Lassi Lehtovaara

KÄYTTÖLIITTYMÄSUUNNITTELU TURVALLISUUS-KRIITTISISSÄ JÄRJESTELMISSÄ

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2020

TIIVISTELMÄ

Lehtovaara, Lassi
Käyttöliittymäsuunnittelu turvallisuuskriittisissä järjestelmissä
Jyväskylä: Jyväskylän yliopisto, 2020, 33 s.
Tietojärjestelmätiede, kandidaatintutkielma
Ohjaaja(t): Kyppö, Jorma

Turvallisuuskriittinen järjestelmä on järjestelmä, jonka pettäminen tai epäonnistuminen voi aiheuttaa ihmishenkien menetystä tai merkittävää haittaa ympäristölle. Järjestelmien kriittisyyden vuoksi yksi olennainen tekijä turvallisuuskriittisissä järjestelmissä on niiden käyttöliittymä. Huonosti suunniteltu käyttöliittymä voi johtaa siihen, että käyttäjä tekee virheen josta voi aiheutua haittaa ihmisille, omaisuudelle tai ympäristölle. Ihminen on usein turvallisuuskriittisten järjestelmien heikoin lenkki, jonka takia ihmisen ja teknologian väliseen vuorovaikutukseen tulisi panostaa tällaisten kriittisten virheiden välttämiseksi. Tämä tutkielma keskittyy tutkimaan suunnittelussa huomioitavia asioita. Tutkielma on toteutettu systemaattisena kirjallisuuskatsauksena, jossa on käytetty lähteinä ajankohtaisia ja aiheen kannalta merkittäviä tieteellisiä lähteitä. Tutkielma pyrkii vastaamaan tutkimuskysymykseen: mitä tulee huomioida turvallisuuskriittistä käyttöliittymää suunnitellessa?

Asiasanat: käyttöliittymät, turvallisuuskriittiset järjestelmät, käytettävyys

ABSTRACT

Lehtovaara, Lassi
User interface design in Safety critical systems
Jyväskylä: University of Jyväskylä, 2020, 33 pp.
Information systems science, Bachelor's Thesis
Supervisor(s): Kyppö, Jorma

A safety-critical system is a system in which failure could cause loss of life or serious damage to the environment. Because of the critical nature of these systems, a crucial factor in a safety-critical system is the user interface. A badly designed user interface can cause the human operator to make a critical mistake that can then lead to the same result as stated above. A human is usually the weakest point in a safety-critical system which is the reason why organizations should invest in human-computer interaction to prevent these critical mistakes. This paper focuses on researching what are the things designers need to take into account when designing safety-critical systems. This paper is a systematic literature review and the sources used in this paper are the most relevant scientific papers and research for this topic. This paper focuses on finding an answer to the research question: what things to consider when designing safety-critical systems?

Keywords: user interface, safety critical systems, usability

SISÄLLYS

1 JOHDANTO

Erilaiset fyysiset ja digitaaliset järjestelmät ympäröivät meitä kaikkialla. Arjessa erilaiset järjestelmät ja laitteet, kuten palovaroittimet turvaavat elämäämme. Kun menemme töihin toimistollamme voi sprinklerijärjestelmä olla estämässä tulipaloja. Jos käymme lääkärissä, useat järjestelmät varmistavat, että saamme turvallista hoitoa. Matkalle lähtiessä luotamme lentokoneisiin turvallisena matkustusmuotona. Kaikki edellä mainitut järjestelmät ovat turvallisuuskriittisiä järjestelmiä. Turvallisuuskriittisyydellä tarkoitetaan sitä, että sellaisen järjestelmän pettäminen voi aiheuttaa ihmishenkien menetystä tai merkittävää haittaa ihmisille, omaisuudelle ja ympäristölle (Knight, 2002, s. 547).

Turvallisuuskriittisten järjestelmien käyttäjät ovat vuorovaikutuksessa järjestelmien kanssa jonkin käyttöliittymän välityksellä. Nämä käyttöliittymät ovat yleensä esimerkiksi tietokoneen näytöllä esiintyviä graafisia käyttöliittymiä, joiden kanssa käyttäjä on vuorovaikutuksessa eri tavoilla esimerkiksi äänen, oheislaitteiden kuten hiiren tai kosketusnäytön välityksellä. Turvallisuuskriittinen käyttöliittymäsuunnittelu keskittyy suunnittelemaan käyttäjille sellaisia järjestelmiä, jotka ovat turvallisia käyttää, mutta samalla käytettävyydeltään tehokkaita ja intuitiivisia.

Tämä tutkielma pyrkii kasaamaan tieteellisestä kirjallisuudesta löydettyjä ohjeita ja periaatteita, jonka avulla voidaan luoda mahdollisimman turvallisia ja käytettävyydeltään hyviä käyttöliittymiä turvallisuuskriittisille järjestelmille. Käydään ensimmäisenä läpi mitä kriittiset järjestelmät ovat ja millaisia erilaisia järjestelmätyyppejä on olemassa. Tutkitaan sen jälkeen turvallisuuskriittisiä järjestelmiä yhteiskunnassa ja mikä vaikutus kyberturvallisuudella on turvallisuuskriittisiin järjestelmiin. Kolmannessa sisältökappaleessa tutkitaan kirjallisuudessa esitettyjä yleisiä periaatteita esimerkiksi käytettävyyden ja käyttöliittymien suunnitteluun. Tutkitaan myös miten käyttöliittymiä tulisi testata ja arvioida, jotta ne vastaavat mahdollisimman hyvin käyttäjien tarpeita. Viimeinen sisältökappale syventyy tutkimaan mitä tulee huomioida turvallisuuskriittisen järjestelmän suunnittelussa ja millaisia haasteita suunnittelutyössä voidaan kohdata.

Tutkielman aihe sai alkunsa henkilökohtaisesta kiinnostuksesta, joka syntyi SpaceX:n Crew-1 tehtävän yhteydessä. SpaceX:n uuden Crew Dragon kapselin kosketusnäytöllinen käyttöliittymä oli radikaalisti erilainen verrattuna esimerkiksi NASA:n avaruussukkuloiden ohjausjärjestelmiin. Tämä herätti tutkijan mielenkiinnon, josta nousi kysymys mitä asioita pitää huomioida, kun suunnitellaan käyttöliittymää astronauteille? Tutkielman aihe on merkittävä, koska kuten aiemmin mainittiin turvallisuuskriittiset järjestelmät ympäröivät elämäämme kaikkialla ja ne ovat turvallisen arjen ja yhteiskunnan taustalla. Onnettomuuksia kuitenkin sattuu jatkuvasti ja siksi aiheen tutkiminen on tärkeää. Näin voidaan pelastaa mahdollisimman monia ihmishenkiä ja kehittää entistä turvallisempia järjestelmiä ja käyttöliittymiä.

Aihetta on aikaisemmin tutkittu melko kattavasti ja tieteellistä kirjallisuutta löytyy melko paljon. Tutkimukset kuitenkin keskittyvät aina vain yhteen kapeaan osa-alueeseen turvallisuuskriittisissä järjestelmissä, joten tämän tutkielman tarkoituksena on kasata tietoa yhteen paikkaan ja mahdollisimman yksinkertaiseen muotoon.

1.1 Tutkimuskysymys ja tutkimusmenetelmä

Tutkielmassa pyritään vastaamaan seuraavaan tutkimuskysymykseen:

- Mitä asioita tulee huomioida turvallisuuskriittistä käyttöliittymää suunnitellessa?

Tutkimuskysymykseen pyritään löytämään vastaus perehtymällä syvällisesti alan tieteelliseen kirjallisuuteen etsimällä eri tutkimuksista periaatteita ja näkökulmia siihen mitä suunnittelussa tulee huomioida.

Tutkielma on toteutettu systemaattisena kirjallisuuskatsauksena Jyväskylän yliopiston IT-tiedekunnan ohjesääntöjen mukaan. Tutkielmassa käytetyt lähteet on valittu niiden asianmukaisuuden ja ajankohtaisuuden perusteella. Vaikuttavina tekijöinä lähteiden valintaan oli myös niiden saama lainauksien määrä, jolla pyrittiin varmistamaan mahdollisimman laadukkaat lähteet. Lähteiden hakemiseen käytettiin seuraavia tietokantoja: Google, Google Scholar, IEE Explorer, JykDok ja Iris.ai. Tietokannoista käytettyjä hakusanoja olivat pääasiallisesti user interface, safety critical systems, design principles, HCI ja usability. Näistä hakutermeistä käytettiin myös suomenkielisiä käännöksiä.

2 KRIITTISET JÄRJESTELMÄT

Kriittisellä järjestelmällä tarkoitetaan sellaista järjestelmää, jonka pettäminen tai epäonnistuminen voi aiheuttaa uhkan ihmisen hengelle ja terveydelle, organisaation olemassaololle, merkittävälle taloudelliselle menetykselle tai merkittävälle ympäristöön kohdistuvalle haitalle (”Identity”, 2020). Tanhuamäki kirjoittaa tutkielmassaan toisen määritelmän kriittisistä järjestelmistä ”Kriittisiä tietojärjestelmiä ovat järjestelmät, joiden pettäminen johtaa menetyksiin, joita ei voida sallia. Tällaisia menetyksiä olisivat esimerkiksi ihmishenkien menetykset.” (Tanhuamäki, 2006, s. 2) Kriittisiä järjestelmiä on neljää eri tyyppiä ja niiden pettämisellä on kaikilla omanlaisensa vaikutukset. Neljä kriittisen järjestelmän tyyppiä ovat seuraavat: (Hinchey & Coyle, 2010, s. 431)

Turvallisuuskriittinen (Safety-Critical): Jos tällaisessa järjestelmässä tapahtuu häiriö tai vika, voi seurauksena olla ihmishenkien menetys, merkittävä loukkaantuminen tai vahinkoa ympäristölle/omaisuudelle (Hinchey & Coyle, 2010, s. 431). Turvallisuuskriittisiä (safety) järjestelmiä ovat esimerkiksi paloturvallisuuslaitteet kuten palohälyttimet, terveydenhuollon järjestelmät ja laitteet, ydinvoimaloiden järjestelmät, autojen aktiiviset ja passiiviset turvajärjestelmät ja muut liikkumisen välineet kuten lentokoneet, junat ja avaruusalukset. Tosiasiassa tämän tyyppisiä turvallisuuskriittisiä järjestelmiä on kaikkialla ja kaikkia ei tulla tässä tutkielmassa käsittelemään.

Tehtäväkriittinen (Mission-Critical): Tällaisen järjestelmän kaatuminen tai vioittuminen voi johtaa siihen ettei jotakin tavoitetta tai tehtävää voida suorittaa loppuun. Tämä voi johtaa kriittisen infrastruktuurin tai datan menettämiseen tai tuhoutumiseen (Hinchey & Coyle, 2010, s. 431). Esimerkki tehtäväkriittisestä järjestelmästä on pelastajien käyttämät viestintäjärjestelmät, joita pelastajat ja muut viranomaiset käyttävät suorittaessaan kriittisiä tehtäviä.

Liiketoimintakriittinen (Business-Critical): Ongelmat liiketoimintakriittisissä järjestelmissä voivat aiheuttaa merkittävää aineellista tai aineetonta taloudellista vahinkoa. Seurauksena voi olla liiketoiminnan tai maineen mentys (Hinchey & Coyle, 2010, s. 431). Esimerkki liiketoimintakriittisestä järjestelmästä on yrityksen ERP-järjestelmä tai digitaalinen pankkijärjestelmä.

Turvallisuuskriittinen (Security-Critical): Tämän tyyppisen kriittisenjärjestelmän ongelmat voivat aiheuttaa arkaluontoisen datan menetyksen varkauden tai vahingon seurauksena (Hinchey & Coyle, 2010, s. 431). Esimerkkinä tällaisesta järjestelmästä toimii psykoterapiakeskus Vastaamon potilastietokanta, johon murtauduttiin vuoden 2020 lokakuussa (Rimpiläinen, 2020). Hakkeri oli päässyt murtautumaan Vastaamon potilastietokantaan ja varastamaan sieltä erittäin arkaluontoista tietoa, kuten potilaskertomuksia ja henkilötunnuksia.

Tämä tutkielma keskittyy edellä mainituista kriittisistä järjestelmistä ensimmäiseen eli turvallisuuskriittisiin (safety) järjestelmiin. Aina kun tutkielmassa mainitaan turvallisuuskriittinen järjestelmä, sillä tarkoitetaan safety-critical-järjestelmää.

2.1 Turvallisuuskriittiset järjestelmät

Knight määrittelee turvallisuuskriittiset järjestelmät seuraavasti "Turvallisuuskriittiset järjestelmät ovat sellaisia järjestelmiä, joiden pettäminen voi aiheuttaa kuoleman, merkittävää vahinkoa omaisuudelle tai ympäristölle" (Knight, 2002, s. 547). Knight jaottelee turvallisuuskriittiset järjestelmät vielä kahteen ala-luokkaan. Perinteisiin järjestelmiin ja ei-perinteisiin järjestelmiin.

Perinteisillä järjestelmillä tarkoitetaan muun muassa terveydenhuollon, ilmailualan, ydinvoimaloiden ja aseiden järjestelmiä. Esimerkkinä tällaisesta järjestelmästä on lentokoneen järjestelmät joilla hallitaan lentokoneen eri ominaisuuksia. Tämän tyyppisten järjestelmien pettäminen aiheuttaisi merkittävää haittaa ihmisille ja ympäristölle (Knight, 2002, s. 547).

Ei-perinteisillä järjestelmillä tarkoitetaan sellaisia turvallisuuskriittisiä järjestelmiä, jotka eivät välttämättä pettäessään suoranaisesti aiheuta vahinkoa, mutta pettämisen vaikutus on silti merkittävä. Esimerkiksi jos puhelinlinja hajoaa se ei suoranaisesti aiheuta vaaraa ihmisille tai ympäristölle, mutta jos ihminen ei tämän järjestelmän pettämisen seurauksena saa yhteyttä hätäkeskukseen voi järjestelmän pettäminen aiheuttaa kuoleman tai muuta merkittävää haittaa ihmisille tai ympäristölle. Tästä johtuen yhteiskunnan kriittinen infrastruktuuri ja niihin liittyviä järjestelmiä pidetään turvallisuuskriittisinä (Knight, 2002, ss. 547–548).

2.1.1 Turvallisuuskriittiset järjestelmät yhteiskunnassa

Turvallisuuskriittiset järjestelmät ympäröivät meitä kaikkialla. Niitä on kotona, työpaikoilla, kouluissa ja vapaa-ajan keskuksissa turvaamassa ihmisiä, omaisuutta ja ympäristöä. Laajemmassa perspektiivissä turvallisuuskriittiset järjestelmät nivoutuvat arkeemme, mutta ne ovat myös laajemmin yhteiskunnan elintärkeiden toimintojen takana. Yhteiskunnan elintärkeitä toimintoja Suomessa ovat henkinen kriisinkestävyys, johtaminen, kansainvälinen- ja EU-toiminta, sisäinen turvallisuus, talous, infrastruktuuri ja huoltovarmuus, puolustuskyky ja väestön toimintakyky ja palvelut (*Elintärkeät toiminnot – Turvallisuuskomitea*, 29.10). Näiden elintärkeiden toimintojen taustalla toimii useita järjestelmiä, jotka mahdollistavat turvallisen arjen suomalaisessa yhteiskunnassa. Suomessa yhteiskunnan elintärkeiden toimintojen suojaaminen tapahtuu kokonaisturvallisuuden toimintamallilla, jossa yhteiskunnan elintärkeät toiminnot turvataan viranomaisten, järjestöjen ja elinkeinoelämän yhteistoimintana (Mattsson, 2013, s. 26).

Yhteiskunnan elintärkeisiin toimintoihin ja niiden turvaamiseen liittyy olennaisesti yhteiskunnan kriittinen infrastruktuuri. Tietoja Suomen kokonaisturvallisuudesta-raportissa (2013) kriittinen infrastruktuuri määritellään seuraavasti: Kriittinen infrastruktuuri ”…käsittää ne rakenteet ja toiminnot, jotka ovat välttämättömiä yhteiskunnan jatkuvalle toiminnalle. Kriittiseen infrastruktuuriin kuuluu sekä fyysisiä laitoksia ja rakenteita että sähköisiä toimintoja ja palveluja.” Suomessa kriittinen infrastruktuuri on kehittynyt monimutkaiseksi ja toisistaan riippuvaiseksi. Kriittisen infrastruktuurin osat eivät myöskään kuulu vain yhdelle omistajalle, vaan ne ovat niin valtion, kuin yksityisten yritysten omistuksessa. Tämän takia kriittisen infrastruktuurin turvaamisessa täytyy siviiliväestön, liike-elämän ja valtion toimia yhteistyössä (Mattsson, 2013, s. 79) Turvallisuuskriittiset järjestelmät linkittyvät olennaisesti kriittiseen infrastruktuuriin. Infrastruktuurin kriittisyyden takia niissä toimivat järjestelmät ovat usein turvallisuuskriittisiä. Kriittisen infrastruktuurin taustalla toimii usein monia erilaisia turvallisuuskriittisiä järjestelmiä, jotka turvaavat ihmisiä ja laajemmin yhteiskuntaa. Kriittisen infrastruktuurin riippuvuus tietoteknisistä järjestelmistä on kasvanut ja näiden sähköisten toimintojen ja palveluiden turvaamiseen liittyy olennaisesti kyberuhat ja niiden torjuminen. (Mattsson, 2013, s. 79)

2.1.2 Kyberturvallisuus turvallisuuskriittisissä järjestelmissä

Mattson kirjoittaa Maanpuolustuskorkeakoulun raportissa Tietoja Suomen kokonaisturvallisuudesta (2013), että ”Yhteiskuntamme elintärkeät toiminnot ovat riippuvaisia turvallisesta ja toimintavarmasta kyberympäristöstä.” Raportin mukaan kyberuhat ovat yksi suurimmista uhkista yhteiskunnan kokonaisturvallisuudelle. Jos kriittisissä järjestelmissä esiintyy häiriöitä, ne voivat synnyttää laajoja häiriötilanteita, joilla voi olla vaikutus koko yhteiskuntaan. (Mattsson, 2013, s. 109) Tällainen häiriö voisi tapahtua esimerkiksi tietoliikenne infrastruktuurissa, jolla olisi laaja-alaisia vaikutuksia yhteiskunnan toiminnalle. Suomi on tietoyhteiskunta ja sitä kautta riippuvainen tietoverkkojen ja -järjestelmien toiminnasta (Lehto & Limnéll, 2017, s. 207).

Tällä hetkellä Suomessa järjestelmiä kehitettäessä tietoturvallisuus on tunnistettu pakolliseksi ominaisuudeksi, mutta käytäntönä on toteuttaa ne erillisinä järjestelmän osina (Valtioneuvostonkanslia, 2016, s. 49). Valtioneuvostonkanslia raportissaan ”Kyberosaaminen Suomessa” (2016) nostaa esille, että uusissa järjestelmissä tietoturvallisuus pitäisi olla sisäänrakennettu ominaisuus, jota tukee security-by-design – ajattelumalli. Tietoturvatietoisuuden kasvattaminen niin yrityksissä, kuin myös tavallisten kansalaisten keskuudessa on raportin mukaan myös tärkeää, kun halutaan kasvattaa yhteiskunnan kokonaisvaltaista kyberturvallisuutta ja resilienssiä. Käyttöliittymäsuunnittelun näkökulmasta tämä tarkoittaa sitä, että tietoturvaominaisuudet ovat sisäänrakennettuja järjestelmiin ja ne ovat mahdollisimman helppokäyttöisiä tavalliselle käyttäjälle.

Tietoturvaketjun heikoin lenkki on ihminen ja käyttäjän manipulaatio on yksi suurimmista kyberuhista, joka kohdistuu tietojärjestelmiin (Salahdine & Kaabouch, 2019). Käyttäjän manipulaatio asettaa käyttöliittymäsuunnittelulle erityisiä haasteita ja suunnittelijat joutuvat miettimään miten turvallisuuskriittisissä järjestelmissä voidaan minimoida käyttäjän manipulaation mahdollisuus ja siten suojata kriittistä järjestelmää.

2.1.3 Turvallisuuskriittisten järjestelmien pettäminen

Järjestelmät saattavat pettää tai vioittua ja turvallisuuskriittisten järjestelmien näkökulmasta tällaisella tapahtumalla saattaa olla merkittäviä seuraamuksia. Kuten aiemmin määriteltiin turvallisuuskriittisen järjestelmän pettäminen voi aiheuttaa ihmishenkien menetyksen tai merkittävää haittaa omaisuudelle ja ympäristölle. Esimerkiksi, jos lentokone putoaa kriittisen järjestelmän pettämisen johdosta, siitä voi seurata matkustajien ja henkilökunnan kuolema ja merkittävä haitta ympäristölle.

Kriittiset järjestelmät voidaan jakaa kahteen ryhmään perustuen niiden virheen sietokykyyn (Bozzano & Villafiorita, 2010, s. 5).

- *Fail-operational:* Järjestelmien tulee toimia normaalisti vaikka osa järjestelmästä olisi pettänyt tai vioittunut. Esimerkki Fail-operational järjestelmästä on lentokone, joka pystyy laskeutumaan turvallisesti ilman moottoreita.
- *Fail-safe:* Järjestelmät ovat suunniteltu sammumaan tai muuntamaan itsensä turvalliseksi jos yksi tai useampi järjestelmän komponentti hajoaa. Esimerkkinä fail-safe järjestelmästä on ydinvoimala reaktori, joka sammuu ja muuttuu turvalliseksi, jos tunnistetaan järjestelmän tai laitteiston vioittuminen.

Turvallisuuskriittisen järjestelmän pettäessä tai hajotessa taustalla on usein jokin seuraavista ongelmista (Bozzano & Villafiorita, 2010, ss. 5–6):

- *Puutteellinen tai kykenemätön järjestelmä.* jos esimerkiksi järjestelmän kehitysvaiheessa jokin ominaisuus jää lisäämättä, jonka seurauksena järjestelmä ei pysty suorittamaan sille annettua tehtävää.
- *Järjestelmä ylikuormittuu.* Järjestelmä saattaa joutua ympäristöön johon sitä ei ole suunniteltu. Tämä saattaa aiheuttaa järjestelmän vioittumisen. Esimerkkinä lentokone, joka joutuu lentämään niin kovassa myrskyssä, että sen järjestelmät vioittuvat.
- *Vaihtelevuus tuotannossa.* jos fyysisen järjestelmän tuotannossa ilmenee vaihtelevuuksia, ne saattavat aiheuttaa ennalta-arvaamattomia vaikutuksia järjestelmän toimintaan.
- *Kuluminen.* Järjestelmän fyysiset osat saattavat kulua ajan myötä. Tämä saattaa aiheuttaa ongelmia järjestelmän toiminnassa.

- *Virheet.* Tietojärjestelmien näkökulmasta virheillä tarkoitetaan yleensä ohjelmistokehityksessä tapahtuvia tahattomia virheitä. Virheitä voi tapahtua myös esimerkiksi fyysisen laitteen asennus- tai kokoamisvaiheessa.

Turvallisuuskriittisiä järjestelmiä kehitettäessä on tärkeää ottaa huomioon mitä mahdollisia seuraamuksia järjestelmän pettämisestä voi seurata. Turvallisuuskriittisten järjestelmien tulee kestää käyttäjien tekemiä virheitä, osata palautua niistä ja pitää järjestelmä toiminnassa. Turvallisuuskriittisten järjestelmien kehittäjien tulisi myös seurata tarkasti määriteltyjä turvallisuusvaatimuksia, jotta järjestelmä on kehitetty niin turvalliseksi kuin se on mahdollista.

3 KÄYTTÖLIITTYMÄT JA NIIDEN YLEISET SUUNNITTELUPERIAATTEET

Tämä kappale käsittelee ihmisen ja teknologian välisen vuorovaikutuksen keskeisimmät termit ja käsitteet, jotka ovat kaikkein merkityksellisimpiä tutkielman tutkimuskysymyksen kannalta. Tarkastellaan myös millaisia suunnitteluperiaatteita kirjallisuudessa on esitelty ja millä tavoin käyttöliittymiä tulisi kirjallisuuden mukaan arvioida, kun luodaan mahdollisimman hyviä järjestelmiä.

Aihe on tutkimuksen kannalta merkittävä, koska esimerkiksi terveydenhuollon järjestelmissä on perinteisesti jouduttu kärsimään heikosta käyttäjäkokemuksesta ja huonoista käyttöliittymistä verraten muiden alojen käyttämiin järjestelmiin (Goswami & Gitta, 2018). Hyvän käyttäjäkokemuksen ja käyttöliittymän pitäisi olla ensisijaisen tärkeää terveydenhuollon järjestelmiä suunnitellessa, koska niillä voi olla suora vaikutus potilaiden hyvinvointiin. Hyvin suunniteltu käyttäjäkokemus ja käyttöliittymä pienentää virheiden mahdollisuutta ja siten vähentää potilaisiin kohdistuvaa riskiä (Goswami & Gitta, 2018). Hyvin suunnitellut järjestelmät auttavat myös muita turvallisuuskriittisiä aloja minimoimaan ihmisiin ja ympäristöön kohdistuvaa riskiä. Sen vuoksi kaikkien, jotka valmistavat turvallisuuskriittisiä järjestelmiä, tulisi ottaa ihmisen ja teknologian välinen vuorovaikutus huomioon järjestelmiä suunnitellessa ja valmistettaessa.

3.1 Käyttäjäkokemus

Käyttäjäkokemukseen määritellään kaikki ne aspektit, miten käyttäjä on vuorovaikutuksessa jonkin tuotteen tai palvelun kanssa. Siihen liittyy muun muassa miltä tuote tuntuu, miten hyvin käyttäjä ymmärtää sen miten tuotetta käytetään, miltä käyttäjästä tuntuu tuotetta käytettäessä ja kuinka hyvin tuote palvelee käyttäjän tarpeita. Jos nämä kaikki kokemukset ovat onnistuneita voidaan tuotteella sanoa olevan hyvä käyttäjäkokemus (Alben, 1996, s. 12). Käyttäjäkokemusta ja käytettävyyttä ei pidä sekoittaa keskenään, vaikka ne kuulostavat melko samantyyppisiltä termeiltä. Käyttäjäkokemus sisältää tuotteen tai järjestelmän käytettävyyden, mutta se koskettaa myös muita osia tuotteesta tai järjestelmästä.

Käyttäjäkokemukseen panostamisella voidaan saavuttaa muun muassa seuraavia hyötyjä (Zink, 2017):

- *Asiakashankinta*: hyvä käyttäjäkokemus antaa kilpailuedun ja sitä kautta tuo liiketoiminnallista hyötyä.
- *Asiakkaiden säilyminen:* kun järjestelmä on suunniteltu intuitiiviseksi ihmiset haluavat käyttää sitä ja jatkavat sen käyttämistä.
- *Pienemmät tukikulut:* Järjestelmä, joka toimii ja jota on helppo käyttää pienentää tarvetta koulutukselle, dokumentaatiolle ja käyttäjätuelle. Nämä johtavat säästöihin, josta on liiketoiminnallista hyötyä.
- *Tehokkuus kasvaa:* Parempi käyttäjäkokemus johtaa parempaan tehokkuuteen, kun huomioidaan käyttäjien määrä ja käytetyt tunnit järjestelmän parissa.

Kun puhutaan turvallisuuskriittisistä järjestelmistä, täytyy kuitenkin huomioida että ne ovat usein järjestelmiä, jotka ovat monimutkaisia ja niiden käyttäjillä ei ole mahdollisuutta vaihtaa toiseen järjestelmään jos nykyisessä järjestelmässä havaitaan puutteita tai ongelmia. Jos käyttäjäkokemukseen ei panosteta riittävästi, voi seurauksena olla esimerkiksi luottamuksen rapautuminen käytettyä järjestelmää kohtaan. McCarthy ja Wright (2005) nostavat esille esimerkin sairaanhoitajista, jotka eivät luottaneet sairaalassa käyttöönotettuun järjestelmään, vaikka se olisi tuonut heille merkittäviä etuja muun muassa työn tehokkuuteen. He sen sijaan jatkoivat manuaalisen järjestelmän käyttöä, koska he tunsivat ahdistusta heidän digitaalisesta kompetenssista, he eivät luottaneet muiden syöttämän tiedon tarkkuuteen tai laatuun ja he kokivat, että digitaalisen tietojärjestelmän käyttäminen vie heidät pois potilaiden luota, joka vaikuttaisi heidän suhteeseen potilaiden kanssa (McCarthy & Wright, 2005, s. 266). Tämä on yksi monista esimerkeistä miksi järjestelmien käyttäjäkokemukseen tulisi panostaa, jotta uudet järjestelmät olisivat käyttäjille mieluisia.

3.2 Käytettävyys

Nielsen määrittelee käytettävyyden attribuutiksi, joka mittaa miten helppoa käyttöliittymää on käyttää. Käytettävyyttä voidaan mitata Nielsenin mukaan viidellä komponentilla (Nielsen, 2012):

- *Opittavuus:* kuinka helppoa käyttäjälle on suorittaa tavanomainen tehtävä ensimmäisellä kerralla, kun hän käyttää järjestelmää.
- *Tehokkuus:* kun käyttäjä on oppinut käyttämään järjestelmää, kuinka nopeasti hän pystyy suorittamaan tehtäviä.
- *Muistettavuus:* jos käyttäjä pitää tauon järjestelmän käytöstä ja palaa sitten käyttämään sitä kauanko hänellä kestää tulla taitavaksi sen käytössä.

- *Virheet:* kuinka paljon virheitä käyttäjät tekevät, kuinka vakavia nämä virheet ovat ja miten helposti virheistä voidaan palautua.
- *Tyytyväisyys:* kuinka mukavaa järjestelmää oli käyttää.

Nielsen kertoo käytettävyyteen liittyvän olennaisesti myös järjestelmän hyödyllisyys. Vaikka järjestelmä olisi käyttäjälleen todella helppo käyttää ja oppia ei siitä kuitenkaan ole paljoa hyötyä, jos se ei auta käyttäjää tekemään jotain hänelle merkityksellistä (Nielsen, 2012).

Käytettävyys on tärkeä huomioida suunnittelussa. Esimerkiksi jos verkkosivun käytettävyys on huono, se voi johtaa siihen, että ihmiset eivät löydä sieltä tietoa, he hukkuvat sivulle, sivujen sisältö on vaikeasti ymmärrettävää tai jos jokin muu käytettävyyteen liittyvä elementti ei ole kunnossa, käyttäjä luultavasti poistuu sivulta ja etsii tiedon jostain muualta. Jos kyseessä on yrityksen sisäinen järjestelmä, voi huono käytettävyys johtaa työntekijöiden tuottavuuden laskuun, koska he käyttävät työaikaansa tiedon löytämiseen, joka huonon käytettävyyden takia on haastavaa (Nielsen, 2012). Käytettävyydellä on myös merkittävä vaikutus, kun suunnitellaan turvallisuuskriittisiä järjestelmiä. Nielsen esittää useita esimerkkejä sille, miten huono suunnittelu saattaa johtaa käyttäjän kuolemaan ja merkittävään vahinkoon ympäristölle tai omaisuudelle. Mainitaan esimerkki autojen keskikonsolin käyttöliittymistä. Nielsen (2005) esittää, että tuhannet kuolemat vuodessa johtuvat siitä, että kuljettajat ovat keskittyneet monimutkaiseen järjestelmään ajamisen sijaan. Nielsen (2005) esittelee toisen tutkimuksen, joka käsitteli sairaalassa käytettyä lääkkeen tilausjärjestelmää, jolla määriteltiin potilaille annetut lääkkeet. Tutkimus osoitti 22 tapaa, miten järjestelmä saattoi ohjata potilaalle väärän lääkkeet. Nielsen kertoo, että suurin osa näistä johtui järjestelmän heikosta käytettävyydestä. Esimerkiksi huono luettavuus ja liian monimutkainen työnkulku olivat pääsyinä sille, että potilaat saivat vääriä lääkkeitä (Nielsen, 2005).

Järjestelmän käytettävyyttä voidaan parantaa muun muassa haastattelemalla sen käyttäjiä. Tätä metodia kutsutaan käyttäjätestaamiseksi. Käyttäjätestaamisen ajatuksena on antaa jokin järjestelmä testaajan käyttöön ja seurata miten käyttäjä on vuorovaikutuksessa sovelluksen kanssa. Testaus voi tapahtua haastattelu-tyyppisesti tai verkon välityksellä. Käyttäjätestaamisessa yleensä halutaan saada selville, onko järjestelmän käytettävyys hyvällä tasolla, ratkaiseeko se käyttäjälle jonkin ongelman tai soveltuuko se käyttäjän arkeen tai rutiineihin. Kun haastatellaan käyttäjää, on ensiarvoisen tärkeää päästä ymmärtämään käyttäjän ajatuksia ja kuulemaan mitä käyttäjä ajattelee, kun hän suorittaa jotain tehtävää. Käyttäjältä voidaan kysyä kysymyksiä kuten ”voitko kuvailla mitä näet?”, ”jos haluaisit vaihtaa salasanan, miten tekisit sen?” Tämän tyyppisillä avoimilla kysymyksillä saadaan käyttäjä kertomaan mitä hän ajattelee, joka tarjoaa tärkeää tietoa järjestelmän käytettävyydestä. Testauksen aikana saadaan paljon pieniä kehitysehdotuksia, joita ei välttämättä ole aiemmin edes harkittu. Tämä on yksi monista tavoista kehittää järjestelmien käytettävyyttä.

3.3 Käyttöliittymät

Galitz (2007) määrittelee käyttöliittymän osaksi tietokoneen ohjelmistoa, jonka kanssa ihminen on tekemisissä erilaisilla tavoilla: näkemällä, kuulemalla, koskettamalla, puhumalla tai muuten ohjaamalla ja ymmärtämällä. Galitzn mukaansa käyttöliittymissä on aina kaksi tärkeää osaa: syöte ja tuloste. Syötteellä tarkoitetaan kaikkia tapoja joilla käyttäjä kommunikoi tietokoneelle. Käyttäjä voi syöttää tietokoneelle sisältöä esimerkiksi näppäimistön, hiiren tai puheen avulla. Tulosteella tarkoitetaan tietokoneen tuottamia tuloksia käyttäjän syötteen perusteella. Tällä hetkellä yleisin tapa millä tietokone esittää tulosteen käyttäjälle, on näyttö tai monitori (Galitz, 2007, s. 4). Uusien teknologioiden myötä on myös muodostunut uusia tapoja on muodostunut syöttää ja tulostaa tietoa. Esimerkiksi puheohjauksen avulla voidaan ohjata älypuhelimia sekä tietokoneita ja useat älylaitteet pystyvät tulostamaan käyttäjälle palautetta muun muassa värinän ja äänien avulla. Galitz nostaa esille parhaan käyttöliittymän olevan sellainen, joka ei asetu käyttäjän tielle, vaan antaa käyttäjän keskittyä suoritettavaan tehtävään ja informaatioon mahdollisimman tehokkaasti sen sijaan, että käyttöliittymä vaatisi käyttäjän keskittymistä erilaisiin ohjausmekanismeihin ja ylimääräisiin elementteihin (Galitz, 2007, s. 4) Kun määritellään sanaa käyttöliittymä on huomioitava erilaiset käyttöliittymien tyypit. Voidaan puhua komentorivi- tai graafisista käyttöliittymistä. Tässä tutkielmassa keskitytään jälkimmäiseen eli graafiseen käyttöliittymään, koska niissä käyttöliittymän muotoilulla on merkittävästi suurempi vaikutus käyttäjän toimintaan verrattuna komentoriviin. Graafisessa käyttöliittymässä käyttäjälle on mahdollista esittää monia erilaisia sisältötyyppejä, kuten tekstiä, kuvia, videoita, grafiikkaa ja animaatioita. Komentorivi-käyttöliittymässä kaikki sisältö on tekstiä.

3.4 Käyttöliittymien yleiset suunnitteluperiaatteet

Käyttöliittymien suunnittelua voidaan pitää luovana prosessina ja välttämättä kaikkiin tilanteisiin sopivia suunnitteluperiaatteita ei ole olemassa, mutta kirjallisuudessa on esitelty useita erilaisia periaatteita joita seuraamalla on mahdollista luoda toimivia ja hyödyllisiä käyttöliittymiä. Näihin periaatteisiin voi lisätä suunnittelijan omaa luovuutta, jolla voidaan erottautua kilpailevista järjestelmistä ja jolla voidaan luoda turvallisuuskriittisyyden näkökulmasta turvallisia ja tehokkaita käyttöliittymiä. Esitellään seuraavaksi kirjallisuudessa esitettyjä periaatteita, joita voidaan seurata kun suunnitellaan käyttöliittymiä.

3.4.1 Schneidermanin kultaiset säännöt

Schneiderman esittelee kahdeksan kultaista sääntöä sille miten suunnitellaan käyttöliittymä, joka mahdollistaa tehokkaan vuorovaikutuksen ihmisen ja tietokoneen välillä (Mazumder & Das, 2014). Käyttöliittymä suunnittelun kahdeksan sääntöä esimerkeillä ovat seuraavat:

- *Pyri johdonmukaisuuteen:* Kaikki toiminnot ja elementit tulisi suunnitella samalla tavalla, jotta ne ovat johdonmukaisia. Esimerkkinä voidaan pitää MacOS-käyttöjärjestelmän ylävalikkoa, joka pysyy aina samassa paikassa tilanteesta riippumatta.
- *Mahdollista oikoteiden käyttö:* Anna käyttäjälle mahdollisuus käyttää oikoteitä, jotka nopeuttavat tavallisten tehtävien tekemistä. Oikoteiden käyttämisessä pidän parhaana esimerkkinä Superhuman sähköposti sovellusta, jossa kaikki toiminnot voidaan suorittaa ilman hiirtä. Jos haluat kirjoittaa uuden viestin paina C, jos haluat lähettää viestin paina Cmd + Enter.
- *Tarjoa informatiivista palautetta:* Järjestelmän tulisi antaa jonkinlaista palautetta kaikista käyttäjän tekemistä toimista. Palaute voi olla muun muassa ääni, visuaalinen muutos tai laitteen tuottama värinä. Esimerkkinä palautteesta voisi olla iOS-käyttöjärjestelmässä viestin lähettäminen. Kun painat lähetä nappia järjestelmä antaa pienen äänihuomautuksen, kun viesti on lähetetty.
- *Dialogi kun tehtävä suoritetaan:* Tarjoa käyttäjälle selkeä tieto, kun jokin tehtävä on saatu suoritettua. Esimerkkinä toimii MacOS-käyttöjärjestelmän roskakorin tyhjennys. Kun painat "Tyhjennä roskakori"-painiketta järjestelmä näyttää ikkunan, jossa näet prosessin etenemisen visuaalisesti. Kun roskakori on tyhjennetty ikkuna katoaa ja järjestelmä tekee äänimerkin.
- *Tarjoa yksinkertaista virheiden hallintaa:* Suunnitellaan järjestelmä sellaiseksi, että käyttäjän on vaikeaa tehdä katastrofaalista virhettä. Jos käyttäjä tekee virheen täytyy järjestelmän tunnistaa se, tarjota yksinkertainen selitys sille miksi virhe tapahtui ja antaa ohje miten virheestä voidaan palautua.
- *Salli toimien peruminen:* Jos käyttäjä tekee virheen, käyttäjälle tulisi antaa mahdollisuus peruuttaa se ja yrittää uudelleen. Tämä rohkaisee käyttäjää kokeilemaan ominaisuuksia ilman pelkoa siitä, että järjestelmä hajoaa. Esimerkkinä toimii Windows- ja MacOS-järjestelmien Cmd/Ctrl + Z komento, jolla voidaan peruuttaa yksittäinen komento tai sarja komentoja.
- *Tuetaan hallinnan tunnetta:* Järjestelmä tulisi suunnitella siten, että sen käyttäjä tuntee olevansa järjestelmänhallitsija ja järjestelmä vastaa käyttäjän komentoihin. Esimerkkinä toimii MacOS-järjestelmän "Pakota lopettamaan"-toiminto, joka antaa käyttäjälle mahdollisuuden sammuttaa minkä tahansa sovelluksen, jos sovellus lopettaa toimintansa.
- *Vähennä lyhyenaikavälin muistitaakkaa:* ihminen pystyy käsittelemään vain tietyn verran dataa kerrallaan, joten järjestelmä tulisi suunnitella siten, että sen esittämä tieto pysyy tiiviinä ja relevanttina kyseiseen hetkeen ja tehtävään sopivaksi.

3.4.2 Nielsenin heuristiikat

Nielsen esitteli vuonna 1994 kymmenen heuristista ohjetta, jotka toimivat tähän päivään saakka yhtenä alan johtavista ohjeista käyttöliittymä ssuunnittelulle. Nielsenin säännöt muistuttavat Schneidermanin kultaisia sääntöjä, mutta niissä on kuitenkin havaittavissa eroja. Nielsen mainitsee, että sääntöjä kutsutaan heuristiikoiksi siitä syystä, että ne ovat enemmänkin suuntaviivoja suunnittelun tueksi (Nielsen, 1994b). Nielsenin heuristiikat tarjoavatkin suunnittelijalle apua, ei niinkään tarkkoihin muotoilukysymyksiin vaan siihen, että suunniteltu järjestelmä on mahdollisimman tehokas ja helposti käytettävä loppukäyttäjälle. Nielsenin kymmenen heuristista ohjetta ovat seuraavat (Nielsen, 1994b):

- *Järjestelmän tilan näkyvyys:* järjestelmän tulisi aina pitää käyttäjä ajan tasalla siitä mitä tapahtuu, oikea-aikaisella palautteella ja viesteillä
- *Käyttäjälle tuttu esitystapa:* järjestelmän tulee puhua käyttäjän kieltä ja järjestelmän täytyy hyödyntää konsepteja, jotka ovat käyttäjälle tuttuja oikeasta maailmasta.
- *Käyttäjän hallinta ja vapaus:* Käyttäjät valitsevat joskus vääriä toimintoja jolloin heille tulisi tarjota helppo hätäuloskäynti. Tärkeää on tukea peruuttamista ja ”undo”-toimintoa.
- *Johdonmukaisuus ja standardien noudattaminen:* Käyttäjien ei pitäisi joutua miettimään, tarkoittaako eri sanat käyttöliittymässä samaa asiaa. Esimerkiksi jos käyttöliittymässä on kaksi nappia, joista toinen sanoo ”tallenna” ja toinen ”vahvista”, niin käyttäjälle ei välttämättä käy selväksi toimivatko napit samalla tavalla
- *Virheiden estäminen:* hyvien virheviestien lisäksi tulisi järjestelmä suunnitella siten, että se pystyy estämään käyttäjää tekemästä katastrofaalista virhettä ja tarvittaessa ohjata käyttäjää oikeaan suuntaan.
- *Tunnistettavuus muistettavuuden sijaan:* Tarkoituksena on vähentää käyttäjän muistitaakkaa tekemällä objektit ja toiminnot näkyviksi. Käyttäjän ei tarvitse muistaa edellisessä vuorovaikutuksessa esiteltyä toimintoa. Käyttöohjeet pitäisi olla helposti saavutettavissa, aina kun niille on tarvetta.
- *Joustavuus ja käytön tehokkuus:* ammattilaiskäyttäjille suunnatut tehokkuutta lisäävät toiminnot, jotka voidaan ottaa käyttöön ja muokata käyttäjän mieltymysten mukaan.
- *Esteettinen ja minimalistinen muotoilu:* Valintaikkunoiden ei pitäisi sisältää sellaista tietoa, joka ei ole hyödyllistä tai että se on harvoin tarvittua. Jos ylimääräistä informaatiota lisätään, se saattaa viedä huomiota pois käyttäjälle oikeasti merkitykselliseltä informaatiolta.
- *Auta käyttäjiä tunnistamaan, diagnostisoimaan ja palautumaan virheistä:* Virhe viestit tulisi esittää selkeällä kielellä ilman vaikeita koodeja. Niiden tulisi tarkasti kertoa ongelma ja ehdottaa ratkaisua, jonka käyttäjä pystyy itse toteuttamaan.
- *Tuki ja dokumentaatio:* järjestelmälle olisi toivottavaa, että sitä voisi käyttää ilman ohjeita ja dokumentaatiota, mutta dokumentaation pitäisi olla

helposti saavutettavissa ja sen pitäisi selkeillä askelilla keskittyä ratkaisemaan käyttäjän sen hetkinen ongelma.

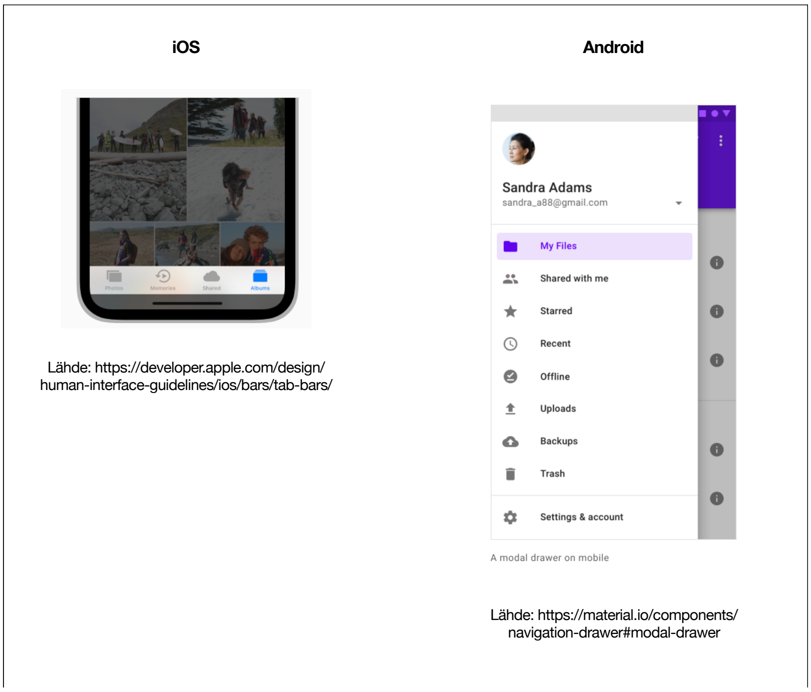
3.4.3 Käyttöjärjestelmien omat suunnitteluperiaatteet

Järjestelmien ja sovellusten käyttöliittymä suunnittelua ohjaa nykyään myös eri käyttöjärjestelmä alustojen omat käyttöliittymä suunnitteluperiaatteet. Nämä erilaiset suunnittelu ohjeet erottavat alustat selkeästi toisistaan ja nämä ohjeet tekevät kustakin käyttöjärjestelmästä tunnistettavan. Seuraavat suunnittelu periaatteet ovat eniten käytetyt ja helpoiten tunnistettavat:

- *Human Interface Guidelines, Apple:* Human interface guidelines on Applen luoma suunnitteluohjeistus, jonka mukaan suunnitellaan Apple-laitteille suunnatut sovellukset ja järjestelmät.
- *Material Design, Google:* Material design on Googlen luoma suunnitteluohjeistus, jonka mukaan useat Android-sovellukset ja Googlen-verkkopalvelut on suunniteltu.
- *Fluent Design Systen, Microsoft:* Fluent on Microsoftin luoma suunnitteluohjeistus, johon on pohjautunut muun muassa Windows-käyttöjärjestelmä ja Microsoft Office-sovellukset.

Kun eri alustoille suunnitellaan järjestelmiä ja sovelluksia täytyy huomioida järjestelmän käyttämät periaatteet. Käyttäjä saattaa olla tottunut esimerkiksi iOS-järjestelmän interaktioihin ja tyyleihin, jolloin jos hänelle esitetään Android-käyttöliittymä voi käyttäjällä olla merkittäviä haasteita ymmärtää järjestelmän toimintaperiaatetta.

Esimerkiksi kun suunnitellaan tavallinen valikko- tai navigaatiorakenne on iOS- ja Android-järjestelmissä suuri ero. iOS-valikko on suunniteltu staattiseksi aina näytön alareunassa näkyväksi valikoksi, josta käyttäjä voi navigoida haluamalleen sivulle sovelluksen sisällä. Android-valikko on piilotettu käyttäjältä ennen kuin käyttäjä painaa nappia tai vetää näytön vasemmasta reunasta paljastaen valikon, josta pystyy navigoimaan sovelluksen eri sivuille. Kuten kuvasta 1 huomaa Applen- ja Googlen-käyttöliittymät eroavat toisistaan ja ovat helposti tunnistettavissa muun muassa ikonien, tekstityylien ja värien mukaan.



Kuva 1: Valikkojen esitys tyyli iOS- ja Android-järjestelmissä

3.5 Käyttöliittymien arviointi ja testaaminen

Kaikkia järjestelmiä voidaan arvioida ja testata ja siksi myös käyttöliittymät tulisi arvioida ja testata, jotta ne täyttävät järjestelmälle asetetut kriteerit ja vaatimukset. Kirjallisuudessa graafisen käyttöliittymän testaamisesta puhutaan GUI-testauksena (Graphical User Interface). GUI-testaus on ohjelmiston testaamista, jossa on graafinen käyttöliittymä (Banerjee ym., 2013, s. 2).

Käyttöliittymiä voidaan testata tekemällä niille heuristinen arvio, jossa tutkittavaa käyttöliittymää verrataan Nielsenin heuristiikkoihin. Heuristinen arviointi toteutetaan Nielsenin (1994) mukaan seuraavasti: arvio toteutetaan pienen testaajaryhmän toimesta, jotka tutkivat käyttöliittymää ja arvioivat sen yhteensopivuutta Nielsenin heuristiikkojen kanssa. Arvion tekemiseen tarvitaan lähes kaikissa tilanteissa ryhmä, koska yksi ihminen ei välttämättä tunnista kaikkia käyttöliittymän sisältäviä käytettävyysongelmia. Tehtävään valitut testaajat toteuttavat arvion yksin ja vasta sitten kun kaikki testaajat ovat tehneet arvionsa, saavat he keskenään kommunikoida ja keskustella tuloksista. Yksilöllinen testaaminen on tärkeää, jotta arvio pysyy puolueettomana ja toisten arvioijien ajattelu ei vaikuta yksittäisen arvioijan toimintaan. Arvioinneille voidaan asettaa valvoja, jonka tehtävänä on avustaa testaajia ongelma tilanteissa ja kerätä yhteen muistiinpanot ja esille tulleet ongelmat. Valvojan käyttäminen ei ole Nielsenin mukaan pakollista, mutta se auttaa nopeuttamaan palauteprosessia, jotta arvioinnin tulokset ovat välittömästi suunnittelutiimin saatavilla (Nielsen, 1994a). Heuristinen arvio eroaa tässäkin tutkielmassa aiemmin mainitusta käyttäjätestaamisesta siten, että heuristisessa arviossa valvoja voi vastata testaajan kysymyksiin kesken arvioinnin ja testaajalle voidaan näyttää vihjeitä miten käyttöliittymää tulisi käyttää. Nielsen huomauttaa kuitenkin, että apua tulisi tarjota vasta sitten, kun testaaja on tuonut esille käytettävyysongelman ja kun testaaja on tilassa, josta hän ei osaa edetä. Arviointitilaisuus kestää yleensä tunnista kahteen tuntiin, jonka aikana testaaja käy läpi koko käyttöliittymän useita kertoja tarkastaen ja verraten sitä Nielsenin heuristiikkoihin. Testaaja voi kuitenkin ehdottaa myös heuristiikkojen ulkopuolisia käytettävyysongelmia. Arvioinnin tuloksena on lista käytettävyysongelmia, jotka viittaavat Nielsenin heuristiikkoihin. Tulosten perusteella voidaan mahdollisuuksien mukaan suunnitella uusi paranneltu käyttöliittymä, joka keskittyy näiden vikojen korjaamiseen. Arvioinnin jälkeen voidaan pitää myös keskustelu testaajan, valvojan ja suunnittelijan välillä, jossa löydettyihin käytettävyysongelmiin pyritään löytämään ratkaisut (Nielsen, 1994a).

Nielsenin heuristiikkojen arviointi keskittyy pääasiallisesti käytettävyyteen, joten käyttöliittymän testaamisessa olisi tärkeää huomioida myös muut osa-alueet käytettävyyden lisäksi. Käyttöliittymien testaamisessa pitäisi huomioida visuaalinen muotoilu, käytännöllisyys, käytettävyys, tehokkuus ja ohjeidenmukaisuus (Podoler, 2020).

4 TURVALLISUUSKRIITTISTEN JÄRJESTELMIEN SUUNNITTELU

Tässä luvussa kerrotaan mitä asioita turvallisuuskriittistä järjestelmää suunnitellessa täytyy ottaa huomioon. Ennen huomioitavia asioita käsitellään ensin turvallisuuskriittisten järjestelmien suunnittelun haasteita.

4.1 Turvallisuuskriittisten järjestelmien suunnittelun haasteet

Turvallisuuskriittisiä järjestelmiä suunnitellessa kohdataan useita haasteita, joita asettavat niiden käyttäjät ja järjestelmän toimintaympäristö. Yleensä järjestelmän turvallisuus saavutetaan useiden järjestelmien yhteistyönä, jotka ovat riippuvaisia eri teknologioista kuten mekaanisista, hydraulisista, sähköisistä, elektronisista tai ohjelmoitavista osista. Tästä johtuen järjestelmän turvallisuusstrategian tulee huomioida ei pelkästään yhden järjestelmän osat, vaan kaikkien turvallisuutta edistävien järjestelmien osat (International Electrotechnical Commission, 2010).

Turvallisuuskriittisten järjestelmien haasteena on usein se, että järjestelmää suunnitellessa joudutaan tekemään valintoja hyvän käytettävyyden ja turvallisuuden välillä. Usein, kun järjestelmä suunnitellaan mahdollisimman turvalliseksi joutuu järjestelmän käyttäjä tekemään enemmän töitä tehtävien suorittamiseksi (Murphy, 1998). Järjestelmää suunnitellessa joudutaan siis myös näkemään enemmän vaivaa sen eteen miten nämä turvalliset, mutta monimutkaiset user flow:t suunnitellaan mahdollisimman helpoiksi ja yksinkertaisiksi käyttää.

Ihminen on suurin riskitekijä turvallisuuskriittiselle järjestelmälle ja ihmisen tekemät virheet ovat aiheuttaneet merkittävän osan turvallisuuskriittisten järjestelmien onnettomuuksista. Järjestelmien suunnittelussa haasteeksi nousee ihmisen toiminnan vaihtelevuuden hallinta. Tämä aiheuttaa järjestelmälle haasteita, koska usein helposti toteutettavat tehtävät ovat korkeasti automatisoitua ja ihmiselle jää kaikkein monimutkaisimmat ja tilannetajua vaativammat tehtävät(Oedewald & Reiman, 2006, s. 17). Korkea automatisointi aiheuttaa kuitenkin niin sanotun out of the loop-ongelman järjestelmän käyttäjälle. Korkeasti automatisoitujen järjestelmien käyttäjillä on havaittu vähentynyt kyky tunnistaa järjestelmässä tapahtuvia virheitä ja virheiden tapahtuessa ottaa manuaalinen hallinta järjestelmästä verrattuna manuaalisesti ohjattavaan samanlaiseen järjestelmään. Out of the loop-ongelma linkittyy kahteen automatisoitujen järjestelmien ilmiöön: käyttäjä ei ohjaa järjestelmää manuaalisesti ja käyttäjällä ei ole riittävää ymmärrystä järjestelmän nykyisestä tilasta ja käynnissä olevasta prosessista (Endsley & Kiris, 1995, s. 2). Tämä johtaa käyttäjän tilannetietoisuuden vähenemiseen, joka altistaa järjestelmän mahdollisille uhkille.

4.2 Turvallisuuskriittisten järjestelmien suunnittelussa huomioitavat asiat

Kuten kaikille järjestelmille on myös turvallisuuskriittisille järjestelmille suunnitteluperiaatteita, joita voidaan seurata kun halutaan maksimoida järjestelmän käytettävyys ja myös sen turvallisuus. Aiemmin tutkielmassa mainitut Schneidermanin kultaiset säännöt ja Nielsenin heuristiikat toimivat myös turvallisuuskriittisten järjestelmien suunnittelun selkärankana. Tämä kappale pyrkii syventämään aiemmin käsiteltyjä suunnitteluperiaatteita ja vastaamaan tutkielman tutkimuskysymykseen.

4.2.1 Järjestelmien vaatimukset ja sertifikaatit

Kappaleessa 2 määriteltiin turvallisuuskriittiset järjestelmät sellaisiksi järjestelmiksi, joiden pettäminen voi aiheuttaa ihmishenkien menetyksen ja merkittävää haittaa omaisuudelle ja ympäristölle. Koska järjestelmien pettämisellä on niin suuret haitat tulisi järjestelmien saavuttaa tietyt vaatimukset ennen kuin sitä aletaan käyttämään.

Yksi tärkeimmistä vaiheista turvallisuuskriittisiä järjestelmiä suunnitellessa on määritellä mitä standardeja ja vaatimuksia kehitystyössä aiotaan seurata (Kraeling, 2014). Koska jokainen turvallisuuskriittinen järjestelmä on erilainen ja jokaiselle järjestelmälle on erilaiset vaatimukset, on hyödyllistä katsoa millainen prosessi tukee vaatimuksien määrittelyä kehitysprosessissa. Vuori (2011) määrittelee mitä vaatimuksia turvallisuuskriittisen järjestelmän kehitys prosessissa tulisi huomioida(Vuori, 2011, ss. 12–13):

- *Ymmärrys riskeistä*: Kehitysprosessissa on välttämätöntä ymmärtää mitä vaaroja ja riskejä järjestelmä saattaa aiheuttaa. Kehittäjien täytyy tehdä analyysi kaikista turvallisuusvaatimuksista sisältäen analyysin järjestelmän todellisesta käyttötarkoituksesta.
- *Laatu*: Järjestelmän kehityksen laaduntarkkailu tulisi olla huipputasolla. Järjestelmän kehittämiseen tulisi osallistua vain ammattilaisia, joilta voidaan olettaa korkeaa osaamista.

- *Hallinta*: Prosessi tarvitsee hyvää johtamista ja hallintaa. On tärkeää huomata, että turvallisuuskriittisen järjestelmän kehitys on äärimmäisen monimutkaista, joten prosessin tulisi tähdätä yksinkertaisuuteen ja selkeyteen aina kun mahdollista. Turvallisuusanalyytikoiden tulisi myös osallistua prosessiin.
- *Analyysi*: turvallisuustiedot tulee dokumentoida ja dokumentaatiota tulee päivittää aina kun järjestelmään tehdään muutoksia.
- *Aika ja resurssit*: kehitykselle täytyy varata riittävästi aikaa, jotta kaikki tehtävät voidaan suorittaa kunnolla. Tällä vältetään se, että tehtäisiin kompromisseja ajanpuutteen takia.
- *Auditointi*: kehitystä tulee voida tarkkailla ja seurata.

International Electrotechnical Comission (IEC) on luonut yleisen standardin EC 61508 jota käytetään turvallisuuskriittisten järjestelmien suunnitteluun, käyttöönottoon ja ylläpitoon (Kraeling, 2014).

Järjestelmän saamat sertifikaatit ovat nykyään tärkeä osa arviointia ja kehitystä. Turvallisuuskriittistä järjestelmää kehittävälle yritykselle tai organisaatiolle on tärkeää saada järjestelmä sertifioitua, jotta voidaan todistetusti sanoa, että tuote täyttää tietyt vaatimukset ja sitä voidaan pitää turvallisena. Sertifiointiprosessi sisältää neljä kokonaisuutta: hakijan, sertifikaatin myöntäjän, standardit jotka vaaditaan sertifikaatin saamiseksi ja arviointi ryhmän, joka toteuttaa arvioinnin. Hakijan täytyy toimittaa niin sanottu sertifikaattipaketti, joka osoittaa että järjestelmä noudattaa valittuja standardeja. Arviointiryhmä käy järjestelmän ja paketin läpi osoittaakseen, että järjestelmä noudattaa vaatimuksia. Jos arviointiryhmä toteaa järjestelmän täyttävän vaatimukset, niin sertifikaatin myöntäjä myöntää järjestelmälle kyseisen sertifikaatin (Pietrantuono & Russo, 2013).

4.2.2 Käyttäjäkeskeiset menetelmät ja käytettävyys turvallisuuskriittisissä järjestelmissä

Aiemmassa kappaleessa mainittiin että järjestelmän turvallisuutta kasvatettaessa muuttuu käyttäjän tehtävät vaikeammiksi (Murphy, 1998). Tätä ilmiötä vastaan tulisi hyödyntää käyttäjäkeskeisiä suunnittelu menetelmiä, joissa käyttäjä on suunnittelun keskipisteenä turvallisuudesta tinkimättä.

Savioja (2003) kertoo käyttäjäkeskeisistä menetelmistä monimutkaisten järjestelmien suunnittelussa. Käyttäjäkeskeiset menetelmät pyrkivät parantamaan suunniteltavien tuotteiden käytettävyys ominaisuuksia. Saviojan (2003) mukaan ”käytettävyys lisää tuotteen hyödyllisyyttä, tehokkuutta ja käyttömukavuutta”. Aiemmin tutkielmassa nostettiin esille Nielsenin 5 tapaa mitata järjestelmän käytettävyyttä, jotka olivat: *mitattavuus, tehokkuus, muistettavuus, virheet* ja *tyytyväisyys* (Nielsen, 2012). Hussey (1999) esittelee turvallisuus-käytettävyyden periaatteet, jotka ovat johdettu perinteisistä käytettävyyden periaatteista, mutta muokattu turvallisuuskriittisille järjestelmille sopiviksi. Hussey:n turvallisuuskäytettävyyden periaatteet ovat (Hussey ym., 1999, ss. 6–8):

- *Tehtävän tehokkuus*: Järjestelmän tulisi auttaa eri tasoisia käyttäjiä minimoimaan tehtävän suorittamiseen tarvitun työn. Esimerkiksi virheenesto-tekniikat ovat yksi tapa miten käyttäjän tehokkuutta voidaan lisätä. Virheen esto voi olla muun muassa toiminnon peruminen.
- *Uudelleenkäytettävyys*: Varmistetaan, että käyttäjä pystyy hyödyntämään jo olemassa olevaa tietoa. Esimerkiksi kuvakkeet ja painikkeet tulisi suunnitella siten, että ne tekevät aina saman toiminnon.
- *Ihmisen ja tietokoneen välinen kommunikaatio*: hallitaan ihmisen ja tietokoneen välistä kommunikaatiota esimerkiksi näyttämällä mikä on järjestelmän tila kullakin hetkellä.
- *Joustavuus*: suunnittelussa täytyy huomioida käyttäjien erilaiset taustat ja mahdollistaa, että tehtävien suorittaminen on mahdollista eri keinoilla ja että järjestelmä voidaan tehdä tulevaisuudessa muutoksia.

4.2.3 Järjestelmien käyttövarmuus

VTT määrittelee raportissaan Käyttövarmuuden hallinta – standardista käytäntöön, käyttövarmuuden seuraavasti ”…käyttövarmuus on kohteen kyky olla tilassa, jossa se kykenee suorittamaan vaaditun toiminnon tietyissä olosuhteissa ja tietyllä ajanhetkellä tai tietyn ajanjakson aikana, olettaen että vaadittavat ulkoiset resurssit ovat saatavilla.” (Ahonen ym., 2012, s. 11) Käyttövarmuus on yksi merkittävistä tekijöistä joka vaikuttaa järjestelmän luotettavuuteen. Luotettavuudella tarkoitetaan järjestelmän käyttäjän luottamusta siihen, että järjestelmä toimii oletetulla tavalla ilman häiriöitä tai vikoja. (Sommerville ym., 2006, ss. 3–4) Sommerville (2006) kertoo, että neljä olennaista tekijää vaikuttavat järjestelmän käyttövarmuuteen.

- *Saatavuus*: saatavuudella tarkoitetaan järjestelmän kykyä tuottaa palvelua, kun sitä pyydetään.
- *Toimintavarmuus:* toimintavarmuus on järjestelmän kyky tuottaa palvelua kuten sille on määritelty.
- *Turvallisuus:* (Turvallisuudelle on kaksi määritelmää, koska englannin kielen sanoissa *safety* ja *security* tarkoittavat eri asiaa.)
  - *Safety:* turvallisuudella tarkoitetaan järjestelmän kykyä toimia ilman katastrofaalista pettämistä.
  - *Security:* turvallisuudella tarkoitetaan myös järjestelmän kykyä torjua tahallisia tai tahattomia uhkia.

Joskus käyttövarmuus termin alla käytetään myös muita ominaisuuksia, joita ovat esimerkiksi *korjattavuus, ylläpidettävyys, selviytymiskyky* ja *virheen kestävyys.* (Sommerville ym., 2006, ss. 4–5)

Järjestelmiä kehitettäessä on olemassa tapoja jolla järjestelmän käyttövarmuutta voidaan kasvattaa. (Laprie, 1995, s. 2)

- *Vikojen välttäminen:* vältetään vikojen esiintymistä ja esiintyvyyttä.

- *Vikojen sietokyky:* varmistetaan, että järjestelmä toimii odotetusti vaikka vikoja on olemassa.
- *Vikojen poistaminen:* vähennetään vikojen määrää ja vakavuutta.
- *Vikojen ennustaminen:* arvioidaan vikojen määrää ja niiden vakavuutta ja vaikutusta järjestelmälle.

4.2.4 Ympäristön huomiointi käyttöliittymän suunnittelussa

Järjestelmän toimintaympäristöllä on myös suuri vaikutus järjestelmän suunnitteluun. Käyttöliittymäsuunnittelun kannalta merkittäviä ympäristö tekijöitä on viisi kappaletta: fyysinen, turvallisuus, sosiaalinen, organisaatio ja käyttäjän tuki ympäristö (Stone ym., 2005, s. 38).

- *Fyysisellä* ympäristöllä tarkoitetaan muutoksia esimerkiksi valon määrässä, lämpötilassa, äänessä ja ympäristön puhtaudessa. Esimerkiksi kylmyys ja melu saattavat vaikuttaa suunnittelupäätöksiin, jos esimerkiksi käyttäjät pitävät hanskoja kädessä järjestelmää käyttäessä.
- *Turvallisuus* ympäristöllä tarkoitetaan kaikkia vaaroja tai turvallisuusriskejä mitkä ympäröivät järjestelmää. Esimerkiksi järjestelmän käyttäjä saattaa olla pukeutunut suojapukuun, käyttäjän stressi taso saattaa olla korkealla tai ympäristö aiheuttaa muun turvallisuus uhan käyttäjälle.
- *Sosiaalinen* ympäristö tarkoittaa ihmisten välistä vuorovaikutusta. Esimerkkeinä saattaa olla kommunikaatiohierarkia, joka saattaa estää informaation vapaata kulkemista työntekijöiden välillä.
- *Organisaatioympäristöllä* ympäristöllä tarkoitetaan sitä miten järjestelmä integroidaan organisaation nykyisiin käytäntöihin, teknologioihin ja prosesseihin. Organisaatio ympäristöön liittyy olennaisesti esimerkiksi organisaation työkulttuuri.
- *Käyttäjän tuella* tarkoitetaan sitä, että käyttäjää tuetaan dokumentaatiolla, koulutuksella ja vertaistuella riittävästi.

Jokainen näistä ympäristötekijöistä vaikuttaa käyttöliittymä suunnittelussa tehtäviin päätöksiin. Fittsn laki määrittelee miten kauan käyttäjällä kestää painaa tiettyä kohdetta, kun muuttujina on kohteen koko ja kohteen etäisyys käyttäjästä (Budiu, 2019). Harley (2019) kertoo, että kosketus kohteet kosketusnäytöllä tulisi suunnitella jokaista käyttötarkoitusta varten riittävän suuriksi, jotta käyttäjän on helppo koskettaa niitä (Harley, 2019). Esimerkkinä toimii Tesla Model S auton kosketusnäyttö. Käyttöliittymän painikkeet on sijoitettu ohjauspyörästä katsottuna haastavaan paikkaan 17 tuumaisen näytön alareunaan, joka on kaukana käyttäjän käden ajoasennosta ohjauspyörällä (Budiu, 2019). Tämä johtaa siihen, että käyttäjän on haastavaa painaa nappeja, koska kosketus kohteet ovat kauempana, ne ovat pienikokoisia ja ne ovat virtuaalisia painikkeita jotka eivät tarjoa muuta kuin visuaalista palautetta. Tämä vaikuttaa turvallisuuteen, koska silloin kuljettaja joutuu käyttämään ylimääräistä aikaa painikkeiden painamiseen Tämä altistaa autossa olevat ihmiset suuremmalle kolarin riskille. Tässä esimerkissä tulee myös huomioida se, että auto saattaa ajaa kuoppaista tietä jolloin kosketusnäytöllä tiettyyn kohtaan osuminen voi olla entistäkin haastavampaa.

5 YHTEENVETO

Tämän kandidaattitutkielman aiheena oli käyttöliittymä suunnittelu turvallisuuskriittisissä järjestelmissä. Tutkielman tavoitteena oli kerätä yhteen kirjallisuudessa esitettyjä periaatteita ja malleja miten turvallisuuskriittisiä järjestelmiä tulisi suunnitella ja mitä suunnittelutyössä tulisi huomioida. Turvallisuuskriittisten järjestelmien tutkimus on merkittävää, koska järjestelmät ympäröivät meitä kaikkialla arjessa ja niiden varassa on merkittäviä määriä ihmisiä päivittäin. Kirjallisuudessa esiteltyjä periaatteita ja malleja on tutkittu melko pitkälle, mutta niistä ei saa helposti kokonaiskuvaa ja ymmärrystä. Tämä tutkielma pyrki kokoamaan yhteen näitä tutkimuksia ja antaa mahdollisuuden jatkotutkimukselle esimerkiksi kokonaisvaltaisen tarkistuslistan luomiselle. Listaa voitaisiin hyödyntää turvallisuuskriittisen käyttöliittymän suunnitteluprosessissa.

Toinen luku keskittyi tarkastelemaan mitä ovat kriittiset järjestelmät ja mitä erilaisia järjestelmien tyyppejä on olemassa. Tunnistettiin järjestelmien tyypeiksi: turvallisuuskriittiset (Safety & Security Critical), tehtäväkriittiset ja liiketoimintakriittiset järjestelmät (Hinchey & Coyle, 2010). Kriittistä järjestelmistä valittiin tutkielmassa tutkittavaksi turvallisuuskriittiset (safety) järjestelmät. Rajauksen jälkeen määriteltiin perinteiset ja ei-perinteiset turvallisuuskriittiset järjestelmät Knightin (2002) määritelmän mukaan. Rajauksen ja määritelmien jälkeen tutkittiin turvallisuuskriittisten järjestelmien roolia yhteiskunnassa ja turvallisuuskriittisten järjestelmien suhdetta yhteiskunnan kriittiseen infrastruktuuriin.

Kolmas luku käsitteli käyttöliittymiä ja niiden yleisiä suunnitteluperiaatteita. Määriteltiin ihmisen ja teknologian välisen vuorovaikutuksen olennaisimmat termit kuten käyttäjäkokemus, käytettävyys ja käyttöliittymät. Esiteltiin sen jälkeen kirjallisuudessa paljon siteerattuja suunnitteluperiaatteita, joita olivat muun muassa Schneidermanin kultaiset säännöt ja Nielsenin heuristiikat. Näytettiin myös miten sovellusten käyttöliittymät eroavat riippuen siitä mille käyttöjärjestelmälle ne on suunniteltu.

Neljäs luku pyrki vastaamaan tutkielman tutkimuskysymykseen. Esiteltiin mitä haasteita turvallisuuskriittisten järjestelmien suunnittelussa on ja mitä tulee huomioida turvallisuuskriittistä käyttöliittymää suunnitellessa. Tutkielmassa ei kuitenkaan ole täydellistä listaa kaikista huomioitavista asioista, vaan kirjallisuudessa yleisimmin esitettyjä asioita. Tämä tarjoaa mahdollisuuden jatkotutkimukselle, joka voi syventää tässä tutkielmassa käsiteltyjä huomion kohteita. Tässä tutkielmassa esitettyjä huomioitavia asioita olivat: järjestelmien vaatimukset ja sertifikaatit, käyttäjäkeskeiset menetelmät ja käytettävyys, käyttövarmuus ja ympäristön vaikutus.

LÄHTEET

Ahonen, T., Jännes, J., Kunttu, S., Valkokari, P., Venho-Ahonen, O., Välisalo, T., Ellman, A., Hietala, J.-P., Multanen, P., Mäkiranta, A., Saarinen, H., & Franssila, H. (2012). *Käyttövarmuuden hallinta – standardista käytäntöön [Dependability management – from standard to practice]*. 84.

Alben, L. (1996). *Defining the Criteria for Effective Interaction Design*. 5.

Banerjee, I., Nguyen, B., Garousi, V., & Memon, A. (2013). Graphical user interface (GUI) testing: Systematic mapping and repository. *Information and Software Technology*, *55*(10), 1679–1694. https://doi.org/10.1016/j.infsof.2013.03.004

Bozzano, M., & Villafiorita, A. (2010). *Design and Safety Assessment of Critical Systems*. CRC Press.

Budiu, R. (2019, toukokuuta 19). *Tesla's Touchscreen UI: A Case Study of Car-Dashboard User Interface*. Nielsen Norman Group. https://www.nngroup.com/articles/tesla-big-touchscreen/

Elintärkeät toiminnot – Turvallisuuskomitea. (29.10). https://turvallisuuskomitea.fi/yhteiskunnan-turvallisuusstrategia/elintarkeat-toiminnot/

Endsley, M. R., & Kiris, E. O. (1995). The Out-of-the-Loop Performance Problem and Level of Control in Automation. *HUMAN FACTORS*, 15.

Galitz, W. O. (2007). *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. John Wiley & Sons.

Goswami, S., & Gitta, S. (2018, maaliskuuta 27). *Why caring about the user is as important as caring for the patient: The importance of UI/UX in Healthcare Clinical Decision Support Systems.* https://www.insight-rx.com/post/why-caring-about-the-user-is-as-important-as-caring-for-the-patient

Harley, A. (2019, toukokuuta 5). *Touch Targets on Touchscreens*. Nielsen Norman Group. https://www.nngroup.com/articles/touch-target-size/

Hinchey, M., & Coyle, L. (2010). Evolving Critical Systems: A Research Agenda for Computer-Based Systems. *2010 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems*, 430–435. https://doi.org/10.1109/ECBS.2010.56

Hussey, A., Mahemoff, M., & Mahemoff, M. (1999). *Patterns for Designing Safety-Critical Interactive Systems*.

Identity: Critical Systems. (2020, lokakuuta 30). *Identity*. https://www.identity.pt/critical-systems/

International Electrotechnical Commission. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems*. International Electrotechnical Commission.

Knight, J. C. (2002). Safety critical systems: Challenges and directions. *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*, 547–550.

Kraeling, M. (2014, huhtikuuta 12). Practical tips on designing safety-critical software. *Embedded.Com*. https://www.embedded.com/practical-tips-on-designing-safety-critical-software/

Laprie, J. (1995). Dependable computing: Concepts, limits, challenges. *In Proceedings 25th IEEE International Symposium on Fault-Tolerant Computing*, 42–54.

Lehto, M., & Limnéll, J. (2017). Kybersodankäynnin kehityksestä ja tulevaisuudesta. *Tiede ja ase*, *75*.

Mattsson, M. (2013). *Tietoja Suomen kokonaisturvallisuudesta*. 149.

Mazumder, F. K., & Das, U. K. (2014). *Usability guidelines for usable user interface*. 4.

McCarthy, J., & Wright, P. (2005). Putting 'felt-life' at the centre of human–computer interaction (HCI). *Cognition, Technology & Work*, *7*(4), 262–271. https://doi.org/10.1007/s10111-005-0011-y

Murphy, N. (1998, elokuuta 1). *How to Design Safer Systems via Better User Interfaces*. Barr Group. https://barrgroup.com/embedded-systems/how-to/product-safety-vs-usability

Nielsen, J. (1994a, tammikuuta 11). *Heuristic Evaluation: How-To: Article by Jakob Nielsen*. Nielsen Norman Group. https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

Nielsen, J. (1994b, huhtikuuta 24). *10 Heuristics for User Interface Design: Article by Jakob Nielsen*. Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/

Nielsen, J. (2005, lokakuuta 4). *Medical Usability: How to Kill Patients Through Bad Design*. Nielsen Norman Group. https://www.nngroup.com/articles/medical-usability/

Nielsen, J. (2012, maaliskuuta 1). *Usability 101: Introduction to Usability*. Nielsen Norman Group. https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Oedewald, P., & Reiman, T. (2006). *Turvallisuuskriittisten organisaatioiden toiminnan erityispiirteet*. VTT.

Pietrantuono, R., & Russo, S. (2013). Introduction to Safety Critical Systems. Teoksessa D. Cotroneo (Toim.), *Innovative Technologies for Dependable OTS-Based Critical Systems: Challenges and Achievements of the CRITICAL STEP Project* (ss. 17–27). Springer Milan. https://doi.org/10.1007/978-88-470-2772-5_2

Podoler, Y. (2020, huhtikuuta 22). UI Testing: A Comprehensive Guide. *TestCraft*. https://www.testcraft.io/ui-testing/

Rimpiläinen, T. (2020, lokakuuta 22). *Psykoterapiakeskus Vastaamon kiristäjä julkaisi yöllä lisää erittäin arkaluontoisia potilaskertomuksia*. Yle Uutiset. https://yle.fi/uutiset/3-11606925

Salahdine, F., & Kaabouch, N. (2019). Social Engineering Attacks: A Survey. *Future Internet*, *11*. https://doi.org/10.3390/fi11040089

Sommerville, I., Dewsbury, G., Clarke, K., & Rouncefield, M. (2006). *Dependability and Trust in Organisational and Domestic Computer Systems", in Trust in Technology: A Socio-technical Perspective*.

Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). *User Interface Design and Evaluation*. Elsevier.

Tanhuamäki, H. (2006). *Kriittisten tietojärjestelmien muutoksen hallinta*. 77.

Valtioneuvostonkanslia. (2016, helmikuuta 25). *Kyberosaaminen Suomessa – Nykytila ja tiekartta tulevaisuuteen*. Selvitys- ja tutkimustoiminta. https://tietokayttoon.fi/julkaisu?pubid=9301

Vuori, M. (2011). *Agile Development of Safety-Critical Software*. 114.

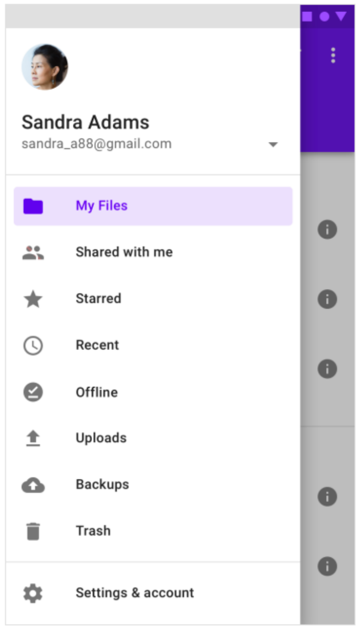
Zink, C. (2017, maaliskuuta 10). *5 Benefits of Great Enterprise UI/UX Design*. https://www.enviance.com/blog/benefits-of-great-enterprise-ui/ux-design

LIITE 1 VALIKKOJEN ESITYS TYYLI IOS JA ANDROID JÄRJESTELMISSÄ

iOS

Lähde: https://developer.apple.com/design/human-interface-guidelines/ios/bars/tab-bars/

Android



A modal drawer on mobile

Lähde: https://material.io/components/navigation-drawer#modal-drawer