

Sanna Nykänen

**DIGITAALISTEN JÄRJESTELMIEN
KÄYTETTÄVYYDEN
ARVIOINTIMENETELMIEN
AUTOMATISOINTI**

TIIVISTELMÄ

Sanna Nykänen: Digitaalisten järjestelmien käytettävyyden arviointimenetelmien automatisointi
Kandidaattitutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Huhtikuu 2023

Käytettävyys muodostuu käyttäjän käyttäessä järjestelmää saavuttaakseen järjestelmän käytölle asettamansa tavoitteen. Käytettävyys kertoo, miten helppoa järjestelmän käyttö käyttäjän mielestä on. Sen rooli digitaalisissa järjestelmissä on kasvanut: enää pelkkä toimiva ohjelmisto ei riitä, sen täytyy tuottaa myös hyvä asiakaskokemus. Käytettävyys voi vaikuttaa siihen, kuinka halukas käyttäjä on käyttämään järjestelmää. Käytettävyysarviointi on edellytys intuitiiviselle ja helppokäyttöiselle järjestelmälle, mutta aina arviointia ei tehdä tai se tehdään vajavaisesti vedoten käytettävyysarvioinnista aiheutuviin kustannuksiin, aikapaineeseen tai resurssipulaan. Käytettävyyden arviointimenetelmien automatisointi esitetään yhtenä ratkaisuna edellä mainittuihin testaamiseen liittyvien haasteiden ratkaisemiseen.

Tämä kandidaattitutkielma käsittelee kirjallisuuskatsauksen avulla automaattisia käytettävyyden arviointimenetelmiä. Tutkielman tarkoitus on selvittää, millaisia erilaisia automaattisia käytettävyyden arviointimenetelmiä on. Kirjallisuuskatsaukseen on valittu alan julkaisuissa julkaistuja vertaisarvioituja artikkeleita, kirjallisuuskatsauksia ja kirjoja.

Tässä tutkielmassa automaattiset käytettävyyden arviointimenetelmät on ryhmitelty kolmeen eri pääkategoriaan: automaattinen tallennus, automaattinen analyysi ja automaattinen arviointi. Automaattisen tallennuksen menetelmät tallentavat testauksen aikana esimerkiksi käyttäjän hiiren liikkeitä ja klikkaukset testattavassa käyttöliittymässä, mutta käytettävyysongelmien tunnistaminen datasta jää ihmisen vastuulle. Automaattisessa analyysissä arviointimenetelmä osaa tunnistaa testattavasta käyttöliittymästä käytettävyysongelmia. Automaattisen arvioinnin menetelmät tunnistavat käytettävyysongelman ja ehdottavat sille korjaustoimenpiteitä.

Automaattiset käytettävyysarviointimenetelmät pyrkivät tekemään käytettävyysarvioinnista tehokkaampaa sekä säästämään aikaa, rahaa ja työresursseja. Kuitenkin kirjallisuuskatsauksessa paljastui, että automaattiset käytettävyyden arviointimenetelmät tarvitsevat myös manuaalista työtä toimiakseen. Lisäksi automaattiset arviointimenetelmät eivät pysty arvioimaan kaikkia olennaisesti käytettävyyteen liitettyjä osatekijöitä, kuten tuloksellisuutta tai tyytyväisyyttä. Automaattiset menetelmät sopivat parhaiten saavutettavuuden ja tehokkuuden arviointiin. Näitä menetelmiä tulee ajatella pikemmin apureina käytettävyysarvioinnin suorittamiseen.

Avainsanat: käytettävyys, käytettävyysarviointimenetelmät, käytettävyystestaus

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

SISÄLLYSLUETTELO

1	Johdanto	1
2	Tutkimusmenetelmä.....	3
3	Keskeiset käsitteet	3
	3.1 Käytettävyys	3
	3.2 Käytettävyyden arviointi	4
	3.3 Käytettävyysarvioinnin automatisointi	5
4	Automaattinen tallennus.....	6
5	Automaattinen analyysi	8
6	Automaattinen arvostelu.....	10
7	Menetelmien yleisiä haasteita	11
8	Keskustelu	12
9	Yhteenveto.....	13
	Lähdeluettelo.....	14

1 Johdanto

” If the user can’t use it, it doesn’t work.”

- Susan Dray

Elämme digitaalisessa maailmassa, jossa elämisen ehto on erilaisten digitaalisten järjestelmien käyttö. Järjestelmät tekevät käyttäjien elämästä mutkattomampaa, kun suurimman osan asioista voi hoitaa omalta kotisohvalta lähes mihin aikaan tahansa – mikäli käyttäjät osaavat käyttää järjestelmiä.

Vaikka käytettävyys on yksi kiinteä osa ohjelmistosuunnittelua, sen suunnittelu ja testaaminen usein jätetään tekemättä vedoten kustannuksiin ja aikaan liittyviin syihin (Khasnis ym., 2019). Edelleen ohjelmistotuotannossa käytetään enemmän resursseja toiminnallisten vaatimusten täyttämiseen kuin ohjelmiston käytettävyyden suunnitteluun ja testaamiseen. Intuiitiivisen ja helppokäyttöisen käyttöliittymän rakentamisen perusedellytyksenä on käytettävyyden testaaminen (Eskonen ym., 2020). Tutkimus myös osoittaa, että suurin syy ohjelmistoprojektien epäonnistumiseen on, ettei käyttäjä osaa käyttää ostamaansa järjestelmää (Harrati ym., 2015).

Käytettävyyden testaamisesta haasteellisempaa ja entistä tärkeämpää tekee erilaisten teknisten laitteiden ja näyttöresoluutioiden kasvava kirjo. Käytettävyydsarviointi voi hidastaa ohjelmiston julkaisua, jos se suoritetaan kaikille pienimmillekin muutoksille käyttöliittymässä. Kehittäjien ja käyttöliittymäsuunnittelijoiden välillä olevan kuilun takia kehittäjien on haastavaa ymmärtää, kuinka yksittäisetkin muutokset käyttöliittymässä voivat johtaa järjestelmän käytettävyyden ja saavutettavuuden huononemiseen. Esimerkiksi pienikin muutos käyttöliittymän fonttien värimaailmassa saattaa aiheuttaa suuria käytettävyysongelmia. (Marenkov ym., 2016) Sen vuoksi käyttäjätestaukset tulisi suorittaa aina, kun uusi käyttöliittymään vaikuttava ominaisuus lisätään järjestelmään tai jotain ominaisuutta päivitetään (Eskonen ym., 2020). Käytettävyydtestaus on manuaalista ja sen vuoksi kehitysiteraatiokierrokset nostavat testaamiseen liittyviä kustannuksia tuntuvasti.

Käytettävyys voi vaikuttaa siihen, kuinka halukkaita käyttäjät ovat käyttämään kyseistä järjestelmää (ISO 9241-11, 2018, 11). Esimerkiksi käyttäjien ensimmäiset kokemukset web-sivustosta vaikuttavat pidempiaikaisesti käyttäjän käsitykseen siitä ja siihen, vieraileeko käyttäjä uudestaan tulevaisuudessa sivustolla (Namoun ym., 2021). Hyvä käytettävyys voidaan katsoa järjestelmälle kilpailueduksi muihin markkinoilla oleviin järjestelmiin verrattuna. Tehokkuus järjestelmän käytössä voi tuoda liiketoiminnallisia säästöjä parantamalla organisaatioiden tehokkuutta. (ISO 9241-11, 2018, 11)

Yleisesti käytettävyyden huomioiminen järjestelmien suunnittelussa palvelee käyttäjien lisäksi liiketoiminnallisia motiiveja. Tietotekniikan alalla ohjelmisto-osaamisen ja yritysten välisen kilpailun kasvaessa yritykset tarvitsevat keinoja erottuakseen toisistaan. Enää pelkkä toimiva ohjelma ei riitä vaan sen pitää myös tuottaa hyvä asiakaskokemus.

Ohjelmistot tehdään käyttäjien ehdoilla sen sijaan, että käyttäjät muuttaisivat toimintaansa ohjelmiston logiikkaa vastaavaksi. Käyttöliittymän suunnitteluun on saatavilla myös automaattisia suunnittelutyökaluja, mutta tässä tutkielmassa keskitytään vain arviointityökaluihin.

Tämän kandidaattitutkielman tutkimuskysymyksenä on, miten ohjelmistojen käytettävyyssarviointimenetelmiä voidaan automatisoida. Näkökulmana on lisätä alan tietoutta suomeksi automaattisista käytettävyyden arviointimenetelmistä, niiden hyödyistä ja haasteista. Tässä tutkielmassa tarkastelen automaattisia järjestelmien käytettävyyden arviointiin tarkoitettuja menetelmiä, jotka tähtäävät käytön tuloksellisuuteen, tehokkuuteen, saavutettavuuteen ja tyytyväisyyteen. Tutkielmasta on rajattu pois muut tavoiteltavat lopputulemat, kuten käyttäjäkokemus. Tutkielma on toteutettu kirjallisuuskatsauksena.

Automatisoimalla käytettävyyden arviointimenetelmiä pyritään parantamaan arviointin tehokkuutta ja vaikuttavuutta. Automatisoinnin etuja ovat alhaisemmat kustannukset verrattuna ihmisten tekemiin testauksiin ja testaamisen tehokkuuden paraneminen. Lisäksi automaattiset testit voivat havaita käytettävyyssongelmia aikaisemmassa vaiheessa, esimerkiksi ohjelmistokehityksen aikana (Namoun ym., 2021, 293). Automatisoinnilla voidaan myös varmistaa, että käytettävyyssarviointi tapahtuu yhtä lailla kuin muutkin ohjelmistolle suoritettavat testit.

Automaattiset käytettävyyden arviointimenetelmät tallentavat, analysoivat ja/tai arvioivat käyttöliittymiä. Tallentavat menetelmät nauhoittavat käyttäjän ja käyttöliittymän interaktiossa tapahtuvia liikkeitä ja aikaa tehtävien suorittamiseen. Analysoivat menetelmät eivät tarvitse käyttäjää testitilanteeseen, vaan voivat suoraan etsiä käytettävyyssongelmia analysoimalla käyttöliittymää tai lokitiedostoja. Arvioivat menetelmät antavat tunnistamiinsa käytettävyyssongelmiin korjausehdotuksia.

Käytettävyyssarviointimenetelmien automatisoinnilla tavoitellaan usein testaamiseen käytettävän ajan ja rahan säästöjä, mutta menetelmien käyttöönotto ja ylläpito ei tapahdu ilman ihmisen työpanosta. Lisäksi menetelmät eivät voi korvata kokonaan käyttäjien tekemää testausta ja asiantuntijoiden käytettävyyssarviointeja. Automaattiset arviointimenetelmät onnistuvat mittaamaan järjestelmän saavutettavuutta ja tehokkuutta, mutta esimerkiksi tyytyväisyyden ja tuloksellisuuden mittaaminen ei niiltä onnistu.

Tässä tutkielmassa kuvaan ensiksi tiedonhankintaprosessia: miten tiedonhaku ja artikkeleiden valinta on toteutettu. Sen jälkeen määrittelen tutkielman kannalta olennaisimmat käsitteet. Tuloksissa esittelen kirjallisuuskatsauksen aineistossa esiteltyjä automaattisia käytettävyyssarviointimenetelmiä. Keskustelussa pohdin käytettävyyssarviointin tulevaisuutta ja uusien teknologioiden vaikutusta arviointimenetelmien kehitykseen.

2 Tutkimusmenetelmä

Tämän kandidaattitutkielman tutkimusmenetelmä on kirjallisuuskatsaus. Aineisto tutkimusta varten on haettu tammi-huhtikuussa 2023. Tiedonhaku kohdistui seuraaviin tietokantoihin: ACM Digital Library, IEEE Xplore, SpringerLink ja Google Scholar.

Tiedonhaussa käytetyt hakusanat on kirjattu alla olevaan taulukkoon (taulukko 1). Tiedonhakua on tehty yhdistelemällä hakusanaryhmää 1 ja 2 vaihtoehtoisesti ryhmän 3 tai 4 kanssa. Tietokantoihin kohdistetussa haussa käytettiin vain englanninkielisiä hakusanoja. Ensin haku aloitettiin käyttämällä hakusanaryhmiä 1–4. Lopuksi lisäksi hakuun vielä ryhmän 5, koska järjestelmiä käytetään käyttöliittymien kautta.

Taulukko 1. Tietokantoihin kohdistetut hakusanat.

Hakusanaryhmä	Hakusana suomeksi	Hakusana englanniksi
1	käytettävyys	usability
2	automatisointi	automat*
3	arviointi	evaluat*
4	testaus	test*
5	käyttöliittymä	user interface, UI, GUI

Tiedonhaussa käytettyjä lauseita olivat seuraavat:

- usability AND automat* AND test*
- usability AND automat* AND evaluat*
- user interface AND automat* AND test*
- gui AND automat* AND test*
- review AND automat* AND usability AND tool*

Kirjallisuuskatsaukseen valittiin tieteellisesti vertaisarvioituja artikkeleita, jotka on julkaistu vuosina 2015–2022. Artikkelit on valittu siten, että niiden otsikossa esiintyy sana ”usability” ja ne käsittelevät automaattisia käytettävyysarviointimenetelmiä. Mukaan on otettu myös kaksi käyttöliittymän testaamista käsittelevää artikkelia tuomaan esimerkkejä tulevaisuuden tekniikoiden hyödyntämisessä automaattisen testauksen kehittämisessä. Tutkielmassa on käytetty myös kirjallisuuskatsauksia.

3 Keskeiset käsitteet

Seuraavassa kuvataan tämän kirjallisuuskatsauksen kannalta olennaisimmat käsitteet.

3.1 Käytettävyys

ISO 9241-11 -standardi on yksi tapa määritellä käytettävyys. Sen mukaan käytettävyys muodostuu käyttäjän ja käytettävän järjestelmän vuorovaikutuksessa käyttäjän toimiessa saavuttaakseen tavoitteensa. Käytettävyys tarkoittaa kuinka tuloksellisesti, tehokkaasti ja

tyytyväisenä käyttäjä saavuttaa tavoitteensa. Tuloksellisuudella tarkoitetaan sitä, millä tarkkuudella ja miten täydellisesti käyttäjä saavuttaa tavoitteensa. Tehokkuus kuvaa käytettyjä resursseja tavoitteen saavuttamiseen. Näitä resursseja ovat suoritukseen käytetty aika, henkinen ja fyysinen vaiva, käytetyt materiaalit sekä kustannukset ja kulut. Tyytyväisyys kuvaa käyttäjän tunnetta ja reaktioita, jotka muodostuvat järjestelmän käytöstä sekä kuinka nämä tunteet ja reaktiot vastaavat käyttäjän järjestelmän käytölle asettamia tarpeita ja odotuksia. Käytettävyyden toteutumiseen vaikuttavat järjestelmän lisäksi sen käytölle määritellyt tavoitteet, tehtävät, käyttäjät ja käyttöympäristö. (ISO 9241-11, 2018, 14–17.)

Käytettävyyttä paljon tutkinut Jakob Nielsen (1993) määrittelee käytettävyyden olevan laadullinen ja moniulotteinen käsite kuvaamaan sitä, kuinka helppoa käyttöliittymien käyttö on. Nielsen määrittää käytettävyyttä ISO 9241-11 -standardin tavoin tehokkuudella ja tyytyväisyydellä. Näiden lisäksi Nielsen nostaa esiin opittavuuden (miten helppoa käyttäjien on saavuttaa tavoitteensa uudessa ympäristössä), muistettavuuden (muistavatko käyttäjät järjestelmän käytön periaatteet palatessaan sen pariin tietyn ajanjakson jälkeen) ja virheiden määrän (kuinka paljon virheitä tapahtuu ja miten käyttäjä toipuu niistä). Nielsen ottaa huomioon myös hyödyllisyyden: toteuttaako tuote ne ominaisuudet, jotka käyttäjä tarvitsee. Yhdessä käytettävyys ja hyödyllisyys muodostavat sen, onko käytettävä järjestelmä ylipäätään tarpeellinen. (Nielsen, 1993, 24–26)

Käytettävyyttä määrittelee myös järjestelmän käyttötarkoitus ja käyttäjät. Nielsenin määrittämien viiden käytettävyyssattribuutin painotukset voivat olla hyvin erilaisia eri käyttötilanteille. Esimerkiksi tavallisen kirjeenkirjoittajan vaatimukset tekstinkäsittelyohjelmiston käytettävyydelle ovat erilaiset verrattuna käyttäjään, joka käyttää ohjelmistoa satojen sivujen mittaisen teknisen dokumentaation ylläpitämiseen. (Nielsen, 1993, 27)

Käytettävyys voidaan tiivistää käsittämään sitä, miten miellyttävää järjestelmän käyttäminen käyttäjistä on, miten tehokkaasti käyttäjä saavuttaa järjestelmän käytölle asettamansa tavoitteet, minkälaisia virhetilanteita käyttäjä voi kohdata ja miten järjestelmä auttaa käyttäjää toipumaan niistä.

3.2 Käytettävyyden arviointi

Käytettävyyden arviointimenetelmät ovat prosesseja, joilla arvioidaan tarkasteltavan järjestelmän käytettävyyttä. Käytettävyyden määrittely vaikuttaa arviointimenetelmien valitsemiseen, koska jokainen arviointimenetelmä keskittyy käytettävyyden eri osa-alueen/osa-alueiden mittaamiseen. Käytettävyysarviointimenetelmät voidaan jakaa kahteen eri pääluokkaan: empiirisiin menetelmiin ja asiantuntija-arvioihin. (Fernandez ym., 2011)

Käytettävyystestausta voidaan tehdä formatiivisesti tai summatiivisesti. Formattiivista arviointia voidaan tehdä keskeneräiselle tuotteelle ja arvioinnin tarkoitus on parantaa järjestelmän käytettävyyttä. Formattiivisessa testauksessa yritetään selvittää, mikä testatta-

vassa järjestelmässä toimii hyvin, mikä huonosti ja miten järjestelmän käytettävyyttä voidaan parantaa. Summatiivisessa arvioinnissa arvioidaan valmista tuotetta ja sen laatua. (Nielsen, 1993, 130)

Empiiriset arviointimenetelmät

Empiirisissä menetelmissä loppukäyttäjä suorittaa ennalta määrättyjä tehtäviä järjestelmässä tai siitä tehdyssä prototyypissä. Järjestelmän tai prototyypin käytöstä syntynyt data tallennetaan analysointia varten. Empiirisissä menetelmissä oikea loppukäyttäjä käyttää järjestelmää ja käytöstä syntyvän datan perusteella tehdään analyyskejä järjestelmän käytettävyydestä. (Fernandez ym., 2011)

Asiantuntija-arvioinnit

Nimensä mukaan asiantuntija-arvioinneissa asiantuntijat tai suunnittelijat arvioivat järjestelmän käyttöliittymän käytettävyyttä ohjeistuksien avulla. Empiirisistä menetelmistä asiantuntija-arvioinnit eroavat siten, että loppukäyttäjät eivät ole mukana arvioinnissa. (Fernandez ym., 2011)

Heuristinen arviointi on systemaattinen tapa tehdä asiantuntija-arvioita. Arvioinnin tarkoituksena on löytää käytettävyysongelmia järjestelmästä ja näin parantaa iteraatioiden avulla järjestelmän käytettävyyttä. Yksi tunnetuista menetelmistä on heuristinen arviointi, jossa järjestelmää arvioidaan kymmenen heuristiikan avulla. (Nielsen, 1993, 155) Toinen esimerkki asiantuntija-arvioinnista on kognitiivinen läpikäynti, jossa järjestelmässä suoritettavat tehtävät jaetaan yksittäisiin vaiheisiin. Nämä vaiheet käydään yksitellen läpi suorittaen jokaisen kohdalla ennalta määritellyn kysymyspatteriston. (Lewis ym., 1990)

3.3 Käytettävyyсарvioinnin automatisointi

Käsitteenä automaattinen tarkoittaa laitteen tai ohjelmiston toimimista itsenäisesti ilman ihmisen ohjausta (Cambridge, 2023). Automatisointi puolestaan tarkoittaa aiemmin ihmisten tekemien rutiinomaisten tehtävien siirtämistä koneiden tehtäväksi (Oxford, 2023).

Käytettävyyсарvioinnin automatisointi tarkoittaa arvioinnin automatisointia osittain tai kokonaan. Käytettävyyсарvioinnin automatisointia voidaan luokitella neljän eri lähestymistavan kautta: ei-automaattiset menetelmät, automaattinen tallennus, automaattinen analyysi ja automaattinen arvostelu. (Ivory & Hearst, 2011) Tämä kirjallisuuskatsaus on jaoteltu näiden pääluokkien mukaan.

Ei-automaattisissa menetelmissä käytettävyyden arvioinnin ja datan keräämisen tekevät ihmiset. Automaattisessa tallennuksessa järjestelmä tallentaa arvioinnille merkityksellistä tietoa käyttäjästä ja testattavasta järjestelmästä, esimerkiksi kirjaamalla käyttöliit-

tymän tapahtumia. Automaattisessa analyysissä menetelmä osaa itse automaattisesti tunnistaa käytettävyysongelman. Automaattisessa arvostelussa menetelmä osaa ongelman tunnistamisen lisäksi ehdottaa keinoja, joilla ratkaista löydettyjä käytettävyysongelmia. Automaattisten arvostelumenetelmien antamat arvioinnit ja ratkaisut käytettävyysongelmille pohjautuvat asiantuntijoiden tekemiin ohjeistuksiin. (Ivory & Hearst, 2011.)

4 Automaattinen tallennus

Automaattisessa tallennuksessa käytettävyyssarviointiin käytettävä järjestelmä voi käyttöliittymän näytöllä tapahtuvien liikkeiden ja tapahtumien lisäksi tallentaa testisuoritukset videoina. Tässä luvussa esitellään tutkimusesimerkkien kautta erilaisia tapoja hyödyntää automaattista tallennusta käytettävyyssarviointissa.

Liikkeiden nauhoittaminen

Dixit ja Padmadas (2016) ovat tutkineet web-sivuston käytettävyyttä parantavaa apuria, jossa käyttäjä syöttää sivustolla olevalle apurille sivuston käytölle asettamansa tavoitteen, ja apuri tarjoaa suoraa linkkiä tavoitteen saavuttamiseksi. Apurin käytettävyyden testauksessa oli käytetty Selenium IDE -järjestelmää, joka laskee mm. käyttäjän testitehtäviin käyttämää aikaa ja sivustolla tehtyjä klikkauksia. (Dixit & Padmadas (2016, 142–144) Selenium IDE on avoimen lähdekoodin järjestelmä, jonka voi ladata liitännäiseksi selaimeen. Se tallentaa web-sivustolla tapahtuvat liikkeet ja järjestelmään voi luoda valmiita testitapauksia. (Software Freedom Conservancy, 2019)

Tutkimusmenetelmänä oli A/B-testaus, jossa toinen ryhmä testaa sivustoa ilman apuria ja toinen apurin kanssa. Tutkimuksen tavoitteena oli mitata, kuinka kauan käyttäjältä kesti oppia uuden järjestelmän käyttö ja tekikö apuri sivustolla navigoimisen helpommaksi. Selenium IDE -järjestelmä tallensi automaattisesti käyttäjän kursorin liikkeet käyttöliittymässä ja tehtävien suorittamiseen käytetyn ajan. Tästä kerätystä datasta Selenium IDE muodosti tilastoja, joiden avulla A/B-testejä voitiin vertailla toisiinsa. (Dixit & Padmadas, 2016, 144–146)

Järjestelmässä tapahtuvien toimintojen automaattinen tallentaminen on yhä yleisempää, ja sitä varten on kehitetty monia JavaScriptiin perustuvia työkaluja. Nämä työkalut mahdollistavat tapahtumakuuntelijoiden avulla erilaisten kursorin ja näppäimistön toimintojen havaitsemisen. Esimerkkinä näistä työkaluista ovat Piwik ja Google Analytics. Tallennetusta käyttödatasta voidaan johtaa esimerkiksi lämpökartan tyyllisiä analyysejä, jotka visualisoivat sen, missä kohdissa järjestelmää on eniten käyttäjän toimintaa. (Harms, 2015, 21; 26)

Kursorin liikkeistä ja tapahtumista kerätyn datan lisäksi, testitilanteen näytöstä voidaan tehdä videotallenne. Datasta tehtyjen tilastojen ja videotallenteiden avulla voidaan

määrittää käyttöliittymästä turhia komponentteja, jotka hidastavat käyttäjää suorittamaan testitilanteessa annettuja tehtäviä. (Khasnis ym., 2019)

Mallisuoritukseen vertaaminen

Harratin ja kumppaneiden (2015) tutkimuksessa verkkosivustojen käytettävyyden arviointiin käytettiin valmiiksi määriteltyyn mallisuoritukseen perustuvaa arviointimenetelmää, jossa käytettävyydsiantuntijan mallisuoritusta verrattiin käyttäjän suoriutumiseen. Arviointimenetelmässä oli kolme päävaihetta: 1) käytettävyydsiantuntijan tekemä malli tehtävän suorittamisesta, 2) käyttäjän tehtävien suorittamisen seuraaminen ja 3) datan analysointi.

Mallisuoritus purettiin puurakenteeseen, jotta se voitiin siirtää XML-kielelle. Käyttäjän tehtävässä suoriutumista seurattiin JavaScriptillä koodatulla ohjelmalla. JavaScriptin tapahtumakäsittelijöiden kautta järjestelmä keräsi dataa kaikista ruudulla tapahtuvista toiminnoista, ei vain käyttäjän tekemistä klikkauksista. Datan analysointivaiheessa käyttäjien tuloksia verrattiin käytettävyydsiantuntijan tekemään mallipolkuun tehtävän suorittamisesta. Mallisuorituksen ja käyttäjän tekemän suorituksen eroaminen merkittävästi toisistaan viesti potentiaalisesta käytettävyydsongelmasta. (Harrati ym., 2015)

Mallisuoritukseen perustuvaa käytettävyydsarviointia voi tehdä myös ilman käytettävyydsarvointilannetta. Silloin mallisuoritusta verrataan käyttäjien todelliseen järjestelmän käyttöön. WebHint, Web Automatic Usability Evaluation EnviRonment (WAUTER) ja Web Usability Probe mahdollistavat kyseisen arviointimenetelmän. Poikkeamat mallisuoritusta verratessa todelliseen käyttöön indikoivat käytettävyydsongelmista. (Harms, 2015, 27)

Käytettävyydsongelmia voidaan etsiä myös ennalta määriteltyjen tapahtumaketjujen kautta, joita tunnistetaan käytön aikana kerätystä tapahtumadatasta. Esimerkiksi tietyt kursorin ja hiiren liikkeet, kuten ylös-alas skrollaaminen ja sivulta toiselle hyppele, voivat kertoa siitä, ettei käyttäjä löydä etsimäänsä informaatiota. (Harms, 2015, 27–28)

Malliin perustuvassa seurannassa haasteena on kokonaisvaltaisten mallien luominen järjestelmästä. Mallien luominen vie paljon aikaa ja suurissa järjestelmissä kokonaisvaltaisten mallien toteuttaminen on liki mahdotonta. Lisäksi järjestelmän muuttuessa luotuja malleja on ylläpidettävä. (Harms, 2015, 27)

5 Automaattinen analyysi

Automaattinen analyysi tunnistaa sille annetusta aineistosta käytettävyyssongelmia. Aineiston ei tarvitse olla käyttäjän tuottamaa, ja osa menetelmistä perustuu enemmän asiantuntija-arvioihin kuin empiiriseen käytettävyytestaukseen.

Koodin vertaaminen ohjeistuksiin

Marenkovin ja kumppaneiden (2016) tutkimuksessa luotiin prototyyppi järjestelmästä, joka toimii käytettävyyden määrittelyyn luodulla kielellä (Usability Guideline Definition Language). Käytettävyyden määrittelevä kieli voitiin kääntää Selenium WebDriverin avulla useimmille suosituille ohjelmointikielille, kuten Java, C#, Python ja PHP. Järjestelmä kävi automaattisesti läpi sille annetun web-sivuston vertailemalla sitä järjestelmälle annettuihin käytettävyyismäärittelyihin.

Käytettävyyttä tarkasteltiin visuaalisen yhdenmukaisuuden kautta esimerkiksi vertailemalla, että kaikki linkit noudattivat samaa ulkoasua, lihavoidun tekstin määrä oli mallittainen, otsakkeet eivät olleet liian pitkiä ja sivustoille sisällytettyjen grafiikoiden koko oli järkevä. Järjestelmä tuotti automaattisesti raportin sivuston näin määritellystä käytettävyydestä ja antoi sivustolle prosenttiluvun, kuinka paljon sivustolla oli havaittu poikkeamia ohjeistoista. (Marenkov ym., 2016.)

Toinen koodin pohjalta käytettävyyttä arvioiva työkalu on USability Evaluation Framework (USEFUL). USEFUL arvioi verkkosivuston käytettävyyttä vertaamalla verkkosivuston lähdekoodia ohjelmalle syötettyihin ohjeistuksiin. Ohjelmalle on syötetty 240 käytettävyysohjeistusta ja ne perustuvat Yhdysvaltojen terveysministeriön teettämään web-sivustojen käytettävyystudkimukseen. USEFUL:ssa on esimerkiksi ohjeistus, joka tutkii, että html kielessä käytettävän ”a” tagin ”href” attribuutin sisältämä teksti on alle 70 merkkiä. (Mifsud & Dingli, 2011)

Lokitiedostojen analysointi

AMME (Automatic Mental Model Evaluator) muodostaa analyysejä järjestelmän lokitiedostoista. Aineistona ovat asiantuntijan ja loppukäyttäjän järjestelmän kanssa interaktiosta tallentuneet lokitiedostot. AMME muodostaa lokitiedostojen perusteella erilaisia analyysejä, kuten vierusmatriisin, taajuusmatriisin ja kvantitatiivisen kompetenssin. (Khasnis ym., 2019) Arviointimenetelmä ei itse tunnista analyyseistä käytettävyyssongelmia.

Lokitiedostojen analysoinnissa voi ongelmaksi muodostua niiden koko, mikäli kaikki käyttöliittymän tapahtumat, kuten cursorin kaikki liikkeet, tallennetaan. Lisäksi lokitiedostot voivat olla epäjohdonmukaisia ja poiketa toisistaan esimerkiksi järjestelmän kaatumisen takia. (Harms, 2015, 21)

Analyttinen mallintaminen

Analyttiset mallit tuovat insinöörimäisen lähestymistavan käytettävyyden arviointiin. Niissä arvioidaan tehtävien suorittamiseen kuluva aika erilaisten mallien avulla. (Fernandez ym., 2011) Halbrüggen (2018, 24–26) mukaan nämä käytettävyyсарviointimenetelmät pohjaavat Cardin, Moranin ja Newellin jo 1980-luvun alussa esittelemiin tekniikoihin: Goals, Operators, Methods and Selection rules (GOMS), The Keystroke-Level Model (KLM) ja ACT-R. GOMS on alun perin kehitetty mallintamaan ihmisen ongelmanratkaisuun käyttämää aikaa. Ratkaisun vaatima aika on laskettu summaamalla kognitiivisiin ja motorisiin prosesseihin sekä havaintojen tekemiseen kuluva aika. KLM-malli on GOMS-mallia yksinkertaisempi, ja sitä on käytetty tehtävien suorittamiseen kuluvan ajan arvioimiseen. ACT-R kartoittaa vain pelkistetyt kognitiiviset ja havainnointitoiminnot, jotka mahdollistavat ihmismielen toimimisen.

CogTool on KML- ja ACT-R -malleihin perustuva käytettävyyсарviointimenetelmä, joka ennustaa ongelmanratkaisuun menevää aikaa analysoimalla siihen vaadittavia kognitiivisia prosesseja. CogTool Explorer arvioi näiden lisäksi web-sivuston tutkimiseen liittyvää käyttäytymistä. Työkalu antaa arvioita siitä, kuinka kauan jonkun tietyn tehtävän suorittaminen käyttöliittymässä kestää. (Halbrügge, 2018, 27) CogTool on edelleen saatavilla ja se markkinoi olevansa ihmisen suorituskykyä ennustava malli (CogTool n.d).

GOMS Language Evaluation and Analysis -menetelmä (GLEAN-menetelmä) tarvitsee toimiakseen tehtäväanalyysin, jonka jälkeen se voi simuloida käyttäjän toimintaa. Työkalu ennustaa tavoitteiden saavuttamiseen menevää aikaa, analysoi käyttöliittymän yhteneväisyyttä ja antaa profiilin, kuinka paljon järjestelmän käyttö kuormittaa käyttäjän muistia. (Halbrügge, 2018, 28)

Näiden työkalujen huono puoli on, että ne vaativat järjestelmästä mallin tietyssä formaatissa toimiakseen. Esimerkiksi CogTool vaatii käyttöliittymästä visuaalisen mallin, eikä sitä voi ajaa suoraan käyttöliittymässä tai sen koodille. (Halbrügge 2018, 30)

Muita automaattisen analyysin menetelmiä

Käytettävyyteen vaikuttavia verkkosivun ominaisuuksia voidaan arvioida automaattisesti mittaamalla sivuston suorituskykyä, esimerkiksi sivuston latautumisaikaa. Sivustoa voidaan arvioida myös hakukoneoptimoinnin, sivuston turvallisuuden ja sivuston saavutettavuuden näkökulmasta. Esimerkiksi WebPage Analyser Tool laskee sivustolta kuvien määrän, kuvien koot ja sivuston lataamisajan. Qualidator Tool antaa prosenttiluvun sivuston käytettävyydelle, joka koostuu mm. saavutettavuuteen ja hakukoneoptimointiin liittyvistä arvioinneista. (Al-Omar, 2017)

Namoun ja kumppanit (2021, 292–298) ovat arvioineet kymmentä eri automaattista verkkosivustojen käytettävyyden arviointimenetelmää. Arviointimenetelmiä arvosteltiin vertaamalla mitä eri käytettävyyden ulottuvuuksia ne kattoivat. Käytettävyyttä kuvattiin

20 eri ulottuvuudella, joiden pohjana oli mm. Nielsenin käytettävyyden määritelmä. Eräs arvioitu työkalu oli Website Grader, joka mittasi sivustolta suorituskyvyn ja hakukoneoptimoinnin lisäksi järjestelmän skaalautuvuutta mobiililaitteelle ja järjestelmän turvallisuutta.

6 Automaattinen arvostelu

Automaattisessa arvostelussa käytettävyyсарviointimenetelmä osaa tunnistaa järjestelmästä käytettävyyssongelman ja antaa sille arvion sekä mahdollisesti korjausehdotuksen. Automaattisen arvostelun menetelmät eivät ole niin yleisiä kuin tallentamista tai analyysistä tuottavat menetelmät. Namounin ja kumppaneiden (2021, 309) arvioidessa automaattisia käytettävyyсарviointimenetelmiä, vain neljä kymmenestä antoi suosituksia käytettävyyssongelmiin korjaamiseksi.

Design-ohjeistuksiin perustuva malli

Kluth ja kumppanit (2014) ovat työskennelleet mobiililaitteiden automaattisen käytettävyyssongelman parissa. Heidän tarkoituksenaan oli luoda täysin automaattinen käytettävyyssongelman arviointijärjestelmä, joka antaa ehdotuksia käytettävyyssongelmiin korjaamiseksi.

Automaattisen arvostelun toteuttamiseksi arviointijärjestelmä vaati sitä varten luotuja yleensä käytettävyyssongelmiin johtavia tapahtumaketjuja. Näihin ennalta järjestelmään syötettyihin ongelmatilanteisiin yhdistettiin ratkaisuksi ihmisen ja tietokoneen vuorovaikutukseen tehtyjä design-malleja. Arviointi voitiin toteuttaa järjestelmän käytöstä tallennetulle datalle. Datan ei tarvinnut olla systemaattisesti kerättyä, vaan tavallisesta mobiililaitteen käytöstä muodostunut lokidata kelpasi. (Kluth ym., 2014)

Arviointityökalu toimi siten, että järjestelmän käytöstä muodostunut lokidata lähetettiin palvelimelle, jossa käytettävyyssongelmiin muodostettujen tapahtumaketjujen avulla lokidatasta tunnistettiin käytettävyyssongelmiä. Lokidataan kohdistuvan analyysin yhteyteen työkalu liitti löydetylle ongelmalle sille määritellyn korjausehdotuksen. Esimerkiksi jos käyttäjä jättää jonkun käyttöliittymäelementin usein huomiotta päästäkseen tavoitteeseensa, ratkaisu on tehdä tästä elementistä suurempi tai sijoittaa se keskeisemmälle paikalle. (Kluth ym., 2014)

Tehtäväpuuhun perustuva malli

Harms (2015, 11) on esitellyt väitöskirjassaan täysin automaattisen käytettävyyssongelman arviointityökalun, joka perustuu oikeasta käyttödatasta muodostettuun tehtäväpuuhun. Tapahtumapuuhun on hierarkkisesti järjestetty listaus toimista, joita käyttäjä suorittaa saavuttaakseen järjestelmälle asettamansa tavoitteen. Tapahtumaketjujen avulla työkalu tunnistaa tarkasteltavasta järjestelmästä ”käytettävyyssongelmiä”, jotka indikoivat käytettävyyssongelmiä.

Työkalu on ohjelmoitu tunnistamaan 14 erilaista käytettävyysongelmaan viittaavaa käytettävyyshajua. Näitä ovat esimerkiksi yleisimpien tehtävien suorittaminen minimaalisella ponnistelulla, tehottoman toiminnan vaatiminen käyttäjältä (esimerkiksi liian suuri lomake sivustolla, joka vaatii skrollausta) ja toiminnan suorittamisesta saatavan palautteen puuttuminen. Jokaiselle eri käytettävyyshajulle on määritelty arviointi ja korjausehdotus siihen liittyvään käytettävyysongelmaan. (Harms, 2015, 62–68.)

7 Menetelmien yleisiä haasteita

Namounin ja kumppaneiden (2021, 300; 308–309) arvioinnissa suurin osa tarkastelluista automaattisista käytettävyyсарviointityökaluista mittasi sivuston saavutettavuutta, sisältöä, latausaikaa ja suorituskykyä. Työkalut eivät kuitenkaan onnistuneet liittämään analyysin tuloksia käytettävyyden perusulottuvuuksiin, kuten opittavuuteen, muistettavuuteen ja tyytyväisyyteen. Lisäksi eri työkalujen arvioinnit samasta käytettävyyden osa-alueesta eivät olleet linjassa toistensa kanssa: vaikka työkalut mittasivat samalta sivustolta samaa asiaa, ne antoivat sille eri arvion.

Käytettävyyden arviointimenetelmien ongelma on se, että ne on kehitetty löytämään ongelmia eikä ratkomaan niitä. Arviointimenetelmät eivät tarjoa ratkaisuja, miten ongelmat voisi korjata asianmukaisesti. Käytettävyysongelmiensa listaaminen irrottaa ne asiayhteydestään ja näin pohjimmainen syy käytettävyysongelmalle jää raportoimatta. (Fernandez ym., 2011) Osa automaattisista käytettävyyсарviointityökaluista muodostaa analyysin tai kokonaisarvosanan sivuston käytettävyydelle tarjoamatta läpinäkyvää raporttia siitä, miksi kyseinen asia on työkalun mielestä käytettävyysongelma tai millä perusteella arvosana on annettu. Automaattista arviointia tekevien työkalujen korjausehdotukset käytettävyysongelmiille ovat usein ympäripyöreitä, geneerisiä ja eivät juuri liittyneet käytettävyyteen. (Namoun ym., 2021, 309)

Automaattiset käytettävyyсарviointit eivät kerää laadullista dataa ja jättävät usein subjektiiviset havainnot huomioimatta, kuten testitilanteessa ilmenevät käyttäjän mieltymykset ja väärinkäsitykset (Ivory & Hearst, 2001). Staattiset analyysit eivät pysty arvioimaan, ymmärtävätkö käyttäjät sivustolla käytettyjä termejä, ellei termistöä ole erikseen koodattu osaksi tarkastuslistaa (Harms, 2015, 26). Automaattinen tallennus tuottaa paljon dataa käyttäjän interaktiosta järjestelmän kanssa. Asiantuntijoiden tutkiessa massiivista määrää dataa saattaa unohtua ymmärrys käyttäjän käytölle asetetuista tavoitteista. (Khasnis ym., 2019)

Bakaev ja kumppanit (2022, 407–406; 411) tutkivat automaattisten työkalujen käyttöä käyttöliittymäsuunnittelussa. Tutkimukseen valittujen työkalujen tuli sisältää jonkin tason automaatiota ja niiden käyttö ei saanut vaatia loppukäyttäjää. Tutkimuksessa haastateltiin 34 digitaalisen suunnittelun ammattilaista ja käytettävyystudkijaa. Vain 8,8 %

vastaajista olivat tietoisia haastattelussa heille esiteltyjen automaattisten validointi- ja arviointi työkalujen olemassaolosta. Vastaajista kukaan ei käyttänyt näitä työkaluja työssään.

8 Keskustelu

Automaattista käytettävyyssarviointia on tehty verkkosivuille jo vuosia. Erilaiset saavutettavuutta arvioivat sivustot ovat esimerkkejä tästä. Kuitenkin näiden arviointimenetelmien kehittyminen tai lähinnä jämättäminen oli yllättävää. Muihin ohjelmistotestauksen osa-alueisiin verrattuna automaattiset käytettävyyssarviointimenetelmät ovat vielä kehityksensä alkupäässä. Automaattisten työkalujen hyödyt ohjelmistokehitykselle ja liiketoiminnalle voivat luoda illuusion siitä, ne ratkaisevat kaikki ongelmat. Kutenkin niiden saama kritiikki on väkevää ja osoittaa suuria haasteita työkalujen kehityksessä.

Tällä hetkellä käytettävyyssarvioinnin automatisoinnista isoin hyöty on edelleen saavutettavuuden, suorituskyvyn ja latausviiveen tarkastamisessa. Tyytyväisyyden, opittavuuden ja tuloksellisuuden mittaaminen ei automaattisilta työkaluilta onnistu ainakaan nykyään saatavilla teknologiolla.

Käytettävyyssarviointimenetelmien automatisointi voisi hyötyä tulevaisuuden teknologioista ja niiden mahdollistamista ratkaisuista automatisoinnin nykyisiin ongelmiin. Mahdollisia tulevaisuuden esimerkkejä käytettävyyssarviointimenetelmien kehitykseen etsin käyttöliittymätestauksen puolelta. Zimmermann (2022) esittelee idean käyttöliittymätestauksen automatisoimisen mallista, jossa käytetään hyväksi syväoppimiseen liittyviä hermoverkkoja. Tässä mallissa syväoppimiseen perustuva agentti pystyisi liikkumaan järjestelmässä virtuaalisen hiiren ja näppäimistön avulla. Tavoitteena agentilla olisi mahdollisimman hyvä testikattavuus järjestelmälle ja uusien ennen havaitsemattomien järjestelmävirheiden havaitseminen. Agentille ei entuudestaan suunniteltaisi testitapauksia, vaan sen pitää itse löytää sopivat testistrategiat. (Zimmermann, 2022)

Syväoppimiseen perustuu myös Android-käyttöliittymätestaukseen luotu työkalu Humanoid. Humanoidin tavoitteena on oppia, kuinka oikeat käyttäjät ovat vuorovaikutuksessa laitteen kanssa ja käyttää siitä luotuja toimintamalleja testaamaan muita käyttöliittymiä (Eskonen ym., 2020).

Syväoppimiseen perustuvat tekniikat antavat ajatuksia, miten käytettävyyden automaattisia arviointimenetelmiä voitaisiin kehittää tulevaisuudessa. Humanoid on kiinnostava esimerkki, miten syväoppimisen avulla voidaan opettaa järjestelmä jäljittelemään ihmisen käyttäytymistä. Kutkuttava ajatus olisi luoda malli, joka simuloi realistisesti ihmisen käyttäytymistä ja ymmärrystä järjestelmää testattaessa. Kuinka futuristista on pohdita sitä, että tulevaisuudessa olisi mahdollista tehdä käyttäjätestausta tekoälyn ihmissimulaatiolla? Voisiko tulevaisuuden ihmissimulaattorille suunnitella erilaisia persoonia, jotta se voisi testata järjestelmää eri näkökulmista?

9 Yhteenveto

Tämän kirjallisuuskatsauksen aiheena oli selvittää, miten käytettävyyden arviointimenetelmiä voi automatisoida. Aiheen selvittämiseksi työssä käytettiin tietokantahauilla löydettyjä vertaisarvioituja artikkeleita ja tutkimuksia.

Automaattiset käytettävyydsarviointimenetelmät voidaan jakaa kolmeen eri pääryhmään: tallentavat, analysoivat ja kritisoivat menetelmät. Tallentavissa menetelmissä työkalu tallentaa käytettävyydtestauksen aikana muodostuvaa dataa järjestelmän käytöstä ja tekee keräämästään datasta yhteenvedon. Tallennettavaa dataa ovat käyttäjän kursorin liikkeet, klikkaukset ja ennalta määriteltävien tehtävien suorittamiseen käytetty aika. Analysoivat menetelmät osaavat tunnistaa keräämästään datasta käytettävyydsongelmat, mutta eivät anna niihin korjausehdotuksia. Ne ajavat analyysin järjestelmälle, eivätkä tarvitse käyttäjää testaamaan järjestelmää. Saavutettavuuteen liittyvät analyysit ja sivustolla tehtäväpolkujen suorittamiseen kuluvan ajan ennustavat järjestelmät ovat esimerkkinä näistä. Arvosteltavat menetelmät antavat parannusehdotuksia sivustoille. Tallentavat menetelmät ovat lähempänä empiirisiä arviointimenetelmiä, kun taas analysoivat ja arvosteltavat menetelmät pohjautuvat enemmän asiantuntija-arvioihin.

Käytettävyyden arviointimenetelmien automatisoinnilla pyritään säästämään käytettävyydsarviointiin menevää aikaa, resursseja ja rahaa. Tallentavat menetelmät ovat mahdollistaneet arvioinnin tekemisen etänä.

Tutkimuksen perusteella myös automaattiset käytettävyydsarviointimenetelmät vaativat asiantuntijoiden työpanosta, testikäyttäjiä ja ylläpitoa. Käytettävyydsarviointia ei voi tehdä kokonaisvaltaisesti käyttämällä vain automaattisia arviointimenetelmiä. Automaattiset menetelmät ovat hyviä apureita saavutettavuuden ja tehokkuuden arviointiin, mutta ne eivät onnistu arvioimaan tuloksellisuutta, tyytyväisyyttä tai opittavuutta. Vaikka automaattisia arviointityökaluja on olemassa, ne eivät kirjallisuuskatsauksen tulosten perusteella ole tuttuja suunnittelijoille ja käytettävyydsammattilaisille.

Tulevaisuudessa aiheeseen liittyen voisi tutkia, kuinka uusia teknologioita voisi käyttää hyödyksi käytettävyydsarviointimenetelmien automatisointiin. Aiheesta kirjoitetut julkaisut vanhenevat nopeasti teknologioiden kehittyessä, joten olemassa olevien automaattisten arviointityökalujen hyödyllisyyden analysoinnille on tarvetta myös tulevaisuudessa.

Lähdeluettelo

- Al-Omar, K. (2017). Automated Usability Evaluation of E-Learning Websites in Saudi Arabia. *5th International Conference of Advanced Computer Science & Information Technology*, 01–10. <https://doi.org/10.5121/csit.2017.70801>
- Bakaev, M., Speicher, M., Jagow, J., Heil, S., Gaedke, M. (2022). We Don't Need No Real Users?! Surveying the Adoption of User-less Automation Tools by UI Design Practitioners. In: Di Noia, T., Ko, IY., Schedl, M., Ardito, C. (eds) *Web Engineering. ICWE 2022. Lecture Notes in Computer Science*, vol 13362. Springer, Cham. https://doi.org/10.1007/978-3-031-09917-5_28
- Cambridge. (2023). Cambridge University Press & Assessment, Cambridge Dictionary - Meaning of Automatic in English. Verkkosivu. Viitattu 21.3.2023. <https://dictionary.cambridge.org/dictionary/english/automatic>
- CogTool. n.d. CogTool – Predictive human performance modeling for UI design Verkkosivu. Viitattu 23.4.2023. <https://www.cogtool.org/>
- Dixit, S. & Padmadas, V. (2016). Automated Usability Evaluation of Web Applications. In: Satapathy, S., Bhatt, Y., Joshi, A., Mishra, D. (eds) *Proceedings of the International Congress on Information and Communication Technology. Advances in Intelligent Systems and Computing*, vol 439, 139–149. Springer, Singapore. https://doi-org/10.1007/978-981-10-0755-2_16
- Eskonen, J., Kahles, J. & Reijonen, J. (2020). Automating GUI Testing with Image-Based Deep Reinforcement Learning. *IEEE International Conference on Automatic Computing and Self-Organizing Systems (ACSOS)*, 160–167. <https://doi.org/10.1109/ACSOS49614.2020.00038>
- Fernandez, A., Insfran, E. & Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, 53(8), 789–817. <https://doi.org/10.1016/j.infsof.2011.02.007>
- Halbrügge, M. (2018). Automated Usability Evaluation (AUE). *Predicting User Performance and Errors*. T-Labs Series in Telecommunication Services. Springer, Cham. https://doi-org/10.1007/978-3-319-60369-8_4
- Harms, P. (2015). Automated Field Usability Evaluation Using Generated Task Trees. [Doctoral thesis, Georg-August-Universität Göttingen]. <https://d-nb.info/1081246847/34>
- Harrati, N., Bouchrika, I., Tari, A. & Ladjailia, A. (2015). Automating the Evaluation of Usability Remotely for Web Applications via a Model-Based Approach. *First International Conference on New Technologies of Information and Communication (NTIC) Conference Paper*. IEEE, 1–6. <https://doi.org/10.1109/NTIC.2015.7368757>
- ISO 9241-11. (2018). Ihmisen ja järjestelmän vuorovaikutuksen ergonomia. Osa 11: Käytettävyys. Määritelmiä ja käsitteitä. Helsinki: Suomen Standardisoimisliitto, 11–16.
- Ivory, M. & Hearst, M. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, 44(4), 470–516. <https://doi.org/10.1145/503112.503114>
- Khasnis, S., Raghuram S., Aditi, A., Samrakshini, R., & Namratha, M. (2019). Analysis

- of automation in the field of Usability Evaluation. *2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, 85–91.
<https://doi.org/10.1109/ICATIECE45860.2019.9063859>
- Kluth, W., Krempels, K-H. & Samsel, C. (2014). Automated Usability Testing for Mobile Applications. 10th International Conference on Web Information Systems and Technologies, 149–156. <https://doi.org/10.5220/0004985101490156>
- Lewis, C., Poison, P., Wharton, C., & Rieman, J. (1990). Testing a walk-through methodology for theory-based design of walk-up-and-use interfaces. *Proc. ACM CHI'90 Conf.*, 235–241
- Marenkov, J., Robal, T. & Kalja, A. (2016). A Study on Immediate Automatic Usability Evaluation of Web Application User Interfaces. G. Arnicans, V. Arnicane, J. Borzovs, & L. Niedrite (Eds.), *Databases and Information Systems. DB&IS 2016. Communications in Computer and Information Science*, vol 615, 257–271. Springer. https://doi.org/10.1007/978-3-319-40180-5_18
- Mifsud, J. & Dingli, A. (2011). USEFUL: A Framework to Mainstream Web Site Usability through Automated Evaluation. *International Journal of Human Computer Interaction*. 2011-10.
- Namoun, A., Alrehaili, A. & Tufail, A. (2021). A Review of Automated Website Usability Evaluation Tools: Research Issues and Challenges. In: M. Soares, E. Rosenzweig, & A. Marcus (Eds.), *Design, User Experience, and Usability: UX Research and Design*. HCII 2021. Lecture Notes in Computer Science, vol 12779, 305–318. Springer. https://doi.org/10.1007/978-3-030-78221-4_20
- Nielsen, J. (1993). *Usability Engineering* (1. painos). AP Professional.
- Oxford University Press. 2023. Definition of automation noun from the Oxford Advanced Learner's Dictionary. Verkkosivu. Viitattu 21.3.2023. <https://www.oxfordlearnersdictionaries.com/definition/english/automation>
- Software Freedom Conservancy (SFC). Selenium IDE. Verkkosivu. Viitattu 23.4.2023. <https://www.selenium.dev/selenium-ide/>
- Zimmermann, D. (2022). Automated GUI-based Software-Testing Using Deep Neuro-evolution. In 2022 IEEE Conference on Software Testing, Verification and Validation (ICST), 477–479. Valencia, Spain.
<https://doi.org/10.1109/ICST53961.2022.00060>.