välja pakutud kriteeriumitest [18].

*Web Content Accessibility Guidelines* (WCAG) ehk veebi sisu juurdepääsetavussuunised<sup>1</sup> on rahvusvaheline standard veebi sisu juurdepääsetavuse hindamiseks, mis on rakendatav erinevatele veebitehnoloogiatele ning käsitleb nii automaatseid kui ka inimese poolt

---

<sup>1</sup> <https://www.w3.org/Translations/WCAG20-et/> (27.04.2018)

läbiviidavaid teste. WCAG juhiste järgimine aitab parandada nii üldist kasutatavust, kuid on eriliselt suunatud süsteemi kättesaadavaks tegemisele puuetega inimestele (pimesus ja vaegnägemine, kurtus ja vaegkuulmine, õpiraskused, kognitiivsed piirangud, liikumispuue, kõnepuue, valgustundlikkus ja eelnevate kombinatsioonid).

Sarnaselt WCAG'le kehtestab ka Section 508<sup>1</sup> nõuded juurdepääsetavusele. Täpsemalt on tegu Ameerika Ühendriikide valitsuse poolt loodud standardiga, mille eesmärk on tagada valitsuse asutuste poolt arendatud või hangitud ning hooldatava või kasutuses oleva elektroonilise informatsiooni juurdepääsetavus puuetega inimestele.

*U.S. Department of Health and Human Services* (HSS) ehk Ameerika Ühendriikide Tervishoiu- ja Teenindusministeerium<sup>2</sup> on koostöös Ameerika Ühendriikide Üldteenusteametiga koostanud parimate praktikate kogumi. Kogumi eesmärgiks on luua paremaid ja kasutajasõbralikumaid tervise- ja sotsiaalteenustega seotud asutuste veebilehti.

Kuivõrd eelmainitud allikate suuniseid on kokku üle 300, osutusid neist 98 sobilikuks kasutatavuse automaatse hindamise teostamiseks. Välja valitud reeglid on sobilikud enamike veebielementide hindamiseks, sealhulgas lingid, nupud, tagid jne. Guidelineri eeldefineeritud reeglite koosseisu kuulub 20 mobiilivaate, 23 kasutatavuse ning 55 ligipääsetavusega seotud suunist, mis on sobilikud nii arvuti- kui mobiilivaate hindamiseks [18]. Tarkvara eeldefineeritud reeglid kategooriate kaupa on välja toodud tabelis 2.

Tabel 2. Guidelineri eeldefineeritud reeglid kategooriate kaupa

Kategooria	Mobiil	Üldine	Juurdepääsetavus	Kokku
Informatsiooni ja sisu organiseeritus	2	8	5	15
Tagi atribuudid	0	0	14	14
Lingid	4	3	3	10
Ekraanipõhised nupud	7	3	0	10

<sup>1</sup> <https://www.section508.va.gov/support/webdev/index.asp> (27.04.2018)

<sup>2</sup> [https://www.usability.gov/sites/default/files/documents/guidelines\\_book.pdf](https://www.usability.gov/sites/default/files/documents/guidelines_book.pdf) (27.04.2018)

Tagid	0	0	6	6
Raadionupp	1	2	3	6
Teksti välimus	0	3	6	9
Märkeruut	0	2	3	5
Pealkiri	0	0	5	5
Nupp	4	0	0	4
Graafika, pildid ja multimeedia	0	1	2	3
Selekteeri element	0	1	2	3
Kerimine	2	0	0	2
Parooli väli	0	0	2	2
Fail	0	0	2	2
Tekstiala	0	0	2	2
Kokku	20	23	55	98

#### 4.4 Guidelineri paigaldamine RIA süsteemidesse

Guidelineri paigaldamisega seotud töö koosnes 4 sammust: tarkvara koodimuudatustest, lähtekoodi hoidla välja vahetamisest, tarkvara spetsiifilise juhendi koostamisest ning Jenkinsi<sup>1</sup> projekti loomisest.

Esimese sammuna salvestas töö autor Guidelineri lähtekoodi ning käivitas selle enda masinas. Kuna Guideliner oli algselt kohaldatud vaid Firefoxile, tuli välja vahetada veebidraiver, mis ühilduks kõikide veebibrauseritega. Samuti tuli luua võimalus kasutatavuse testide käivitamiseks vastu konkreetset veebilehte. Selleks lisati võimalus sisendparameetri kaasaandmiseks. Viimasena täiustati logimise konfiguratsiooni rakenduse spetsiifiliste vigade tuvastamiseks.

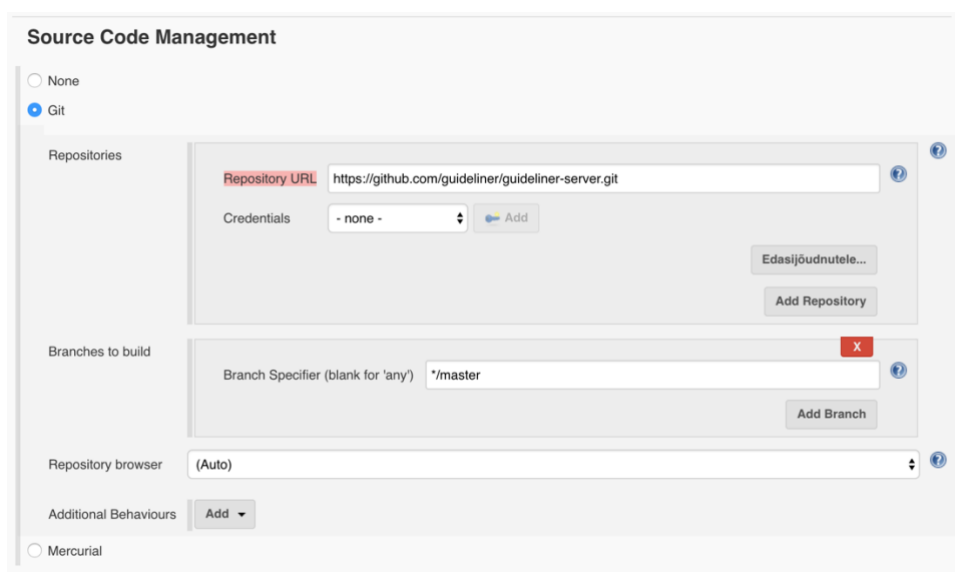
---

<sup>1</sup> <https://jenkins.io/> (27.04.2018)

Teise sammuna tõsteti lähtekood varasemast koodihoidlast Bitbucket'ist uude hoidlasse Github'i. Koodihoidla vahetus oli oluline, sest RIA kasutab tarkvara hoiustamiseks Github'i. Samuti vajab Github'i ka tarkvaraarenduse automatiseerimise tööriist Jenkins.

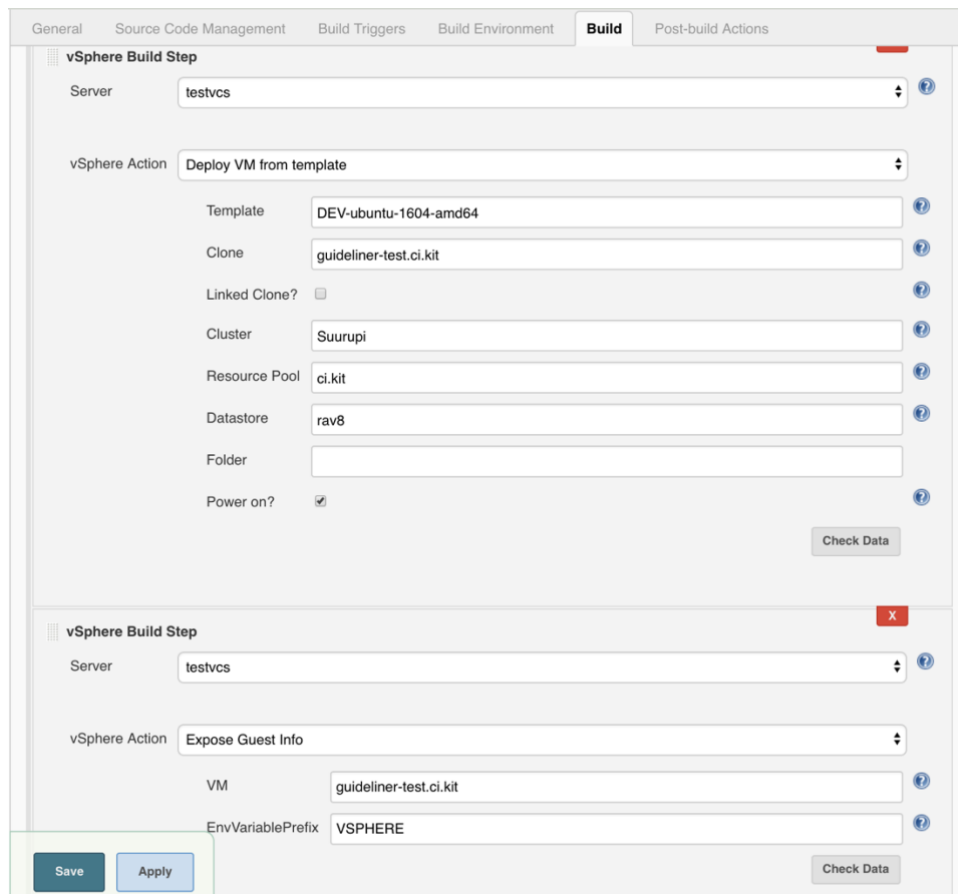
Kolmanda sammuna lisati lähtekoodile README.md fail, milles on lühidalt kirjeldatud Guidelineri eesmärk, reeglibaasi suurus, Java versioon ning vajalikud käsud rakenduse käivitamiseks. README.md dokument asub magistritöö lisas 1.

Neljas ning kõige mahukam etapp oli Guidelineri paigaldamine RIA süsteemi. Kuna RIA sisevõrk pole avalikust veebist kättesaadav, tuli luua ühendus RIA serveri ning enda seadme vahel. Ühenduse loomiseks kasutati RIA VPN klienti. Järgnevalt siseneti Jenkinsi veebirakendusse ning alustati paigaldust teostava projekti koostamist. Kuna töö autor polnud Jenkinsiga varasemalt kokku puutunud, tuli tarkvara spetsiifikat tundma õppida nii RIA kasutusjuhenditest kui internetis leiduvatest materjalidest. Esmalt fikseeriti Jenkinsi projekti lähtekoodi halduse seadistus (näidatud joonisel 21), kus koodihoidla tüübiks märgiti *git*, mis viitab GitHubile (kuhu lähtekood teises sammus tõsteti) ning koodihoidla URL'ks vastava lähtekoodi veebiaadress.



Joonis 21. Lähtekoodi haldus

Järgnevalt tekitati Guidelinerile isiklik tööruum (virtuaalserver), milles rakendus jooksmas panna. Selle loomiseks oli vaja Jenkinsi projektis defineerida vSphere Build Step (näidatud joonis 22), milles seadistati serveri konfiguratsioon.



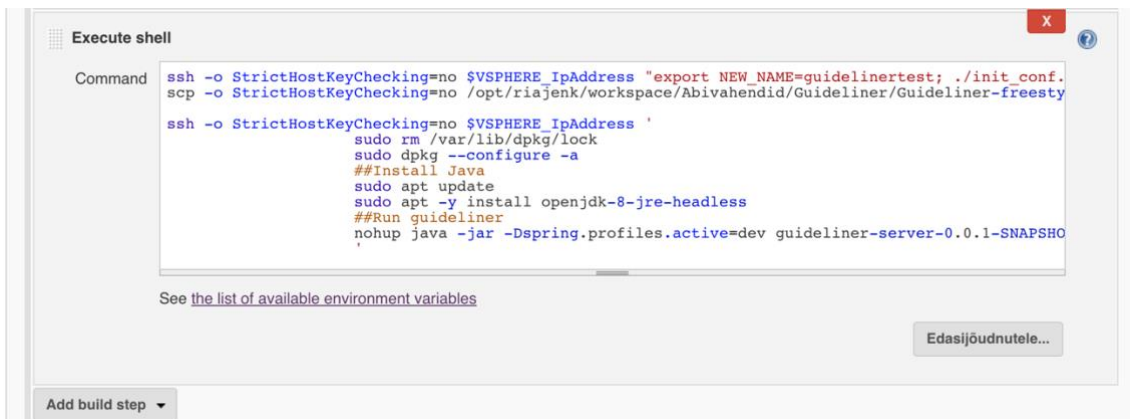
Joonis 22. Uue virtuaalmasina loomine

Samuti seadistati käsk, millega iga koodi muudatuse korral käivitatakse Guidelineri testid (näidatud joonisel 23). Testi sisendparameetriks anti riigiportaali URL.



Joonis 23. Testide käivitamise käsk

Viimasena kirjutati skript, millega virtuaalserverisse installitakse täiendav konfiguratsioon (nt Java 8, mis on vajalik Guidelineri töötamiseks), määratakse serveri spetsiifiliste globaalsete parameetrite nimed, genereeritakse testraport ning käivitatakse Guidelineri tarkvara (näidatud joonisel 24).



```
ssh -o StrictHostKeyChecking=no $VSPHERE_IpAddress "export NEW_NAME=guidelinertest; ./init_conf.  
scp -o StrictHostKeyChecking=no /opt/riajenk/workspace/Abivahendid/Guideliner/Guideliner-freesty  
  
ssh -o StrictHostKeyChecking=no $VSPHERE_IpAddress '  
sudo rm /var/lib/dpkg/lock  
sudo dpkg --configure -a  
##Install Java  
sudo apt update  
sudo apt -y install openjdk-8-jre-headless  
##Run guideliner  
nohup java -jar -Dspring.profiles.active=dev guideliner-server-0.0.1-SNAPSHO  
'
```

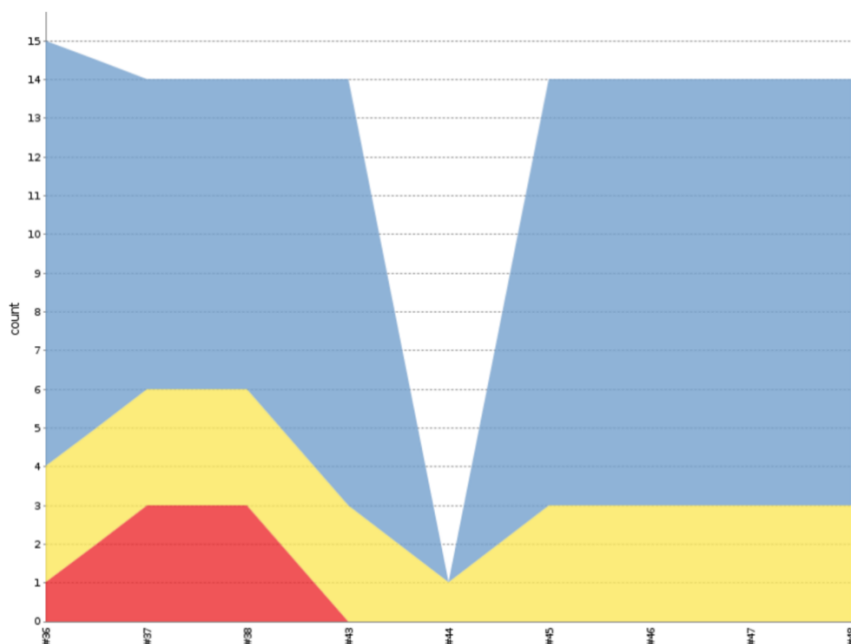
See [the list of available environment variables](#)

Edasijõudnutele...

Add build step ▾

Joonis 24. Jenkins'i projekti skript täiendavate funktsionaalsuste jaoks

Testraportit täiustatakse pärast iga uue muudatuse ülespanekut värskete andmetega. Samas jäävad alles ka tulemused eelnevate muudatuste lisamisest. Raporti graafilise presentatsiooni x-teljel on märgitud uue muudatuse number ning y-teljel testide arv (näidatud joonisel 25). Sinisega demonstreeritakse õnnestunud, kollasega vahele jäetud ning punasega ebaõnnestunud teste. Nagu näidatud joonistel 26 ja 27, siis iga testi tulemusi saab sirvida ka detailselt.



Joonis 25. Jenkins'i testitulemuste trend

## Test Result

0 ebaõnnestumist (±0) , 3 skipped (±0)



### Kõik testid

Package	Kestus	Ebaõnnestumisi (erinevus)	Vahele jäetud (erinevus)	Pass (erinevus)	Kokku (erinevus)
<a href="#">ee.ttu.usability.guideliner</a>	0 ms	0	3	0	3
<a href="#">jenkins</a>	24 sec	0	0	11	11

Joonis 26. Jenkins'i testitulemused kategooriatena

## Test Result : SampleUsabilityTest

0 ebaõnnestumist (±0)

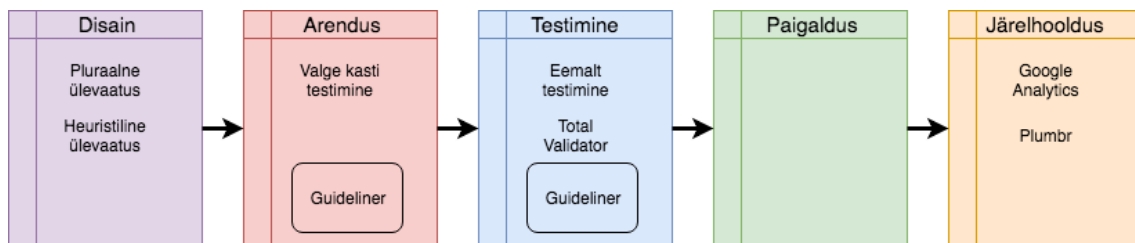


### All Tests

Test name	Duration	Status
<a href="#">testUsabilityGuidelines[0: 7wcag-1-1 ImageShouldHaveAltAttribute]</a>	22 sec	Passed
<a href="#">testUsabilityGuidelines[10: 6wcag-1-1 AreaShouldHaveAltAttribute]</a>	13 ms	Passed
<a href="#">testUsabilityGuidelines[1: 10wcag-1-1 ImageAltTextShouldNotBeAsFileName]</a>	21 ms	Passed
<a href="#">testUsabilityGuidelines[2: 9wcag-1-1 TextShouldNotContainMultipleSpace]</a>	1.3 sec	Passed
<a href="#">testUsabilityGuidelines[3: 8wcag-1-1 AlternativeTextShouldNotHaveProhibitedWords]</a>	0.15 sec	Passed
<a href="#">testUsabilityGuidelines[4: 3wcag-8-15 CheckThatPageDoesNotHaveLinkToItself]</a>	93 ms	Passed
<a href="#">testUsabilityGuidelines[5: 4wcag-8-16 CheckThatEveryPageHasTitle]</a>	15 ms	Passed
<a href="#">testUsabilityGuidelines[6: 11wcag-1-1 LinkAltTextShouldBeDifferentFromText]</a>	35 ms	Passed
<a href="#">testUsabilityGuidelines[7: 03-09 ProvideClientSideImageMaps]</a>	18 ms	Passed
<a href="#">testUsabilityGuidelines[8: 1wcag-8-10 CheckOnClickIsUsedWithOnKeyDown]</a>	16 ms	Passed
<a href="#">testUsabilityGuidelines[9: 5wcag-8-19 CheckThatLanguageIsIncludedToHtmlTag]</a>	14 ms	Passed

Joonis 27. Jenkins'i konkreetse kategooria testi tulemused

Seoses Guidelineri paigaldusega täienes senine RIA arendustsükkel uue kasutatavuse testimise meetodiga. Järgmisena on välja toodud kasutatavuse testid, mida sooritatakse RIA arendustsükli disainifaasist järelhooldusfaasini. Joonisel 28 on kuvatud faasid koos erinevate testmeetoditega. Ringiga on ümbritsetud Guidelineri testid, mis tarkvara paigaldamisega RIA senisesse arendustsükklisse lisandusid. Uus tööriist figureerib arendus- ja testfaasis. Kuivõrd varasemalt kasutati RIA arendusfaasis kasutatavuse hindamiseks vaid valge kasti testimist, siis pärast Guidelineri paigaldust täienes arendusfaas spetsiifiliste kasutatavuse testidega. See võimaldab Guidelineri teste käivitada arendaja enda masinas. Lisaks on paigaldatud Guideliner testkeskkonda, mis tagab vigadest teavitamise juhul, kui arendaja peaks unustama testide käivitamise enda seadmes. Sellega seoses on võimalik avastada kasutatavusega seotud probleeme enne muudatuste paigaldamist testkeskkonda.



Joonis 28. RIA arendustsükkel pärast Guidelinerit

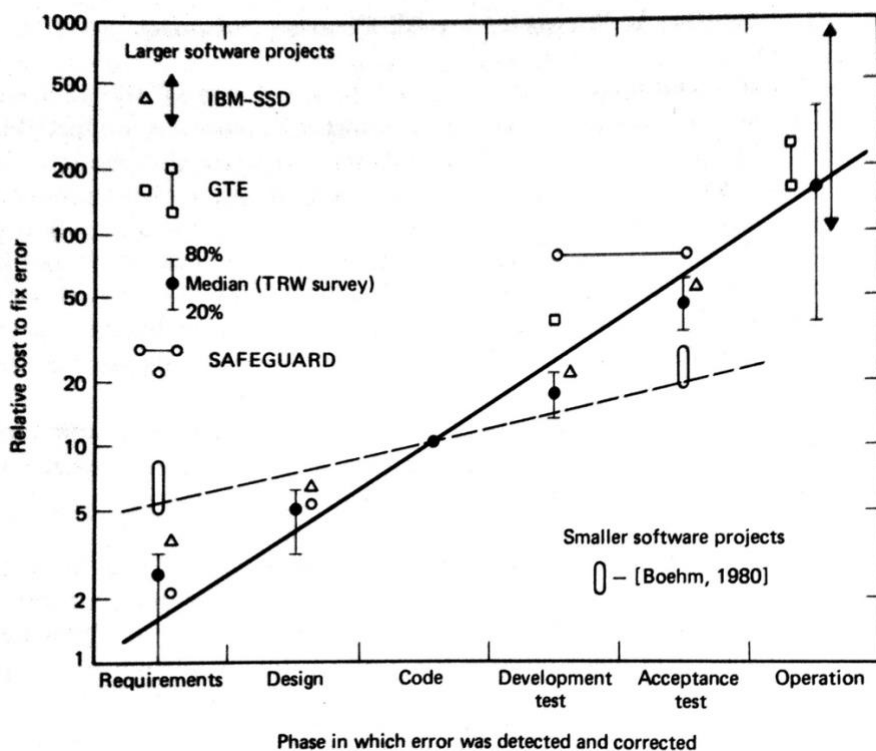
## 4.5 Tarkvara muudatuse maksumus erinevates arendusetappides

Üldlevinud on kaks arusaama tarkvara muutmise maksumusest arendustsükli erinevates faasides. Esimene väidab, et kulud suurenevad eksponentsiaalselt ning muudatuste sisseviimine hilises etapis on ekstreemselt kulukas. Teine ütleb, et muudatusi peaks tegema nii hilja kui võimalik, sest tarkvara muudatuste maksumus erinevates arendusetappides on või vähemalt võiks olla sirgjooneline.

### 4.5.1 Koodimuudatuse maksumuse eksponentsiaalne kõver

1981. aastal avaldas Barry Boehm statistika tarkvara muudatuste maksumusest [19]. Statistika koostamisel tugines Boehm 1970. aastatel kogutud TRW ja IBM'i projektide andmetele, kus tarkvara arenduseks kasutati kose mudelit. Boehmi statistika näitas, et muudatuse või paranduse tegemise hind kasvab iga arendustsükli faasiga. Boehmi poolt joonistatud hinna kasvu kõver on kujutatud joonisel 29.





Joonis 29. Boehmi hinna kasvu kõver

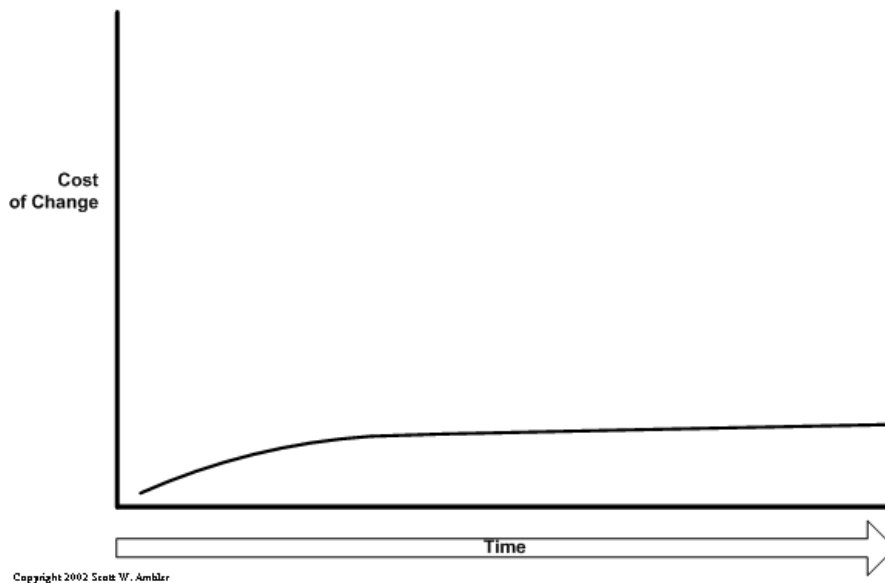
Kui nõuete muutmine analüüsifaasis ei maksa praktiliselt midagi, siis pärast tarkvara paigaldamist kliendi süsteemidesse võib selle maksumus olla sajakordne. Samas olgu lisatud, et nii suur hinna erinevus kaasneb väga suurte projektide puhul, kus põhimõttelise arhitektuurilise vea parandus järelhooldusfaasis võib osutuda äärmiselt kulukaks.

Boehmi tulemusi toetab ka Steve McConnell. Tuginedes uuringutele Hewlett-Packard'is, IBM'i, Huges Aircraft'is ning TRW's [20] väidab McConnell, et vabanedes vigadest arendusetapi varajases faasis, vähendatakse muudatuste sisse viimisele tehtavaid kulutusi 10-100 korda võrreldes sellega, kui need likvideerida pärast kliendipoolseid vastuvõtu teste või tarkvara üleandmist.

#### 4.5.2 Koodimuudatuse maksumuse sirgjoon

Kuna Boehm lähtus kose mudelist, kus enne järgmist arendustsükli etappi viiakse eelmine etapp täielikult lõpuni ning lõpetatud etapis enam muudatusi ei tehta (nt pärast analüüsifaasi arendusfaasis uusi nõudeid ei lisata), tuleks tänapäeval arvestada võimalusega lisada tarkvarale uut funktsionaalsust eraldiseisvate iteratiivsete arendustsüklitena. Antud ideest lähtub ka tänapäevane agiilne arendus, mille alla kuuluva *extreme programming* (XP) ehk ekstreemprogrammeerimise peamisi eestvedajaid Kent

Beck on öelnud, et muudatuse maksumuse minimeerimine on üks XP tehnilistest eeldustest [21]. Beck'i sõnul saab koodimuudatuste kõvera muuta sirgjooneliseks (illustreeritud joonisel 30), kui järgida XP võtmeprintsiipe. Sinna alla kuuluvad näiteks lihtne ja selge arhitektuur, pidev ning distsiplineeritud koodi refaktoormine, testimisest lähtuv arendus ning tihe suhtlus kliendiga. Samas puudub Beckil statistika, mis antud väidet toetaks.



Joonis 30. Kent Becki muudatuste maksumuse sirgjoon

Sirgjoonelise kõvera teooriaga nõustub ka Scott Amber, kuid lihtsa disaini asemel rõhutab Amber tagasisidestamist<sup>1</sup>. Näiteks tagasiside, mis antakse agiilsete arendusmeeskondade siseselt või saadakse pidevast kliendiga näost näkku suhtlusest, aitab maksumuse kõverat tasandada. Primaarseim kõikidest tagasiside liikidest on lõppkasutajate hinnang, mis annab täpseima kinnituse selle kohta, kas tehtud arendus on tähenduslik või ei lähe see kellelegi korda. Amberi maksumuse loogika on järgnev: mida kauem venitatakse disainimise või arendusega, seda rohkem aega kulub lõppkasutaja tagasiside saamisele ning seda kallimaks osutub arendus, kui see lõppastmes kasutajatele ei sobi. Ehk et mõistlik oleks esmase funktsionaalsusega välja tulla nii kiiresti kui võimalik ning veenduda, et kliendid on selle rakendamisest reaalselt huvitatud. Positiivse tagasiside korral võib hilisemas iteratsioonis teha rakendusele juurdearendusi.

---

<sup>1</sup> <http://www.agilemodeling.com/essays/costOfChange.htm> (31.04.2018)

### 4.5.3 Koodimuudatuse maksumuse tänapäevane statistika

Tuginedes eelpool mainitud arusaamadele ning isiklikele kogemustele leiab autor, et tõde muudatuste maksumusest on kahe variandi vahepeal. Kuna Becki avastused on enam kui 30 aasta vanused ning selle ajaga on tehnoloogia ning metoodika jõuliselt edasi arenenud, on ebatõenäoline, et muudatuste maksumus tänapäeval on sama kõrge. Samas tundub kahtlane, et muudatuste hind erinevates arendusetappides on võrdne. Kuna *lives* oleva tarkvara parandamiseks peab lisaks koodi muutmisele ka uuesti testimise ja paigaldama, on ajakulu kindlasti suurem, kui see oleks vea avastamisel arendusfaasis. Juhul kui kood on enne paigaldust vaja üle vaatamiseks saata mõnele teisele arendajale, kulub aega veelgi. Sellest tulenevalt toetub autor lõpliku tulemuse välja selgitamisel *National Institute of Standards and Technology* (NIST) ehk Rahvusvahelise Standardite ja Tehnoloogia Instituutsiooni poolt 2002. aastal väljastatud statistikale [22]. Tabelis 3 on esitatud etapid, millal vigu peamiselt avastatakse. Iga etapi juures on sulgudesse märgitud vea parandamise hinnatõus kordades. Samuti on välja toodud vigade avastamine sagedus protsentides. Nagu tabelist näha, siis enim vigu tuvastatakse integratsiooni- ja süsteemitestide ajal. Samas, märkimisväärsel hulgal vigu avastatakse alles live keskkonnas, kui rakendus on jõudnud lõppkasutajani. Vastavalt NIST'i andmetele on vigade parandamine arendusfaasis ligi 3 korda odavam kui testfaasis ning 6 korda odavam kui järelhoolduse faasis.

Tabel 3. NIST'i statistika muudatuste tegemise maksumusest

<b>Etapid, mil vigu tuvastatakse</b>	<b>Nõuete kirjutamine ja analüüs (1X)</b>	<b>Koodi kirjutamine ja ühiktestid (5X)</b>	<b>Integratsiooni- ja süsteemitestid (10X)</b>	<b>Varajane kliendi tagasiside ja beta testimine (15X)</b>	<b>Rakenduse töötamine <i>lives</i> (30X)</b>	<b>Kokku (%)</b>
Nõuete kirjutamine ja analüüs	3,5	10,5	35	6	15	70
Koodi kirjutamine ja ühiktestid		6	9	2	3	20
Integratsiooni ning süsteemitestid			6,5	1	2,5	10
Kokku (%)	3,5	16,5	50,5	9	20,5	100

## 4.6 Tulemused

Vastavalt veebilehe arenduse tavapraktikale rakendatakse kasutatavuse hindamiseks kvaliteedikontrolli (ingl *Quality Assurance*) ja kasutatavuse eksperte, mille puhul hindamine leiab aset testfaasis pärast koodi kirjutamist ning lisamist testkeskkonda. Antud praktika peamine nõrkus on, et arendajapoolse koodi paigaldamise ning eksperdi testraporti koostamise vahele jääb märkimisväärne ooteaeg. Selle aja jooksul on jõudnud arendaja tegeleda juba millegi muuga ning on vastava koodi spetsiifika ära unustanud. Kui raporti tulemused on negatiivsed ning arendaja peab koodi ümber kirjutama, võib arvestada, et aeg, mis kulub arendajal konkreetse arenduse meeldetuletamisele, on pikem kui kohese tagasiside korral. Võttes aluseks RIA arendusprotsessi enne automaatse kasutatavuse hindamise tarkvara paigaldamist, siis pärast koodi üleslaadimist kohest tagasiside arendajale ei genereeritud. Probleemi likvideeris Guidelineri lisamine arendusfaasi. Peamine tulu, mida kasutatavuse hindamise automatiseerimine kaasa toob, on ajaline ning rahaline kokkuhoid. Tänu kiirele tagasisidele on koodi autor arendusega värskelt kursis ning vigade esinemisel saab hakata parandustöödega kohele pihta. Kuivõrd Guidelineri reeglistik kattub suures osas heuristilises ülevaatuses järgitavate põhimõtetega, saab tarkvara kasutuselevõtuga vabaneda teatud määral praegusest eksperdi poolt läbiviidavast heuristilisest ülevaatuses ning kärpida kulutusi ka selle arvelt.

Pärast Guidelineri paigaldamist RIA süsteemidesse saab arendusfaasis sooritada spetsiifilisi kasutatavuse teste. Sellega seoses võimaldatakse kasutatavuse testimist ka arendaja masinas. Tagamaks vigadest teavitamist juhul kui arendaja on unustanud uue muudatuse kasutatavust enda masinas valideerida, on topeltkontrolli mõttes Guideliner veel testkeskkonnas. Samas kindlustatakse, et esialgsed probleemid oleks parandatud enne arendustsükli jõudmist testfaasi. Kuna testfaasis on omad kasutatavuse hindamise meetodid, tuleb kasuks, kui Guideliner on mõned puudused juba likvideerinud. Nagu peatükis 4.6 järeldati, on vigade parandamine arendusfaasis ligi 3 korda odavam kui testfaasis ning 6 korda odavam kui järelhoolduse faasis. Seega tänu arendusfaasi täiendamisele Guidelineriga, on võimalik RIA'l vähendada arendustööde kulutusi potentsiaalselt kuni 6 korda.

## 5 Kokkuvõte

Eesti.ee puhul on tegu avaliku sektori veebikeskkonnaga, mille kasutatavuse tase on riigi seisukohalt kriitilise tähtsusega. Kuna kasutatavuse hindamist teostatakse enamasti läbi inimtestide, on protseduur aeganõudev ja kulukas. Selleks, et testimist kiirendada ning tuvastada probleeme arendustsükli varajases faasis, soovib RIA kasutatavuse hindamist automatiseerida.

Sellega seoses paigaldati magistritöö esimese eesmärgi saavutamiseks RIA infrastruktuuri automaatse kasutatavuse hindamise tööriist Guideliner. Enne tarkvara paigaldamist täiustas töö autor selle lähtekoodi, et testide käivitamiseks saaks sisendina ette anda veebilehtede aadressid, mille vastu sooritatakse iga uue arenduse ülespanekul kasutatavuse hindamise testid. Samuti kirjutati töö käigus juhend, mille alusel RIA saab vastavaid aadresse muuta ning teste käivitada. Tarkvara paigaldamiseks kasutati tarkvaraarenduse automatiseerimise tööriista Jenkins, mille abil loodud projektiga võetakse Guidelineri lähtekood Github'ist ning tõstetakse soovitud RIA serverisse (nt test- ja *live* serveritesse). Samuti genereerib Jenkinsi projekt raporti testi tulemustest.

Magistritöö teine eesmärk oli tõestada, et kasutatavuse automaatse hindamise tarkvara aitab vähendada inimressursi kaasamist veebilehe kasutatavuse hindamisel ning kärpida üleüldisi arendusprotsessi kulusid. Antud tõestuse raames tuginetakse varasematele materjalidele, milles esmalt uuriti Guidelineri beeta-teste Eesti avaliku sektori veebilehtedega. Vastavate testide käigus ilmnis, et Guideliner annab adekvaatseid tulemusi ning on sobilik RIA süsteemidesse paigaldamiseks. Kuivõrd Guidelineri tööpõhimõte kattub suures osas heuristilise ülevaatusena, võimaldab tarkvara vähendada ekspertide kaasamist kasutatavuse hindamisel. Teisena analüüsiti uuringuid vigade avastamise maksumusest erinevates arendusetappides. Analüüsi käigus ilmnis, et vigade parandamine arendusfaasis on ligi 3 korda odavam kui testfaasis ning 6 korda odavam kui järelhoolduse faasis. Kuna Guidelineri teste saab käivitada juba arendusfaasis, kuid valdav osa RIA tänastest kasutatavuse testidest sooritatakse testimis- ja järelhooldusfaasis, aitab tarkvara kärpida arendustööde kulusid.

Magistritöö edasiarendusena võiks koguda andmeid sellest, kui palju on Guidelineri rakendamine aidanud RIA'l kasutatavuse probleeme tuvastada ning mis on vastavate probleemide täpsusprotsent ehk millisel määral otsustatakse leitud vead ka parandada. Lisaks võiks hinnata, millisel määral on Guidelineri poolt tuvastatud probleemid olnud olulised ka riigiportaali lõppkasutaja jaoks.

## Kasutatud kirjandus

- [1] ISO, “Ergonomic requirements for Office work with visual display terminals ISO-9241-11,” 1998.
- [2] ISO, “Ergonomics of human-system interaction ISO-9241-210,” 2010.
- [3] S. Horton, *A Web for Everyone: Designing Accessible User Experiences* Paperback, 2014.
- [4] D. Travis, *E-Commerce Usability: Tools and Techniques to Perfect the On-Line Experience*, 2003.
- [5] EUROPA, *EU Directive on the accessibility of the websites and mobile applications of public sector bodies*, 2016.
- [6] D. Rajapakse, *A Fresh Graduate’s Guide to Software Development Tools and Technologies*, 2012.
- [7] S. Joines, R. Willenborg and K. Hygh, *Performance Analysis for Java Web Sites*, 2003.
- [8] M. J. Collins, *Pro HTML5 with CSS, JavaScript and Multimedia: Complete Website Development and Best Practices*, 2017.
- [9] R. Maven, *Interview with Jakob Nielsen*. Retrieved from Econsultancy Digital Marketers United, 2007.
- [10] M. Ritter and C. Winterbottom, *UX for the Web: Build websites for user experience and usability*, 2017.
- [11] N. D. Birrell and M. A. Ould, *A Practical Handbook for Software Development*, 1988.
- [12] J. Nielsen and R. L. Mack, *Usability Inspection Methods*, 1994.
- [13] J. Gao, H.-S. J. Tsao and Y. Wu, *Testing and Quality Assurance for Component-based Software*, 2003.
- [14] J. Rubin and D. Chisnell, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, 2011.
- [15] Statistikaamet, *Sotsiaaltrendid*, 2016.
- [16] Statistikaamet, *Eesti Statistika Kvartalikirj. 2/17*, 2017.
- [17] J. Marenkov, T. Robal and A. Kalja, “A study on immediate automatic usability evaluation of web application user interfaces. Databases and Information Systems,” in *12th International Baltic Conference*, 2016.
- [18] J. Marenkov, T. Robal and A. Kalja, “Guideliner – a Tool to Improve Web UI Development for Better Usability (Accepter Paper),” in *8th International Conference on Web Intelligence, Mining and Semantics*, 2018.
- [19] B. W. Boehm, *Software Engineering Economics* 1st Edition, 1981.
- [20] S. McConnell, *Code Complete* 2nd Edition, 2004.

- [21] K. Beck and C. Andres, Extreme Programming Explained: Embrace Change, 2004.
- [22] NIST, The Economic Impacts of Inadequate Infrastructure for Software Testing, 2002.



## Lisa 1 –README.md fail

# Guideliner

[![N|Solid](guidleiner.jpg)]()

Guideliner is a tool for automatic usability evaluation.

# Features

- Allows writing usability unit tests
- Contains more than 40 predefined usability tests for mobile and desktop applications
- Provides REST API for usability evaluation

### Running as standalone server

Guideliner requires [JDK 1.8 +] to run.

Install and start the server.

```
```sh
$ mvn install
$ java -jar guideliner-server.jar
```
```

### Running as unittest

```
```sh
mvn test -Durl=https://www.example.com
```
```

-Durl parameter can contain any URL that you would like to verify