

**Mobiilisovellusten käyttäjädokumentaation ominaispiirteet ja  
rooli käyttäjäkokemuksen luomisessa**

Laura Rautava  
Tampereen yliopisto  
Viestintätieteiden tiedekunta  
Monikielisen viestinnän ja käännöstieteen maisteriopinnot  
Englannin kääntämisen ja tulkkauksen opintosuunta  
Pro gradu -tutkielma  
Toukokuu 2018



Tampereen yliopisto  
Monikielisen viestinnän ja käännöstieteen maisteriohjelma  
Englannin kääntämisen ja tulkkauksen opintosuunta  
Viestintätieteiden tiedekunta

RAUTAVA, LAURA: Mobiilisovellusten käyttäjädokumentaation ominaispiirteet ja rooli käyttäjäkokemuksen luomisessa

Pro gradu -tutkielma, 80 sivua, englanninkielinen lyhennelmä 14 sivua  
Toukokuu 2018

---

Tämän pro gradu -tutkielman tavoitteena on selvittää, millaista mobiilisovellusten käyttäjädokumentaatio on ja miten sillä voidaan vaikuttaa käyttäjäkokemukseen. Tutkimushypoteesi on, että mobiilisovellusten käyttäjädokumentaatio on varsin vaihtelevaa ja että sillä voidaan jossain määrin paikata puutteita sovellusten käytettävyydessä. Tavoitteena on myös luoda käytettävyysheuristiikat mobiilisovellusten käyttäjädokumentaation arvioimiseen.

Tutkimuksen teoriatausta muodostuu yleisestä käyttäjäkokemus- ja käytettävyystudkimuksesta sekä erityisesti mobiilisovellusten käyttäjäkokemusta ja käytettävyyttä käsittelevästä kirjallisuudesta. Teknisen viestinnän näkökulmaa haetaan minimalismista, erityisesti Hans van der Meij'n ja John M. Carrollin (1995) minimalismin periaatteita ja niihin liittyviä heuristiikkoja käsittelevästä artikkelista.

Tutkimusmenetelmänä on heuristinen asiantuntija-arviointi, jota varten luodaan omat heuristiikat, sillä aiempia mobiilisovellusten käyttäjädokumentaation arvioimiseen tarkoitettuja heuristiikkoja ei ole. Heuristiikat perustuvat van der Meij'n ja Carrollin (mt.) minimalismiheuristiikkoihin, Vesa Purhon (2000) koostamiin teknisen dokumentaation käytettävyysheuristiikkoihin ja älypuhelinien käyttöjärjestelmäkohtaisiin dokumentaatiota koskeviin ohjeistuksiin. Heuristiikoista kehitetään aineiston analyysin perusteella toinen versio. Tutkimusaineisto koostuu kolmen mobiilisovelluksen käyttäjädokumentaatiosta: sovellukset ovat Pocket, Evernote ja Trello, ja aineiston laajuus on noin 140 moduulia. Yksi moduuli vastaa esimerkiksi yhtä tehtäväsarjaa tai vinkkiä. Sovellukset ovat hyötysovelluksia, joilla voi esimerkiksi tallentaa erilaisia sisältöjä verkosta, tehdä muistiinpanoja ja hallita projekteja.

Tutkimuksen perusteella mobiilisovellusten käyttäjädokumentaatiota on mahdollista toteuttaa varsin eri tavoin ja myös dokumentaation ongelmat ja ansiot ovat moninaisia. Keskeisimmät ongelmat liittyvät sovelluksen ja dokumentaation samanaikaiseen käyttöön. Käyttäjäkokemukseen dokumentaatiolla voidaan vaikuttaa eniten kognitiivisten rasitteiden minimoimisen osalta tarjoamalla merkityksellistä käytönaikaista dokumentaatiota ja hyödyntämällä esimerkiksi kuvia ymmärtämisen ja muistamisen tukena. Mahdollisia jatkotutkimuksen aiheita on esimerkiksi mobiilipelien käyttäjädokumentaation rooli pelaajakokemuksen luomisessa. Myös laajemman ja monipuolisemman sovellusjoukon tarkastelu tässä tutkielmassa luotuja heuristiikkoja käyttäen olisi paikallaan.

Avainsanat: mobiilisovellus, käyttäjädokumentaatio, käyttäjäkokemus, käytettävyys, tekninen viestintä, minimalismi



# Sisällys

1	Johdanto .....	1
2	Mobiilisovellusten käyttäjäkokemus ja käytettävyys.....	4
2.1	Käyttäjäkokemus.....	5
2.2	Käyttäjäkokemus mobiilisovelluksissa .....	8
2.3	Mobiilisovellusten käytettävyyspiirteet .....	12
3	Minimalismi teknisessä viestinnässä.....	17
3.1	Minimalismin periaatteet .....	17
3.2	Minimalismin haasteita .....	19
3.3	Van der Meij'n ja Carrollin minimalismiheuristiikat.....	21
3.4	Minimalismi mobiilisovellusten käyttäjädokumentaatioissa .....	26
4	Heuristinen asiantuntija-arviointi .....	30
4.1	Heuristinen asiantuntija-arviointi menetelmänä .....	30
4.2	Purhon käytettävyysheuristiikat tekniselle dokumentaatiolle.....	32
4.3	Mobiilisovellusten käyttöjärjestelmäkohtaiset ohjeistukset.....	35
4.4	Heuristiikat mobiilisovellusten käyttäjädokumentaation arvioimiseen .....	37
5	Aineisto .....	44
6	Mobiilisovellusten käyttäjädokumentaation heuristinen asiantuntija-arviointi .....	46
6.1	Pocket.....	46
6.1.1	Käyttäjädokumentaation aiheet ja asiasisältö .....	47
6.1.2	Käyttäjädokumentaation informaatorakenne.....	51
6.1.3	Käyttäjädokumentaation kieli.....	55
6.2	Evernote .....	56
6.2.1	Käytönaikainen dokumentaatio .....	58
6.2.2	Informaatorakenne.....	61
6.2.3	Heuristiikkojen soveltuvuus .....	63
6.3	Trello.....	64
6.3.1	Käyttäjädokumentaation toteutus .....	65
6.3.2	Informaatorakenne ja kieli.....	68
6.4	Yhteenveto .....	71
7	Päätelmät .....	77
	Lähteet.....	81
	Aineisto .....	81
	Kirjallisuus.....	81

English Summary



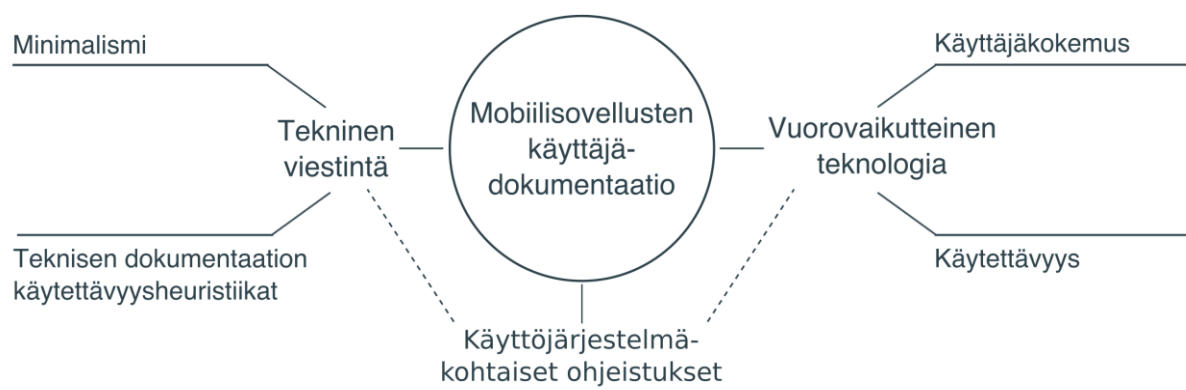
# 1 Johdanto

Älypuhelinien käyttö on suurelle osalle ihmisistä arkipäivää ja Suomessakin älypuhelinien yleistymisen on viime vuosina ollut nopeaa: Tilastokeskuksen (2017) mukaan vuonna 2017 alle 55-vuotiaista suomalaisista älypuhelinia käytti jo 94 prosenttia. Älypuhelinien ja mobiilisovellusten määrä onkin kasvanut jatkuvasti siitä lähtien, kun Apple ja Google avasivat sovelluskauppansa vuonna 2008 (Wikipedia). Vuonna 2018 älypuhelinien käyttäjien määrän maailmanlaajuisesti ennustetaan nousevan 2,53 miljardiin (Statista 2018) – tällöin älypuhelimia käyttäisi 36 prosenttia maailman väestöstä. Mobiilisovelluksia oli vuoden 2017 kesäkuussa Google Play -sovelluskaupassa 3 miljoonaa kappaletta ja saman vuoden tammikuussa Applen App Storessa 2,2 miljoonaa kappaletta: mobiilisovellusten latauksia oli vuonna 2017 yhteensä arviolta 197 miljardia kappaletta (Dogtiev 2018). Kyseessä on valtava liiketoiminnan ala.

Kuten tietokonesovellukset, myös mobiilisovellukset tarvitsevat käyttäjädokumentaatiota tukemaan käyttäjiä sovelluksen käytössä, sen oppimisessa ja ongelmatilanteissa. Vaikka yhä useampi käyttää älypuhelimia ja niitä varten kehitettyjä mobiilisovelluksia ja sitä myöten myös tarve niiden käyttäjädokumentaatiolle lisääntyy, kyseessä on kuitenkin toistaiseksi hyvin vähän tutkittu teknisen viestinnän ala. Tämän pro gradu -tutkielman aiheena ovatkin mobiilisovellusten käyttäjädokumentaation ominaispiirteet. Aihetta on aiemmin käsitelty lähinnä blogiteksteissä (ks. esim. Babich 2017) ja sivuttu mobiilisovellusten käytettävyyshauristiikoissa (ks. esim. Yáñez Gómez, Cascado Caballero & Sevillano 2014) ja mobiililaitteiden käyttöjärjestelmäkohtaisissa ohjeissa (Android Developers 2017; Apple Developers 2017), mutta varsinaista tutkimusta aiheesta en ole löytänyt.

Tutkimustehtäväni on selvittää, millaista mobiilisovellusten käyttäjädokumentaatio on: mitä ovat sen ominaispiirteet ja millainen rooli käyttäjädokumentaatiolla on käyttäjäkokemuksen luomisessa. Lähestyn ongelmaa yhdistämällä vuorovaikutteisen teknologian alalta käyttäjäkokemus- ja käytettävyystudkimusta sekä teknisen viestinnän kirjallisuutta ja luon näiden pohjalta arviointityökalukseni mobiilisovellusten käyttäjädokumentaation käytettävyyshauristiikat. Tutkimukseni tavoite on tarjota tuoretta tietoa siitä, millaista mobiilisovellusten käyttäjädokumentaatio on ja miten sillä voidaan tukea positiivisen käyttäjäkokemuksen muodostumista. Tämä tieto voisi auttaa luomaan käyttäjädokumentaatiota, joka parantaa käyttäjäkokemusta, minkä lisäksi tämä tutkimus täydentää teknisen viestinnän tutkimuskenttää.

Mobiilisovellusten käyttäjädokumentaatiolle sopivia heuristiikkoja ei ole aiemmin luotu, joten pyrin tässä tutkielmassa keräämään heuristiikoilleni mahdollisimman laajan teoriapohjan: tarkastelen mobiilisovellusten käyttäjäkokemukseen ja käytettävyyteen liittyvää kirjallisuutta ja teknisen viestinnän puolelta haen pohjaa minimalismista. Hypoteesini on, että minimalismin periaatteet käyttäjäkeskeisyydessään sopivat hyvin käyttäjäkokemuksen ja käytettävyyden rinnalle mobiilisovellusten käyttäjädokumentaation ominaispiirteiden tarkasteluun. Näiden lisäksi käytän heuristiikkojeni pohjana Vesa Purhon (2000) teknisen viestinnän käytettävyysheuristiikkoja ja mobiilisovellusten käyttöjärjestelmäkohtaisia ohjeistuksia. Teoriataustani keskinäiset suhteet näkyvät kuvassa 1:



*Kuva 1. Tutkielman teoriatausta*

Hans van der Meij ja John M. Carroll (1995, 245) määrittelevät minimalismin teknisen viestinnän lähestymistavaksi, jonka keskeisiä periaatteita ovat toimintakeskeisyys, dokumentaation merkityksellisyys käyttäjälle, virheiden välttäminen ja niistä palautumisen tukeminen sekä erilaisten käyttäjien ja käyttötarkoitusten tukeminen. Minimalistisessa dokumentaatiossa käyttäjälle ei selitetä jokaista yksityiskohtaa kädestä pitäen, vaan tälle annetaan tilaisuus myös itse oivaltaa asioita ja käyttää aiempia tietojaan ja osaamistaan. Pienempi tekstimäärä ja selkeämpi ilmaisu myös helpottavat ymmärtämistä ja navigointia (Guren 2014).

Minimalismia, käyttäjäkokemusta ja käytettävyyttä yhdistää juuri käyttäjän keskeinen asema. Don Normanin ja Jakob Nielsenin (s. d.) mukaan käyttäjäkokemus kattaa käyttäjän vuorovaikutuksen tuotteen tai palvelun kanssa ja ne elämykset, jotka käyttäjä tässä vuorovaikutuksessa saa. Käytettävyys puolestaan on järjestelmän laatupiirre: käytettävä tuote on esimerkiksi helposti opittava ja miellyttävä käyttää. Käyttäjäkokemus on siis käytettävyyttä laajempi käsite. Omalle tutkimukselleni keskeisillä mobiilisovellusten käyttäjäkokemuksella ja



käytettävyydellä on omia erityispiirteitä, jotka johtuvat niiden käyttökonteksteista ja mobiililaitteiden erityisistä ominaisuuksista ja toiminnallisuuksista, ja tämän vuoksi pidän tärkeänä ottaa käyttäjäkokemuksen ja käytettävyyden huomioon tässä tutkielmassa.

Tutkimusmenetelmäni on siis heuristinen asiantuntija-arviointi. Tutkimusaineistokseni olen valinnut kolme mobiilisovellusta, joiden käyttäjädokumentaatiota tarkastelen: nämä ovat Pocket, Evernote ja Trello. Kaikki kolme ovat eräänlaisia hyötysovelluksia, joilla käyttäjä voi esimerkiksi tallentaa erilaisia sisältöjä verkosta, luoda muistiinpanoja ja hallita projekteja. Käytän sovellusten Android-versioita, sillä ne ovat minulle helpommin saatavilla, ja iOS- ja Android-versioiden vertaileminen tekisi aineistosta liian laajan tämän tutkielman tarpeisiin.

Käsittelen erityisesti älypuhelinsovellusten käyttäjädokumentaatiota, joten rajaan muut mobiililaitteet tarkasteluni ulkopuolelle. Älypuhelimella tarkoitan tämän tutkielman kontekstissa matkapuhelinta, jossa on kosketusnäyttö eikä erillistä näppäimistöä. Mobiililaitteita on useita erilaisia: niitä ovat matkapuhelinten lisäksi esimerkiksi tabletit, e-kirjanlukulaitteet ja erilaiset asusteenomaiset älylaitteet kuten älykellot. Vaikka mobiililaitteilla on keskenään joitakin samoja piirteitä, niiden käyttö kuitenkin eroaa jossain määrin, ja siksi myös niiden käyttäjäkokemussuunnittelu on hieman erilaista (Budi 2015). Esimerkiksi e-kirjanlukulaitteiden käyttötarkoitus on hyvin tarkkaan rajattu, toisin kuin älypuhelinien ja tablettien, ja tableteissa taas on suuremman näytön vuoksi vähemmän näyttötilaan liittyviä haasteita kuin älypuhelimissa. Viittaan tässä tutkielmassa mobiilisovelluksella juuri älypuhelinsovelluksiin.

Tämän tutkielman luvussa 2 käsittelen käyttäjäkokemuksen, käyttäjäkokemussuunnittelun ja käytettävyyden tutkimusta sekä yleisellä tasolla että erityisesti mobiilisovellusten kannalta. Luvussa 3 tuon mukaan teknisen viestinnän näkökulmaa käsittelemällä minimalismia, minkä jälkeen luvussa 4 esittelen tutkimusmenetelmäni eli heuristisen asiantuntija-arvioinnin, Purhon (2000) teknisen dokumentaation käytettävyydsheuristiikat ja mobiilisovellusten käyttöjärjestelmäkohtaisia ohjeistuksia sekä lopuksi näiden pohjalta laatimani mobiilisovellusten käyttäjädokumentaation käytettävyydsheuristiikat. Luvussa 5 esittelen aineistoni ja sen valintaperusteet. Luvussa 6 käyn läpi osa-aineistojeni analyysin sovellus sovellukselta vetäen löydökseni lopuksi yhteen ja luvussa 7 esitän johtopäätökset ja mahdolliset jatkotutkimuksen aiheet.

## 2 Mobiilisovellusten käyttäjäkokemus ja käytettävyys

Tässä luvussa käsittelen käyttäjäkokemusta, käyttäjäkokemussuunnittelua ja käytettävyyttä sekä sitä, mitä nämä merkitsevät erityisesti mobiilisovellusten kontekstissa. Tämä luo pohjaa yhdelle tutkimuskysymyksistäni eli sille, miten mobiilisovellusten käyttäjädokumentaatio vaikuttaa käyttäjäkokemukseen. Aloitan määrittelemällä työni kannalta keskeiset käsitteet, minkä jälkeen käsittelen luvussa 2.1 käyttäjäkokemusta ja käyttäjäkokemussuunnittelua yleisesti, luvussa 2.2 käyttäjäkokemussuunnittelua erityisesti mobiilisovellusten näkökulmasta ja lopuksi luvussa 2.3 mobiilisovellusten käytettävyyspiirteitä.

Suomen Standardisoimisliitto SFS (ISO 9241-210 2012, 16) määrittelee *käyttäjäkokemuksen* (*user experience, UX*) seuraavasti: ”henkilön havainnointi ja vasteet, jotka ovat seurausta tuotteen, järjestelmän tai palvelun käytöstä ja/tai ennakoitusta käytöstä”. *Käyttäjäkokemussuunnittelu* (*user experience design, UX design*) taas on prosessi, jolla varmistetaan, että jokainen käyttäjän ja tuotteen välinen vuorovaikutus on tarkoituksenmukainen: kaikki potentiaaliset vuorovaikutustilanteet ja käyttäjän toiminta on otettu huomioon tuotteen suunnittelussa ja käyttäjän tarpeet on ymmärretty ja ennakoitu (Garrett 2002). Michael Cummings (2010) mainitsee käyttäjäkokemussuunnittelun piirteiksi myös ”käyttäjakeskeiset suunnittelukäytännöt ja kontekstisidonnaisuuden sekä tiettyjen metodien ja tekniikoiden käytön johdonmukaisten, ennakoitavien ja haluttujen vaikutusten aikaansaamiseksi kohdekäyttäjässä” siten, että sekä käyttäjät että tuoteorganisaatio saavuttavat tavoitteensa. Käyttäjäkokemussuunnittelu yhdistää useita erikoisaloja, kuten teknologiaa, markkinointia, graafista ja teollisuussuunnittelua sekä käyttöliittymäsuunnittelua (Norman & Nielsen s. d.).

*Käytettävyydellä* (*usability*) on useita määritelmiä. Suomen Standardisoimisliitto SFS (ISO 9241-210 2012, 16) määrittelee käytettävyyden näin: ”mitta, miten hyvin määrätyt käyttäjät voivat käyttää järjestelmää, tuotetta tai palvelua tietyssä käyttötilanteessa saavuttaakseen määritetyt tavoitteet tuloksellisesti, tehokkaasti ja tyytyväisinä”. Yksi yleisimmin käytetyistä käytettävyyden määritelmistä on Nielsenin (1993, 26) malli, jossa käytettävyys jaetaan viiteen osatekijään: opittavuus eli käytön helppous ensimmäisellä käyttökerralla, tehokkuus eli opitun järjestelmän käytön nopeus, muistettavuus tauon jälkeen, virheiden vähäisyys ja korjattavuus sekä käytön miellyttävyys (osatekijöiden suomennokset Ovaska ym. 2005, 3). Käytännössä käytettävyydeltään hyvä järjestelmä on Nielsenin (mt.) mukaan helppo ja miellyttävä käyttää. David Benyon (2014, 76) huomauttaa, että 2000-luvulla määritelmä on laajentunut käytön ja

oppimisen helppoudesta ja joustavuudesta kattamaan monia muita piirteitä, ja esimerkiksi saavutettavuudesta on tullut merkittävä käytettävyyden osa-alue.

## 2.1 Käyttäjäkokemus

Tässä luvussa tarkastelen Jesse James Garrettin (2002) ja Michael Cummingsin (2010) käyttäjäkokemussuunnittelun malleja: Garrettin viisitasomallia (*five planes model*), joka käsittelee käyttäjäkokemussuunnittelua erityisesti prosessin näkökulmasta, ja Cummingsin mallia, joka pohjautuu viisitasomalliin ja esittelee käyttäjäkokemuksen osia käyttäjän näkökulmasta. Pohdin myös, miten nämä mallit soveltuvat mobiilisovellusten käyttäjäkokemuksen suunnitteluun ja tarkasteluun.

Garrett (2002) lähestyy käyttäjäkokemusta ja sen suunnittelua kehittämänsä viisitasomallin kautta. Garrett jakaa käyttäjäkokemussuunnittelun viiteen hierarkkiseen tasoon, jotka puolestaan koostuvat käyttäjäkokemuksen elementeistä. Garrett lähestyy käyttäjäkokemusta erityisesti verkkosivusuunnittelun näkökulmasta, mutta malli on yleistettävissä muihinkin tuotteisiin ja se sopii hyvin sovellusten tarkasteluun, sillä monien sovellusten käyttäjäkokemus rakentuu samankaltaisista tasoista.

Hierarkiassa ylimpänä sijaitseva pintataso (*surface plane*) käsittää tuotteen ulkonäön: esimerkiksi kuvat ja tekstin sekä käytetyt värit. Runkotasoa (*skeleton plane*) muodostuu interaktiivisten elementtien ja esimerkiksi kuvien sijoittelusta, jolla pyritään optimoimaan käytön tehokkuus ja autetaan käyttäjää oppimaan ja muistamaan, miten tuotetta käytetään. Rakennetaso (*structure plane*) on runkotasoa abstraktimpi ja käsittää esimerkiksi navigaatiopolut ja mahdolliset kategoriat. Rakennetasolla tuotteen elementit sovitetaan yhteen kokonaisuudeksi. Laajuustaso (*scope plane*) määrittää sen, mitä toiminnallisuuksia ja ominaisuuksia tuote sisältää, kun taas strategiataso (*strategy plane*) sisältää tuotteen kehittäjien ja käyttäjien tavoitteet, jotka tuotteella on tarkoitus saavuttaa. (Garrett 2002, 22–23.)

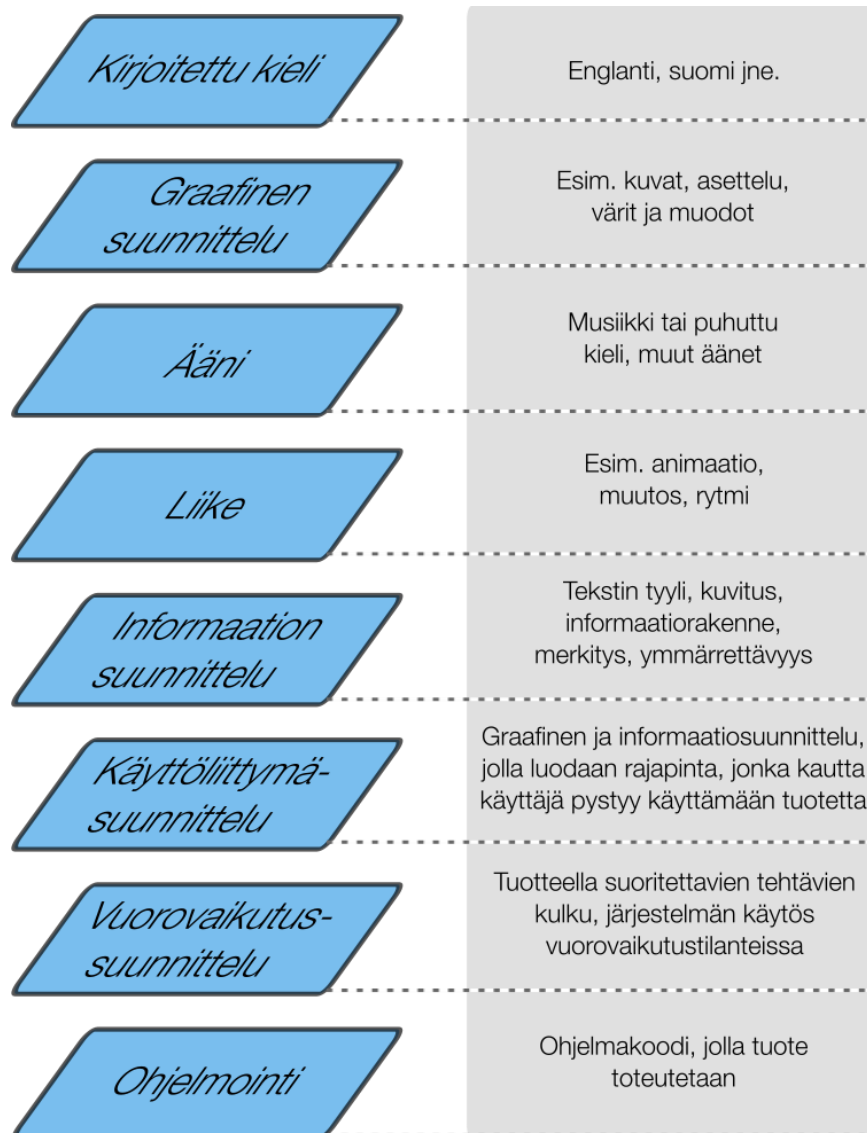
Tuote rakentuu alhaalta ylöspäin ja alempi taso määrittää ylempiään: esimerkiksi tuotteelle asetetut tavoitteet (strategia) määrittävät sitä, millaisia ominaisuuksia (laajuus) tuotteella on oltava. Tasojen väliset riippuvuudet kuitenkin toimivat molempiin suuntiin. Jos tasot eivät sovi yhteen, käyttäjäkokemus kärsii. Myös kehitysprosessille voi aiheutua lisäkustannuksia ja viivästyksiä. Garrett (mt.) toteaa, että olisi ihanteellista käsitellä useampia tasoja yhtä aikaa, mutta siten, että mikään taso ei koskaan valmistu ennen sitä alempia tasoja. (mts. 25–27.)

Garrett (mt.) keskittyy mallissaan verkkosivuihin ja käsittelee myös niiden kahta puolta: verkkosivuja ohjelmistokäyttöliittymänä (*software interface*) ja hypertekstuaalisina informaatiotiloina (*hypertext information space*). Ohjelmistonäkökulmasta verkkosivu on työkalu, jota käytetään tiettyjen tavoitteiden saavuttamiseen. Kyseessä on siis hyvin tehtäväkeskeinen näkökulma. Hypertekstinäkökulmasta katsottuna tärkeintä taas on informaatio. Myös sovelluksia voidaan tarkastella näistä kahdesta näkökulmasta, työkaluina ja informaation välittäjinä – se, kumpi näkökulma on etusijalla, riippuu sovelluksesta. (mts. 27–31.) Näkisin, että useimmat mobiilisovellukset kallistuvat enemmän työkaluiksi kuin informaation välittäjiksi: esimerkiksi verkkopankkisovellukset ovat pankkiasioden hoitamisen väline, vaikka käyttäjä toki saa niistä myös tietoa esimerkiksi omasta rahatilanteestaan ja pystyy vastaanottamaan pankin viestejä ja tiedotteita. Sen sijaan vaikkapa uutisovellukset ovat enemmän informaation välittäjiä kuin tiettyyn päämäärään tähtääviä työkaluja. Suurin osa sovelluksista kuitenkin sisältäne molempia ominaisuuksia.

Garrett (mts. 30–31) esittelee jokaiseen käyttäjäkokemussuunnittelun tasoon liittyvät elementit, joista osan hän yhdistää ohjelmistonäkökulmaan ja osan informaationäkökulmaan. Esimerkiksi rakennetaso muodostuu Garrettin (mt.) mukaan vuorovaikutussuunnittelusta (ohjelmistonäkökulma) ja informaatioarkkitehtuurista (informaationäkökulma). Osa elementeistä ei ole selkeästi liitettävissä vain yhteen näkökulmaan, vaan esimerkiksi pintatasolla visuaalinen suunnittelu liittyy molempiin. Saman tason elementit muodostavat kunkin tason yhdessä, mutta niiden osuus vaihtelee tuotekohtaisesti. Todellisuudessa elementit eivät ole selvärajaisia, ja voi olla vaikea tunnistaa, mitä elementtejä muuttamalla käyttäjäkokemukseen liittyviä ongelmia voitaisiin ratkaista (mts. 35). Tässä voisikin olla paikka, jossa käyttäjädokumentaatio tukee käyttäjäkokemusta, jos ongelmaa ei saada ratkaistua tuotetta itseään muuttamalla.

Käyttäjäkokemussuunnittelun eri tasoilla vaikuttavien elementtien lisäksi Garrett (mt.) mainitsee kaksi muuta käyttäjäkokemukseen vaikuttavaa tekijää: sisällön ja teknologian. Sisältö on Garrettin mukaan kaikista tekijöistä tärkein, sillä juuri se tekee tuotteesta käyttäjälle arvokkaan: käyttäjä ei käytä tuotetta ihastellakseen toimivaa navigaatio suunnittelua. Toinen tekijä on teknologia, joka vaikuttaa niihin mahdollisuuksiin, joita käyttäjäkokemussuunnittelijoilla on käytettävissään. (mts. 35–36.) Nykyteknologia mahdollistaa hyvin hienostuneita ratkaisuja: esimerkiksi mobiililaitteet tarjoavat uusia mahdollisuuksia, mutta toisaalta näytön koko ja toiminnallisuudet sekä käyttöjärjestelmäkohtaiset ominaisuudet asettavat myös omia rajoitteitaan. Näitäkin rajoitteita käyttäjäkokemuksen luomisessa voidaan kuitenkin jossain määrin paikata dokumentaatiolla: tähän palaan luvussa 7, kun esittelen päätelmäni.

Cummings (2010) huomauttaa, että käyttäjä ei havaitse jokaista käyttäjäkokemuksen osaa erikseen, vaan kokemus syntyy niiden kokonaisuudesta. Cummings ottaa käyttäjäkeskeisen näkökulman Garretin viisitasomalliin, joka itsessään keskittyy käyttäjäkokemussuunnittelun prosessiin. Hän esittelee ne elementit, jotka muodostavat käyttäjän kokemuksen tuotteesta. Kuten Garrett, myös Cummings asettaa elementit hierarkkiseen järjestykseen, mutta Cummingsin hierarkiassa ylhäällä ovat ne tekijät, joista käyttäjä on eniten tietoinen, ja alhaalla ne, joita käyttäjä ei välttämättä ollenkaan erikseen tiedosta (kuva 2).



Kuva 2. Cummingsin (2010) käyttäjäkokemusmalli (suomennokset omiani)

Käyttäjän näkökulmasta havaittavimpia käyttäjäkokemuksen luoja ovat siis muun muassa kieli, visuaaliset elementit ja ääni, kun taas esimerkiksi tuotteen teknistä toteutusta käyttäjä tuskin tulee edes ajatelleeksi. Näiden välille sijoittuvat muun muassa informaatorakenne ja

informaation esitystavat sekä käyttöliittymäsuunnittelu. Cummings (mt.) vertaa käyttäjäkokemusta kieleen: käyttäjäkokemussuunnittelu on kuin synteettinen yhdistelmäkieli, joka muodostuu erilaisista sanastoista, jotka kuulija ymmärtää samanaikaisesti ja jotka kietoutuvat yhdeksi kokonaisuudeksi. Käyttäjäkokemussuunnittelun osat muodostavat kielen, jota voidaan käyttää haluttujen vaikutusten saavuttamiseksi.

Cummings (mt.) painottaa myös käyttäjäkokemussuunnittelun kontekstisidonnaisuutta. Jokaisen tuotteen konteksti on erilainen, joten myös jokaisen tuotteen käyttäjäkokemusta on lähestyttävä omasta kontekstistaan. Siinä, missä esimerkiksi käytettävyyssuunnittelu keskittyy tuotteen toiminnallisuuteen, käyttäjäkokemussuunnittelu tunnistaa myös tunne-elämykset, jotka voivat olla arvokkaita itsessään. Käyttäjäkokemussuunnittelussa on hyvä ottaa huomioon esimerkiksi kulttuurinen konteksti, joka vaikuttaa siihen, miten koemme vuorovaikutteisen teknologian.

Cummings (mt.) tiivistää käyttäjäkokemussuunnittelun seuraavasti: siinä yhdistetään kaikki vuorovaikutteisen järjestelmän elementit siten, että käyttäjän tarpeet, tavoitteet, halut ja odotukset täytetään ja rajoitteet otetaan huomioon, minkä seurauksena myös tuoteorganisaation tavoitteet täyttyvät, ja käyttäjäkokemuksen kokonaisuus on enemmän kuin osiensa summa. Hyvä käyttäjädokumentaatio edesauttaa näiden tavoitteiden täyttymistä, ja monet Cummingsin (mt.) esittelemistä käyttäjäkokemuksen elementeistä, kuten kirjoitettu kieli, graafinen suunnittelu ja informaation suunnittelu, ovat myös dokumentaation osatekijöitä.

## **2.2 Käyttäjäkokemus mobiilisovelluksissa**

Mobiililaitteita koskevat monet samoista käyttäjäkokemussuunnittelun periaatteista kuin perinteisempiäkin järjestelmiä ja tuotteita (Benyon 2014, 435). Mobiilisovellusten käyttäjäkokemussuunnittelulla on kuitenkin myös omia erityispiirteitä, sillä sekä mobiilisovellukset itsessään että niiden käyttö ovat erilaisia kuin perinteisten työpöytäsovellusten. Siinä, missä työpöytäsovellusten kohdekäyttäjryhmä on usein ennustettavissa ja rajattavissa, mobiilisovellusten kohdalla tämä voi olla vaikeampaa. Käyttäjät ja käyttökontekstit vaihtelevat suuresti. (mts. 436.) Myös Cummingsin (2010) esittelemien käyttäjäkokemuksen osatekijöiden suhteet saattavat mobiilisovelluksissa erota verkkosivujen ja työpöytäsovellusten käyttäjäkokemukseen nähden: tämä ei kuitenkaan vaikuta Cummingsin elementtien hierarkiaan, sillä käyttäjä ei välttämättä ole esimerkiksi vuorovaikutussuunnittelusta sen tietoisempi kuin työpöytäsovellustenkaan kohdalla. Benyonin (2014, 439) mukaan mobiililaitteiden käytön

tarkan seuraamisen vaikeus hankaloittaa käyttäjäkokemuksen tutkimista ja siten suunnittelua. Laajempia käyttökontekstiin liittyviä asioita ja käyttäytymistä on kuitenkin mahdollista seurata. Seuraavaksi esittelen joitakin mobiilisovellusten käyttäjäkokemussuunnittelun keskeisiä piirteitä, koska ne luovat kontekstia mobiilisovellusten käyttäjädokumentaation arvioinnille.

Kati Kuusinen ja Tommi Mikkonen (2014, 221) mainitsevat tärkeimmiksi mobiilisovellusten käyttöliittymäsuunnittelun tavoitteiksi sen, että käyttäjä saa tehtyä tärkeimmät tehtävät nopeasti ja tuntee hallitsevansa sovellusta. Työpöytäsovelluksiin verrattuna mobiilisovellusten käytön täytyykin olla nopeampaa ja toimintojen yksinkertaisempia. Mobiilisovellusten käyttö on luonteeltaan erilaista kuin työpöytäsovellusten: niitä käytetään usein liikkeellä ollessa, ja käyttösessiot ovat tyypillisesti lyhyempiä. Usein mobiilisovellusten käyttäjät myös tekevät muita asioita samalla, joten huomio on sirpaleisempaa. Tehtävät tulisi saada suoritettua vähimmäismäärällä painalluksia ja näppäilyjä. Mielestäni on kuitenkin huomioitava, ettei mobiililaitteiden käyttö aina ole liikkuvaa ja hetkittäistä, sillä voihan esimerkiksi älypuhelin käyttää myös kotisohvalla.

Mobiililaitteilla on omat vahvuutensa ja heikkoutensa. Vaikka älypuhelimien näyttöjen koko onkin kasvanut jatkuvasti, ovat ne silti pieniä tietokoneiden näyttöihin verrattuna. Puhelimen näytölle mahtuu huomattavasti vähemmän sisältöä kuin tietokoneen näytölle: tietokoneella yhdelle ruudulliselle mahtuvan sisällön näyttämiseen puhelimella vaaditaan useampi ruudunmitta. Tämä taas vaatii käyttäjältä työmuistin käyttöä, kun kaikki informaatio ei ole samanaikaisesti nähtävissä ruudulta. (Budiun 2015.) Kognitiivisen rasituksen määrä kuitenkin riippuu luettavasta tekstistä. Kate Meyerin (2016) mukaan vaikeaselkoisia tekstejä, kuten tieteellisiä artikkeleita, on vaikeampi sisäistää mobiililaitteen näytöltä luettuna, mutta esimerkiksi viihteelliset sisällöt sopivat hyvin pieneltä näytöltä luettaviksi.

Raluca Budiun (2015) mukaan rajallinen näyttötila aiheuttaa myös sen, että sovelluskehittäjiä on mietittävä tarkkaan, mitä halutaan näyttää: jokainen käyttöliittymäelementti vie tilaa joltakin muulta mahdolliselta elementiltä. Useimpien puhelinten näytölle myös mahtuu vain yksi ikkuna kerrallaan, joten sovellusten olisi oltava omavaraisia. Kaikki tarvittavat toiminnot tulisi siis voida suorittaa saman sovelluksen tai verkkosivun sisältä ja tarvittavan tiedon pitäisi olla saatavilla ilman, että käyttäjän täytyy hakea sitä muualta. Nykyään Android-käyttöjärjestelmää käytävissä puhelimissa tosin on olemassa myös eräänlainen jaetun näytön tila, jossa käyttäjä voi käyttää kahta sovellusta yhtä aikaa (Bunton 2016). Kaikki sovellukset eivät kuitenkaan tue

tätä, ja jaetun näytön tilassa yhdelle sovellukselle varattu tila on tietenkin vielä pienempi kuin koko näytön tilassa.

Benyonin (2014, 436) mukaan rajallisen näyttötilan vuoksi osa toiminnoista on aina piilossa esimerkiksi valikoiden takana. Joitakin asioita ei välttämättä voida sijoittaa näytölle ollenkaan, vaan esimerkiksi palautteen antamiseen käyttäjälle täytyy toisinaan käyttää muitakin kuin visuaalisia keinoja: palaute voi perustua kuulo- tai tuntoaistiin, esimerkiksi merkkiääninä tai värinäpalautteenä. Haptista eli tuntoaistiin perustuvaa palautetta käytetäänkin yhä enemmän erilaisiin tarkoituksiin sekä mobiilisovellusten että esimerkiksi virtuaalitodellisuuden ja älyasusteiden kehityksessä, ja sillä on myös saavutettavuuteen liittyviä sovellusmahdollisuuksia (ks. esim. Moss 2015). Jos haptisen käyttöliittymäkerroksen monipuolinen käyttö mobiilisovelluksissa yleistyy, tämä voi luoda oman haasteensa käyttäjädokumentaatiolle: kuinka selittää käyttäjälle eri palautteiden merkitys? Voidaanko haptisesta palautteesta tehdä niin intuitiivista, ettei ohjeistukselle ole tarvetta? En usko tämän kysymyksen kuitenkaan nousevan esiin omassa analyysissäni, sillä monipuolisen värinäpalautteen käyttö mobiilisovelluksissa on toistaiseksi varsin vähäistä. Blake Sirach (2017) huomauttaa, että haptisen palautteen liiallinen käyttö voi myös häiritä käyttäjää, minkä vuoksi sitä tulisikin hyödyntää säästeliäästi.

Toinen mobiililaitteiden käytön haaste on jo mainitsemani käyttösessioiden lyhyys ja alttius ulkoisille häiriöille. Budiun (2015) mukaan keskimääräinen käyttösessio – aika, jonka käyttäjä on syventyneenä sovelluksen puoleen ennen kuin hänen huomionsa suuntautuu muualle – on mobiililaitteella 72 sekuntia, kun taas työpöytäsovelluksissa se on keskimäärin 150 sekuntia. Toisin kuin tietokonesovelluksia, mobiilisovelluksia käytetään myös liikkeellä oltaessa, jolloin keskeytyksiä voi tulla usein. Tämän vuoksi takaisin käsillä olleen tehtävän pariin palaaminen on tehtävä mahdollisimman vaivattomaksi. Tähän auttavat mahdollisimman yksinkertaiset tehtävät ja toiminnot sekä tärkeimpien asioiden priorisointi: käyttäjän pitäisi päästä käsiksi tarvitsemiinsa toimintoihin mahdollisimman helposti. Myös sovelluksen tilan tallentuminen siten, että käyttäjä pääsee palaamaan siihen, mihin jäi viimeksi, on erittäin tärkeä osa sujuvaa käyttäjäkokemusta. (mt.) Käyttäjädokumentaation kannalta tämä merkitsee sitä, että dokumentaation ja itse sovelluksen välillä siirtymisen tulisi olla mahdollisimman vaivatonta ja että käyttäjän tulisi voida palata dokumentaatiosta suoraan käsillä olevan tehtävän pariin. Tähän auttaisi kontekstisensitiivinen eli näkymäkohtainen dokumentaatio: käyttäjä pääsee suoraan käsiksi juuri senhetkistä näkymää tai tehtävää koskevaan dokumentaatioon (Android Developers 2017).



Myös kosketusnäytöstä on sekä hyötyä että haittaa. Kosketusnäyttö mahdollistaa sovellusten ohjaamisen eleillä, jotka ovat hyvin suunniteltuna kätevä lisä, eräänlainen näkymätön lisäkerros käyttöliittymään, mutta sovelluskohtaiset eleet ovat usein vaikeasti löydettäviä ja muistettavia. Tunnettu esimerkki sovelluksesta, jossa eleet ovat tärkeässä osassa, on deittisovellus Tinder (2018): sovellus näyttää käyttäjälle lähistöllä olevien ihmisten profiileja ja käyttäjä ilmaisee kiinnostuksensa pyyhkäisemällä oikealle – jos ei ole kiinnostunut, vasemmalle. Käyttäjän olisi mahdotonta tietää, miten sovellus toimii, jos ei sitä kerrottaisi. Sen sijaan laajemmin käytössä olevat eleet, kuten näytön vierittäminen, ovat hyvin tunnettuja ja siten laajasti hyödynnettävissä. Tällaisia käyttöjärjestelmänlaajuisesti johdonmukaisia toimintoja ei myöskään ole syytä selittää yksittäisen sovelluksen käyttäjädokumentaatiossa. Uskon kuitenkin, että sovelluskohtaisten eleiden hyötyjä voisi tukea hyvällä käyttäjädokumentaatiolla: asia nousee esiin esimerkiksi Googlen Android-sovellusten käyttäjädokumentaatiota koskevissa ohjeistuksissa (Android Developers 2017). Niissä sovelluksen käytön kannalta keskeiset eleet, joita käyttäjä ei itse välttämättä löytäisi, mainitaan asiana, jonka osoittaminen käytönaikaisessa käyttäjädokumentaatiossa on tärkeää.

Kosketusnäytön ongelmia puolestaan ovat näppäilyn vaikeus, koska kosketusnäytöllä oleva näppäimistö ei tarjoa samanlaista fyysistä palautetta kuin fyysinen näppäimistö, ja vahinkopainallukset, kun näytöllä on useita kohteita. Näytön ja siten myös näytöllä olevien painikkeiden ja muiden käyttöliittymäelementtien pieni koko lisää vahinkopainalluksia. (Budi 2015.) Esimerkiksi Apple ohjeistaakin, että iOS-sovelluksissa kaikkien painikkeiden tulisi olla kooltaan vähintään 44x44 pikseliä (Apple Developers 2018).

Nopeiden internetyhteyksien yleistymisestä huolimatta hitaat yhteydet aiheuttavat yhä ajoittain ongelmia. Siksi olisi hyvä suunnitella sovellusten näkymistä kevyitä, kuitenkin niin, että yksi sivu sisältää mahdollisimman paljon sisältöä: näin saadaan minimoitua sivunlatausten määrää. (Budi 2015.) Toisaalta monissa sovelluksissa ainakin osa näkymistä toimii täysin ilman verkkoyhteyttä. Myös laitteen muistin ja tallennustilan rajallisuus sekä akkuvirran vähäisyys voivat vaikuttaa käyttäjäkokemukseen negatiivisesti (Benyon 2014, 436).

Älypuhelin vahvuuksiksi Budi (2015) mainitsee sellaiset ominaisuudet, joita esimerkiksi tietokoneissa ei useimmiten ole: esimerkiksi paikannusta, kameraa ja mikrofonia tulisi hyödyntää sovelluksissa mahdollisimman paljon siten, että käyttäjä pääsisi mahdollisimman vähällä vaivalla. Kameraa voidaan käyttää vaikkapa viivakoodien ja luottokorttien lukemiseen, jolloin käyttäjän ei tarvitse itse näppäillä numerosarjoja, ja salasanojen sijaan voidaan käyttää

sormenjälkitunnistusta. Myös liikettä voidaan hyödyntää mobiilisovellusten ohjauksessa: puhelinta voidaan esimerkiksi käänellä ja ravistaa. Liikeohjausta käytetään eniten hyväksi mobiilipeleissä (ks. esim. Ellis 2017).

Mobiilisovellusten käyttäjäkokemukseen vaikuttavat siis paitsi yleiset käyttäjäkokemussuunnitteluun liittyvät periaatteet ja haasteet, myös eräät erityisesti mobiililaitteisiin liittyvät piirteet. Kuten esimerkiksi verkkosivujen, myös mobiilisovellusten käyttäjäkokemussuunnittelu rakentuu sovelluskehittäjän näkökulmasta eri tasoista, jotka Garrett (2002, 21–36) on esitellyt. Käyttäjän näkökulmasta käyttäjäkokemus rakentuu hieman erilaisista elementeistä, joita käyttäjä ei välttämättä havaitse erikseen, vaan tuotteen kieli, ulkonäkö, käyttöliittymäsuunnittelu ja muut osa-alueet näyttäytyvät käyttäjälle yhtenä kokonaisuutena (Cummings 2010). Kuitenkin esimerkiksi vuorovaikutussuunnittelussa on todennäköisesti eroja verkkosivuihin ja työpöytäsovelluksiin verrattuna, koska paitsi mobiilisovellusten käyttökontekstit, myös vuorovaikutuksen tavat eroavat tietokoneille suunnitelluista sivustoista ja sovelluksista: hiiriosoitimen ja näppäinkomentojen sijaan käyttäjä ohjaa mobiilisovellusta useimmiten kosketuksella, joskus myös laitetta kääntelemällä tai ravistamalla.

### **2.3 Mobiilisovellusten käytettävyySPIIRTEET**

Siinä, missä käyttäjäkokemus kattaa kokonaisvaltaisesti ne elämykset, jotka käyttäjälle järjestelmän käytöstä syntyvät, käytettävyyden näkökulma on tavoitekeskeisempi. Käytettävyys keskittyykin siihen, että käyttäjä saavuttaa tavoitteensa. Muiden muassa Google (Android Developers 2017) ja Microsoft (Windows Dev Center 2017) ohjeistavat, että sovelluksista pitäisi pyrkiä tekemään niin intuitiivisia ja helposti käytettäviä, ettei käyttäjädokumentaatiota tarvittaisi ollenkaan. Tätä tavoitetta on todellisuudessa vaikea saavuttaa muun muassa siksi, että eri käyttäjillä voi olla hyvin erilaisia tarpeita, mutta hyvään käytettävyyteen kannattaa silti pyrkiä. Nielsen (2012) toteaa käytettävyyden olevan ehdottoman tärkeää, sillä jos esimerkiksi verkkosivua on vaikea käyttää, käyttäjät siirtyvät muualle. Saman voi olettaa pitävän paikkansa myös mobiilisovellusten kohdalla.

Tässä luvussa tarkastelen mobiilisovellusten käytettävyySPIIRTEITÄ ja niiden suhdetta sovellusten käyttäjädokumentaatioon. Erilaisilla järjestelmillä on erilaisia käytettävyySPIIRTEITÄ, mutta yhden hyvän yleisluontoisen kuvauksen on luonut Benyon (2014, 81). Hän kuvailee pitkälti Nielsenin (1993) mukailleen käytettävyydeltään hyvää järjestelmää sellaiseksi, jolla käyttäjä pystyy suorittamaan tehtävänsä ilman kohtuutonta vaivannäköä, joka sisältää tarvittavat

toiminnot ja informaatioisällön tarkoitukseen sopivasti järjesteltynä, jonka käytön oppiminen ja muistaminen on helppoa, jonka käyttäminen on turvallista kaikissa käyttötilanteissa ja joka on hyödyllinen eli käyttäjä saa tehtyä sillä sen, mitä haluaa tehdä. Nämä käytettävyyssiirteet sopivat monilta osin myös mobiilisovellusten käytettävyyden määrittelyyn.

Hartmut Hoehle, Ruba Aljafari ja Viswanath Venkatesh (2016, 35, 50) toteavat, että vaikka vuonna 2011 jo 70 prosenttia yrityksistä kehitti mobiilistrategiaa (Forrester Research 2011, Hoehlen ym. 2016 mukaan)<sup>1</sup>, mobiilisovellusten käytettävyyssuunnitteluun ja -arviointiin on olemassa varsin vähän järjestelmällistä ohjeistusta. Hoehlen ja Venkateshin (2015, 436–437) mukaan mobiilisovellusten käytettävyydessä on usein puutteita, koska suunnittelussa ei oteta huomioon mobiililaitteiden erityispiirteitä ja konteksteja. Myöskään mobiilisovellusten käytettävyyssuunnittelu ei Hoehlen ja Venkateshin (mts. 440) mukaan ota mobiililaitteiden erityispiirteitä huomioon, minkä lisäksi he kritisoiivat tutkimuskenttää muun muassa siitä, että tutkimus on hajanaista eikä rakennu riittävästi aiemman tutkimuksen päälle: tutkijat tarkastelevat käytettävyyttä erilaisia mittareilla käyttäen erilaisia käsitteitä. Hoehle ja Venkatesh (mt.) pyrkivätkin käsitteellistämään mobiilisovellusten käytettävyyttä ja luomaan kattavan listan mobiilisovellusten käytettävyyssiirteistä. Tässä he (mts. 440) käyttävät apunaan Applen käytettävyysohjeistuksia perustellen valintaansa sillä, että Applen mobiililaitteet ovat hyvin suosittuja ja niiden käytettävyyttä on keuhuttu laajalti.

Applen käytettävyysohjeita tarkastelemalla Hoehle ja Venkatesh (2015, 454) ovat johtaneet 19 käytettävyyssiirrettä, jotka he jakavat kuuteen kategoriaan (suomennokset omiani):

- Sovelluksen suunnittelu
  - *Sovelias brändäys*: brändäys on hillittyä.
  - *Tiedon tallennus*: sovellus tallentaa käyttäjän etenemisen siten, ettei tietoa menetetä, jos käyttäjä poistuu sovelluksesta.
  - *Nopea käynnistys*: käyttäjä ei joudu odottamaan sovelluksen käynnistymistä.
  - *Näytön suunta*: sovellus tukee käyttöä sekä pysty- että vaakanäkymässä.
- Sovelluksen käyttökelpoisuus
  - *Yhteistoiminta*: sovelluksessa voi tarpeen mukaan esimerkiksi viestiä muiden käyttäjien kanssa tai jakaa sijainteja tai muuta informaatiota.

---

<sup>1</sup> Forrester Research, 2011. *How Mature is Your Mobile Strategy?*

- *Sisällön merkityksellisyys*: sovellus keskittyy päätehtäväänsä ja käyttäjälle merkitykselliseen sisältöön.
- *Haku*: nopea ja hyödyllinen hakutoiminto.
- Käyttöliittymän grafiikat
  - *Esteettiset, korkeatasoiset grafiikat*.
  - *Realismi*: sovellus jäljittelee todellisuutta.
  - *Hienovarainen animaatio*: sovellus käyttää hienovaraisia animaatioita esimerkiksi palautteen antamiseen käyttäjälle.
- Käyttöliittymän syöte
  - *Helppo ohjattavuus*: sovelluksen painikkeet ja muut ohjauselementit ovat selkeitä, ilmeisiä ja johdonmukaisia, mahdollisuuksien mukaan myös ennestään tuttuja.
  - *Asetusten painoarvon vähentäminen*: kun sovellus toimii oletusarvoisesti siten kuin useimmat käyttäjät odottavat, asetuksia tarvitaan vähemmän.
  - *Vaivaton käyttö*: käyttäjien toimintaa helpotetaan minimoimalla tekstin näppäily. Sovellus hakee tarvittavan tiedon automaattisesti käyttöjärjestelmästä tai muista sovelluksista, jos mahdollista.
  - *Tarpeeksi suuret painikkeet*: painikkeiden vähimmäiskoko on sormenpään koko eli noin 44 x 44 pikseliä.
- Käyttöliittymän ulosanti
  - *Kielen ytimekkyys*: käyttäjät sisäistävät lyhyen käyttöliittymätekstin nopeasti. Käyttöliittymässä ei saa olla kielivirheitä.
  - *Käyttöliittymäelementtien johdonmukainen käyttö*: sovelluksen käyttöliittymäelementit toimivat käyttäjän odottamalla tavalla.
  - *Käyttäjakeskeinen termistö*: sovelluksen kieli on käyttäjälle tuttua ja välttää erikoistermistöä.
- Käyttöliittymän rakenne
  - *Looginen eteneminen*: sovelluksen sisällä eteneminen on loogista ja ennakoitavaa ja taaksepäin palaaminen on mahdollista.
  - *Eteneminen ylhäältä alas*: tärkein tieto sijoittuu näkymän yläpäähän ja alaspäin mentäessä tieto etenee yleisestä yksityiskohtaiseen.

Hoehle ja Venkatesh (mts. 445–447) myös esittävät nämä käytettävyySPIIRTEET suhteessa aiempaan tutkimukseen mainiten jokaisen piirteen kohdalla tutkimuksen, jossa sitä on käsitelty.

Näin he vastaavat omaan vaatimukseensa tutkimuksen rakentamisesta aiemman tutkimuksen päälle.

Näissä käytettävyysspiirteissä tulevatkin hyvin esiin mobiilisovellusten erityiset käyttökontekstit ja mobiililaitteiden erityispiirteet. Käytön nopeus, helppous ja vaivattomuus ovat keskeisiä, minkä lisäksi esimerkiksi tiedon jatkuva tallennus, näytön pysty- ja vaakanäkymien tukeminen ja tarpeeksi suuret kuvakkeet liittyvät aivan erityisesti mobiililaitteiden käyttötapaan ja teknisiin ominaisuuksiin, joita käsitelinkin jo edellisessä luvussa käyttäjäkokemuksen yhteydessä. Positiivinen käyttäjäkokemus edellyttää hyvää käytettävyyttä.

Hoehlen ja Venkateshin (2015) käytettävyysspiirteissä on myös selkeästi käyttäjädokumentaatiota koskevia asioita. Käyttöliittymän ulosantiin liittyvät kielen ytimekkyys ja käyttäjäkeskeinen termistö ovat käyttäjädokumentaation kannalta hyvin tärkeitä ja niissä on yhtymäkohtia muun muassa minimalismiin, jota käsitelen tämän tutkielman luvussa 3: minimalismin keskeisiin periaatteisiin kuuluu muun muassa se, että käyttäjälle annetaan vain välttämätön määrä tietoa, jolloin tämä pääsee heti tarttumaan toimeen. Näitä käyttäjädokumentaatiota koskevia käytettävyysspiirteitä tulen myös omassa tutkimuksessani soveltamaan.

Tekninen viestijä ei yleensä pääse vaikuttamaan tuotteen itsensä käytettävyyteen. Useimmat tuotteet kuitenkin vaativat käyttäjädokumentaatiota, ja tekninen viestijä voikin hyvällä dokumentaatiolla jossain määrin paikata puutteita tuotteen käytettävyydessä ja intuitiivisuudessa. Hyvä informaation suunnittelu sovelluksessa vähentää dokumentaation tarvetta, sillä käyttäjän on helpompi hahmottaa käyttämäänsä sovellusta ja sen sisältöä. Kuitenkin käytännössä jokainen sovellus vaatii ainakin jonkin verran dokumentaatiota, ja hyvä informaation suunnittelu dokumentaatioissa taas lisää ohjeistuksen käytettävyyttä.

Ilmeisimpiä mobiilisovellusten käytettävyysspuutteita, joita dokumentaatiolla voidaan paikata, ovat puutteet sovelluksen ohjausten ja toiminnallisuuksien selkeydessä. Sovellusten perusohjausten tulisi aina noudattaa alustan konventioita, mutta esimerkiksi eleillä tehtäviä ohjauksia voidaan joutua esittelemään käyttäjälle. Hoehle ja Venkatesh (2015, 454) mainitsevatkin käytettävyysspiirteissään sekä ohjausten tuttuuden että käyttöliittymäelementtien, kuten painikkeiden, johdonmukaisen käytön. Nämä vähentävät ohjeistuksen tarvetta. Myöskään se, mitä sovelluksella ylipäätään voi tehdä eli mitä toimintoja se sisältää, ei välttämättä aina ole käyttäjälle itsestään selvää. Usein keskeisimpiä toimintoja esitelläänkin sovelluksen ensimmäisellä käyttökerralla.

Mobiilisovellusten käytettävyyssiirteet ovat siis monilta osin samankaltaisia kuin muiden vuorovaikutteisten järjestelmien, mutta myös erityispiirteitä on runsaasti. Näiden piirteiden onnistunut soveltaminen tekee sovelluksesta intuitiivisemmän, ja mitä intuitiivisempi sovellus on, sitä helpompi käyttäjän on sitä käyttää, mikä puolestaan vaikuttaa siihen, paljonko käyttäjädokumentaatiota tarvitaan.

### 3 Minimalismi teknisessä viestinnässä

Tässä luvussa käsittelen minimalismia. Tarkastelen ensin sitä, miten minimalismia on kuvattu teknisen viestinnän tutkimuskirjallisuudessa ja pohdin sitten, miten minimalismia voisi soveltaa erityisesti mobiilisovellusten käyttäjädokumentaatioon.

#### 3.1 Minimalismin periaatteet

Minimalismi syntyi vastaliikkeenä niin kutsutulle systemaattiselle lähestymistavalle (*systems approach*) dokumentaatioon. Carrollin (1990, 4) mukaan systemaattisen lähestymisen heikkous on se, että sitä ei ole suunniteltu aktiivisen oppimisen tueksi. Kun tietotekniikka tuli 1970-luvun loppupuolella osaksi useampien ihmisten arkea, nämä ihmiset eivät niinkään olleet kiinnostuneita tietotekniikasta itsessään vaan heidän tarpeensa ja tavoitteensa liittyivät niihin tehtäviin, joiden avuksi uusi tietotekniikka oli tullut: esimerkiksi erilaisten tekstien tuottamiseen. Perinteisesti tietotekniset manuaalit oli suunniteltu enemmän tietotekniikka-asiantuntijoiden tarpeisiin, mikä oli turhauttavaa ihmisille, joille tietotekniikka oli työkalu eikä päämäärä itsessään, kuten juristeille ja sihteereille. Syntyikin uusi tapa opettaa tietotekniikkaa: käyttöohjeet, jotka koostuivat oppimiskokonaisuuksista ja harjoituksista, kun aiemmat tietotekniset manuaalit olivat sisältäneet lähinnä aakkostettuja luetteloita erilaisista toiminnoista. Nämä käyttöohjeet ammensivat systemaattisesta lähestymistavasta.

Carrollin (mts. 73–75) mukaan systemaattinen lähestymistapa antaa liikaa painoarvoa systemaattisuudelle: jokainen tehtävä jaetaan pieniin osasiin, opittaviin taitoihin. Näitä taitoja opetetaan käyttäjälle alkaen aivan alkeistasolta, esimerkiksi hiiren osoittimen liikuttamisesta, ja kun edelliset taidot on opittu, lisätään niihin jotain uutta. Tämä lähestymistapa ei ota huomioon käyttäjää tai käyttökontekstia eikä myöskään oppimisen edellytyksiä. Kun kaikki on valmiiksi avattu, käyttäjän tai oppijan tarvitsee vain seurata ohjeita askel askeleelta: hän ei pääse soveltamaan aiempia tietojaan eikä tunne hallitsevansa tekemisiään itse. Oppiminen jää ohueksi. Minimalismi vastaa tähän erilaisin keinoin, muun muassa tasapainoilemalla käyttäjän ennakkotietojen ja uuden tiedon välillä ja rakentamalla dokumentaatiota aitojen tehtävien, ei systemaattisten oppimiskokonaisuuksien ympärille. Minimalismin keskeisimpänä periaatteena voidaankin pitää käyttäjäkeskeisyyttä.

Stephen W. Draper (1998, 350–352) käy läpi minimalismia edeltäneen dokumentaation ongelmia. Toisinaan käyttöohjetta ei ole ollut saatavilla ollenkaan, koska tuotteen kehittäjät ovat ajatelleet, että käyttäjä kyllä oppii tekemällä. Vastakkainen ongelma taas on ollut se, että

onkin kerrottu käyttäjälle tuotteesta ihan kaikki: edellä mainittu *systems approach* edustaa tällaista lähestymistä. Myös jo mainitsemani toimintokeskeisyys dokumentaatiossa on yksi Draperin esittämistä ongelmista.

Carrollin (1990, 7) mukaan minimalismin tavoite on, että oppiminen tehdään käyttäjälle mahdollisimman helpoksi: poistetaan oppimisen esteet. Van der Meij ja Carroll (1995) ovat määritelleet minimalismin pääperiaatteet ja niihin liittyvät heuristiikat. Periaatteet ovat seuraavat (mts. 245):

1. Toimintaorientaatio
2. Todelliset tehtävät
3. Virheistä oppiminen
4. Tekeminen, oppiminen, löydettävyys

Käsittelen van der Meij'n ja Carrollin periaatteita ja heuristiikkoja lähemmin luvussa 3.3, sillä hyödynnän niitä omien heuristiikkojeni laatimisessa. Vaikka heuristiikkoja ei yleensä käytetä tyylioppaina vaan käytettävyyden arvioimisen työkaluna, van der Meij ja Carroll (mts. 244) mainitsevat, että nämä periaatteet ja heuristiikat on tarkoitettu ohjenuoraksi minimalistista dokumentaatiota laativille.

Myös Draper (1998, 360) toteaa, että minimalismin kaikkien piirteiden voidaan katsoa juontuvan käyttäjakeskeisyydestä. Hän (mts. 350, 368) tiivistääkin minimalismin ytimen mielestäni hyvin: koska nykyiset käyttöliittymät ovat usein helppokäyttöisiä, dokumentaation ei tarvitse esitellä jokaista toimintoa. Minimalistinen dokumentaatio keskittyykin sen sijaan kertomaan, mitä tehtäviä tuotteella voi suorittaa ja miten niitä suoritetaan, minkä lisäksi se paikkaa tuotteen käyttöliittymän käytettävyydsuutteita. Minimalistisessa dokumentaatiossa olennaista onkin käyttöliittymän ja dokumentaation tasapaino ja toisiaan täydentävä suhde (mts. 357). Mobiilisovelluksissa helppokäyttöisyys korostuu entisestään: eräänä niiden tärkeimmistä käytettävyydsuutteista voidaan pitää sitä, että sovellus toimii kuten käyttäjä odottaa sen toimivan (Hoehle & Venkatesh 2015, 443), mikä myös vähentää perustoimintojen esittelyn tarvetta dokumentaatiossa.

Draper (mts. 366–367) esittää myös havainnollisesti, kuinka minimalismi toteutuu dokumentaation eri tasoilla: yksittäisissä toiminnoissa, yksittäisissä tehtävissä ja kokonaisissa tehtäväsarjoissa. Minimalismi toimii jokaisella tasolla eri tavalla: yksittäisten toimintojen kohdalla se merkitsee ylimääräisten yksityiskohtien karsimista käyttämällä hyväksi



käyttöliittymän ja dokumentaation yhteyttä, mikä parantaa vaiheiden ymmärrettävyyttä ja seurattavuutta. Yksittäisten tehtävien tasolla se merkitsee johdonmukaista tapaa jakaa tehtävä osiin siten, että esimerkiksi tehtävän vaiheet ja varoitukset ovat välittömästi tunnistettavissa, jolloin käyttäjä pystyy ohittamaan itselleen tarpeettomia osioita. Tällöin saadaan minimoitua se lukemisen määrä, joka käyttäjältä vaaditaan, vaikka tarjolla olevan tiedon määrä sinänsä ei olisi minimaalinen. Tehtäväsarjojen tasolla kyse on siitä, kuinka käyttäjää autetaan löytämään tarvitsemansa ohjeistus. Tämä voidaan toteuttaa esimerkiksi käyttämällä sisällysluetteloa, mikä onkin paria sivua pidempien käyttöohjeiden kohdalla lähes välttämätöntä.

Minimalismi on teknisessä viestinnässä hyvin suosittu ajatus, jonka tutkimus on kuitenkin viime vuosina jäänyt vähemmälle. Siihen liittyy myös monenlaisia harhaluuloja ja väärinkäsityksiä, joita Carroll ja van der Meij (1998, 55–56) käsittelevätkin artikkelissaan *Ten Misconceptions about Minimalism*. He painottavat, että vaikka minimalismi rakentuukin tehtäväkeskeisyyden ympärille, se sisältää useita tekijöitä, jotka ovat sidoksissa toisiinsa. Minimalismin ajatusta yksinkertaistetaan usein: yksi yleinen käsitys on se, että minimalismi tarkoittaa tekstimäärän vähentämistä. Muita yleisiä luuloja ovat muun muassa se, että minimalistisen dokumentaation lukijat oppisivat pääasiassa yrittämällä ja erehtymällä ja että minimalismi toimisi vain yksinkertaisille tuotteille.

Draper (1998, 362) toteaaakin minimalismin olevan harhaanjohtava nimi juuri siksi, että minimalismi ei aina keskity tekemään dokumentaatiosta mahdollisimman harvasanaista. Esimerkiksi virhetiedon painottaminen yleensä pikemminkin lisää dokumentaation pituutta kuin vähentää sitä. Draper ehdottaakin, että *toimintakeskeinen dokumentaatio* olisi osuvampi nimi. Taipuisin itse olemaan samaa mieltä Draperin kanssa, sillä nimi *minimalismi* aiheuttaa väärinkäsityksiä ja vääristää mielikuvaa siitä, mikä tämän lähestymistavan ydin on. Väärinkäsitysten runsautta voineekin pitää yhtenä minimalismin haasteista, joita käsittelen seuraavaksi.

### **3.2 Minimalismin haasteita**

Stephanie Rosenbaum (1998, 122) tarkastelee minimalistisen dokumentaation kirjoittamisen haasteita monipuolisesti. Rosenbaumin tutkimuksessa esiin tulleita käytännön haasteita olivat muun muassa se, kuinka paljon tietoa voi jättää pois: kuinka paljon voidaan jättää käyttäjän oman tiedon, kokemuksen ja ajattelun varaan? David K. Farkas (1998, 249) huomauttaa, että jos tietoa ei ole saatavilla tarpeeksi, käyttäjä ei välttämättä pysty suorittamaan tehtävää

onnistuneesti tai tehtävän suorittamiseen voi mennä toivottua enemmän aikaa. Vaikka käyttäjä saisikin tehtävän tehtyä, hänelle saattaa syntyä tuotteesta väärä mentaalinen malli, josta on haittaa myöhemmin. Myös Draper (1998, 354) mainitsee selitysten määrän tasapainottamisen yhtenä minimalismin vaikeuksista. Farkasin (1998, 250) mukaan tekniset viestijät kokevat, ettei heillä ole tarpeeksi tietoa käyttäjistä, eivätkä tuotekehitysorganisaatioiden resurssit riitä dokumentaation iteratiiviseen käyttäjätestaukseen. Mikäli dokumentaation kirjoittajilla olisi enemmän tietoa käyttäjistä, heidän olisi helpompaa päätellä, millainen määrä informaatiota on tarpeeksi. Käyttäjien tuntemus olisi hyvin tärkeää aidon käyttäjakeskeisyyden saavuttamiseksi.

Toinen merkittävä haaste on vaikeus erottaa aidot tehtävät tehtävistä, jotka vain esittelevät tuotteen toimintoja. Tähän liittyy vaikeus rajata tarpeeksi pieniä osioita, jotta minimalismi toteutuisi, koska aidot tehtävät ovat usein liian laajoja. Ratkaisuksi tarjotaan realististen tehtävien luomista aitojen sijaan. (Rosenbaum 1998, 123–124.) Uskoisin, että mobiilisovellusten kohdalla tämä ongelma ei ole niin yleinen, sillä niissä suoritettavat tehtävät ovat yksinkertaisempia kuin työpöytäsovelluksissa, jotka ovat usein varsin laajoja.

Janice Redish (1998, 220–222) taas tuo esiin sen, että minimalismi nojaa ajatukseen käyttäjästä, joka on aktiivinen ja oppimisorientoitunut ja joka oppii parhaiten tekemällä itse. Kaikki käyttäjät eivät kuitenkaan ole tällaisia, vaan ihmiset ovat erilaisia persoonia ja heillä on erilaisia oppimistyyylejä. Samallakin käyttäjällä on erilaisia moodeja, joiden puitteissa hän käyttäjädokumentaatiota lähestyy: toisinaan käyttäjä saattaa lukea dokumentaatiota oppiakseen, toisinaan taas saadaksesi tehtävän suoritettua. Minimalismi myös keskittyy usein uusien käyttäjien opastamiseen, jolloin tuotteen jo tuntevien käyttäjien tukeminen jää vähemmälle huomiolle. Myös Draper (1998, 364–365) toteaa, että käyttäjät toimivat eri tavalla riippuen siitä, mikä heidän tavoitteensa on. Jos käyttäjän tavoitteena on oppia, hän saattaa hyvinkin kokeilla tuotteen toimintoja itsenäisesti, mutta jos käyttäjä haluaakin vain saada työnsä tehtyä, hän vain seuraa tarpeellisia ohjeita eikä välttämättä varsinaisesti opi mitään. Samanlainen dokumentaatio ei välttämättä toimi kaikille käyttäjille tai edes samalle käyttäjälle kaikissa tilanteissa: tältäkin kannalta olisi tärkeää, että teknisellä viestijällä olisi tietoa käyttäjistä, joille hän kirjoittaa.

Yksi keino erilaisten käyttäjien tarpeiden huomioimiseen on kerrostaminen. Kerrostamisella tarkoitetaan sitä, että dokumentaation eri osat on suunnattu erilaisille käyttäjäryhmille (Huckin & Olsen 1991, 65): saatetaan esimerkiksi merkitä edistyneemmille käyttäjille tarkoitettu informaatio siten, että kokeneet käyttäjät löytävät sen helposti ja aloittelevat käyttäjät osaavat

ohittaa sen. Farkas (1998, 254) käsittelee kerrostamista minimalismin lisänä ja toteaa, että vaikka kerrostaminen lisää tekstin määrää, se ei silti ole mitenkään minimalismin periaatteiden vastainen vaan päinvastoin tukee sitä, sillä käyttäjä ei joudu lukemaan dokumentaatiota, jota hän ei tarvitse. William Horton (1994,178–179) toteaa, että kerrostaminen toimii paremmin digitaalisessa kuin painetussa dokumentaatioissa, sillä digitaalisessa dokumentaatioissa voidaan hyödyntää muun muassa hyperlinkkejä.

On huomattava, että minimalismi on hyvin ohjelmistokeskeinen lähestymistapa dokumentaatioon, eivätkä sen periaatteet välttämättä sovellu erilaisten laitteiden ja koneiden dokumentaatioon. Esimerkiksi minimalismin korostama kokeileva oppiminen ei useinkaan sovellu vaikkapa kodinkoneiden tai teollisuuskoneiden käytön opetteluun: paperikoneen käyttöä ei voi lähestyä kokeilemalla, mitä mistäkin napista mahtaa tapahtua, vaan käyttöohjeiden on oltava hyvin tarkat, jottei henkilö- tai taloudellisia vahinkoja syntyisi. Omaan tutkimukseeni minimalismin siis lähtökohtaisesti pitäisi sopia, koska tutkin juuri sovelluksia. Seuraavaksi esittelen van der Meij'n ja Carrollin (1995) minimalismiheuristiikat, jotka avaavat minimalismia perusteellisemmin.

### **3.3 Van der Meij'n ja Carrollin minimalismiheuristiikat**

Van der Meij ja Carroll julkaisivat vuonna 1995 artikkelin *Principles and Heuristics for Designing Minimalist Instruction*, jossa he esittelivät minimalismin periaatteita ja niihin liittyviä heuristiikkoja. Artikkelin syntyi tarpeesta yksityiskohtaisempaan ohjeistukseen minimalistisen dokumentaation kirjoittajille, sillä yhtenä tärkeimmistä hidasteista minimalistisen lähestymistavan yleistymiselle oli sen suunnitteluperiaatteiden suurpiirteisyys. Artikkelissa esitetyt periaatteet ja heuristiikat on tarkoitettu pääasiassa uusille käyttäjille tarkoitetun ohjeistuksen suunnitteluun, mutta niiden hyödyt eivät rajoitu siihen, vaan van der Meij ja Carroll (1995, 244) uskovat niiden soveltuvan kaikenlaisen ohjeistavan tekstin kehittämiseen. Heidän (mts. 245) heuristiikkansa ovat seuraavat:

#### **Periaate 1: Toimintaorientaatio**

Heuristiikka 1.1: Anna käyttäjän ryhtyä heti toimeen.

Heuristiikka 1.2: Rohkaise käyttäjää kokeilemaan.

Heuristiikka 1.3: Avun pitää aina olla käyttäjän ulottuvilla.

#### **Periaate 2: Todelliset tehtävät**

Heuristiikka 2.1: Keskity käyttäjän aitoihin ydintehtäviin.

Heuristiikka 2.2: Ohjeen tulee heijastaa tehtävän todellista rakennetta.

### **Periaate 3: Virheistä oppiminen**

Heuristiikka 3.1: Estä virheet.

Heuristiikka 3.2: Anna vianmäärittystietoa kohdissa, joissa käyttäjä todennäköisesti tekee virheen tai virheen korjaaminen on vaikeaa.

Heuristiikka 3.3: Tarjoa vianmäärittystietoa: vian havaitseminen, diagnoosi ja ratkaisu.

Heuristiikka 3.4: Sijoita vianetsintätieto mahdollisimman lähelle relevanttia kohtaa.

### **Periaate 4: Tekeminen, oppiminen, löydettävyys**

Heuristiikka 4.1: Kirjoita lyhyesti. Kaikkea ei tarvitse selittää tai kuvailla.

Heuristiikka 4.2: Luo kappaleille keskenään johdonmukaiset ja samankaltaiset aloitukset ja lopetukset.

Minimalismin ensimmäinen periaate painottaa toimintakeskeisyyttä. Dokumentaation lukija haluaa päästä tekemään jotain, mikä on hänelle itselleen mielekästä. Heuristiikka 1.1 ohjeistaakin antamaan käyttäjälle välittömän mahdollisuuden toimia: pitkät johdannot vievät huomiota itse oppimiselta, ja käyttäjälle tulisi tarjota vähemmän lukemista ja enemmän tekemistä. Minimalismi onkin tasapainoilua toiminnan ja tiedon välillä. On arvioitava, minkä verran tietoa on välttämätöntä antaa, jotta käyttäjä pystyy oppimaan. Käyttäjää houkutellaan toimimaan ja oppimaan itse. Heuristiikka 1.2 ohjaa rohkaisemaan ja tukemaan sitä, että käyttäjä kokeilee ja oivaltaa asioita itse. Käyttäjällä tulisi aina olla tunne siitä, että hän hallitsee omaa tekemistään ja että hän voi koska tahansa palata ohjatun tekemisen pariin. Tarvitaan tasapainoa ohjeistuksen ja mielekkyyden välille: van der Meij'n ja Carrollin (mt.) mukaan tämä voidaan saavuttaa käyttämällä kutsuvaa kieltä, tarjoamalla vaihtoehtoisia strategioita oikeissa paikoissa ja antamalla käyttäjän arvioida edistymistään itse ulkopuolisen arvioinnin sijaan. (mts. 244–247.)

Heuristiikka 1.3 koskee käyttäjän omia tavoitteita. Käyttäjän hetkittäiset tavoitteet eivät aina ole niin kauaskantoisia tai hienostuneita kuin sovelluksen tai dokumentaation suunnittelijoiden tavoitteet, ja tämä tulisi ottaa huomioon dokumentaatiossa. Dokumentaatio ei saisi häiritä käyttäjän toimintaa: ratkaisuksi van der Meij ja Carroll (mt.) tarjoavat erilaisia passiivisen dokumentaation muotoja, kuten ohjelmiston sisäisiä apunäyttöjä, jotka näyttävät kuhunkin kontekstiin sopivia ohjeita ja vinkkejä keskeyttämättä käyttäjän toimintaa. (mts. 247.)

Mobiilisovellusten dokumentaatiossa välitön mahdollisuus toimia ei välttämättä aina toteudu. Jos sovelluksessa on dokumentaatiolle erillinen näkymänsä, ei käyttäjä voi yhtä aikaa lukea

ohjeistusta ja kokeilla itse. Jos taas ohjeistus on käytönaikaista eli se näytetään esimerkiksi ensimmäisellä käyttökerralla, kun käyttäjä kokeilee uutta toimintoa, tämä mahdollisuus voi hyvinkin toteutua. Kysymys siitä, minkä verran ohjeistavaa tietoa käyttäjälle on välttämätöntä antaa, on kuitenkin tärkeä myös mobiilisovellusten dokumentaatioissa, ja siitä mainitaan myös esimerkiksi Googlen Android-sovellusten dokumentaatiota koskevissa ohjeistuksissa (Android Developers 2017). Vaikka heuristiikka 1.2 tuntuu kutsuineen ("Try it yourself!") hieman vanhentuneelta esimerkiksi työpöytäsovellusten dokumentaatiota ajatellen, saattaa se mobiilisovellusten käytönaikaisessa dokumentaatioissa toimia hyvinkin, kunhan käyttäjä voi halutessaan jättää kehotuksen huomiotta. Sen sijaan heuristiikka 1.3 on mobiilisovellusten kannalta hankalampi, sillä mobiilisovellusten käyttöliittymässä ei useimmiten ole tilaa passiiviselle dokumentaatiolle.

Minimalismin toinen periaate koskee sitä, ettei käytettävä tuote ole käyttäjälle päämäärä itsessään, vaan työkalu, jonka avulla hän pääsee tavoitteisiinsa. Tuotokeskeisyys voi vieraannuttaa käyttäjää. Toinen periaate liittyykin läheisesti ensimmäiseen, sillä sekin painottaa toimintakeskeisyyttä: heuristiikka 2.1 ohjaa rakentamaan dokumentaation aitojen tehtävien ympärille ja heuristiikka 2.2 koskee tehtävien rakenteen hahmottamista. Dokumentaation tehtävät voivat olla vaatimattomiakin, kunhan ne ovat helposti lähestyttävissä ja käyttäjälle relevantteja. Kun käyttäjä ymmärtää tehtävän merkityksen, hän on myös motivoituneempi tekemään sen. Tehtävillä tulisi myös olla yhteys käyttäjän aiempiin tietoihin, taitoihin ja kokemuksiin, jolloin käyttäjän on helpompi hahmottaa, että kyseessä on aito ja merkityksellinen tehtävä. Tehtävien rakenteen ja niiden välisen suhteen hahmottamiseen auttavat erityisesti kuvaavat otsikot, jotka eivät kuitenkaan ole esimerkkikohtaisia, vaan tarpeeksi yleisellä tasolla, jotta käyttäjä löytää omaan tarpeeseensa sopivat ohjeet. (Van der Meij & Carroll 1995, 284–250.)

Tehtävien aitous on tärkeää myös mobiilisovellusten dokumentaation kannalta. Otsikoita ei käytönaikaisessa dokumentaatioissa välttämättä ole, mutta erillisen käyttäjädokumentaation otsikoita koskevat samat periaatteet kuin muillakin tuotteilla: otsikoiden tulee olla kuvaavia ja tehtäväkeskeisiä (mts. 250).

Kolmas periaate koskee virhetilanteita. Dokumentaation tulisi auttaa käyttäjää tunnistamaan virheet ja palautumaan niistä. Uuden tiedon omaksumiseen voi liittyä ahdistumista ja turhautumista, ja virheiden välttäminen, niiden tunnistaminen ja niistä palautumisen tukeminen voivat vähentää näitä tuntemuksia merkittävästi. Van der Meij'n ja Carrollin (mt.) mukaan eri

tyyppiset virheet vaativat erilaista huomioimista: he jakavat virheet semanttisiin ja syntaktisiin virheisiin sekä lipsahduksiin (*slip*). Semanttinen virhe tapahtuu, kun käyttäjä yrittää suorittaa toiminnon tavalla, jolla se ei voi onnistua, esimerkiksi yrittämällä tallentaa tiedoston napsauttamalla tulostuspainiketta. Syntaktinen virhe taas tapahtuu, kun käyttäjä seuraa oikeaa menetelmää, mutta epäonnistuu sen suorittamisessa, esimerkiksi suorittamalla vaiheet väärässä järjestyksessä. Lipsahdukset ovat pieniä virheitä, jotka tapahtuvat näppäilytasolla: käyttäjä esimerkiksi kirjoittaa tiedostopäätteen väärin. (mts. 250–251.)

Näistä virhetyypeistä syntaktiset virheet ovat helpoimpia ennakoida ja siten myös välttää ottamalla ne huomioon dokumentaatiossa. Semanttiset virheet ja lipsahdukset ovat vaikeampia ennakoida, ja niissä käyttäjiä voidaan parhaiten auttaa antamalla dokumentaatiossa yleispäteviä neuvoja virhetilanteisiin, sillä tarkkoja virhekohtaisia ohjeita on vaikea antaa. Käytettävyydestä on näiden virheiden välttämisen kannalta välttämätöntä. (Van der Meij & Carroll 1995, 250–251.)

Vaikka minimalistisessä dokumentaatiossa yleensä on mahdollisimman vähän tekstiä, virhetieto on poikkeus. Jos käyttäjää tuetaan virhetilanteissa, virheistä voi olla oppimisen kannalta jopa enemmän hyötyä kuin haittaa (mts. 252). Lähtökohtaisesti kuitenkin keskitytään virheiden välttämiseen: heuristiikka 3.1 painottaa juuri välttämistä parhaana tapana käsitellä virhetilanteita. Kun dokumentaatio on selkeää ja ymmärrettävää, käyttäjä ei tee virheitä siksi, että olisi ymmärtänyt ohjeen väärin: käytettävyydestä on hyvä tapa selvittää, mitkä kohdat dokumentaatiossa saattaisivat aiheuttaa väärinkäsityksiä. Tilanteissa ja toiminnoissa, joissa virheiden todennäköisyys on suurempi, kannattaa käyttää visuaalisia keinoja herättämään käyttäjän huomiota: esimerkiksi vinkit erottuvat usein muusta tekstistä, ja niitä voi käyttää ennaltaehkäisemään virheitä. Myös tuotteen sisällä voi olla virheitä ehkäiseviä elementtejä: tuote voi esimerkiksi varmistaa, haluaako käyttäjä suorittaa mahdollisesti haitallisen toiminnon, kuten tekstinkäsittelyohjelman sulkemisen tiedostoa tallentamatta. (Van der Meij & Carroll 1995, 250–252.) Myös heuristiikka 3.2 korostaa virheiden ennaltaehkäisemistä antamalla lisätietoa sellaisten toimintojen yhteydessä, jotka ovat virhealttiita, tai jos virheiden korjaaminen on vaikeaa, esimerkiksi jos toiminto on jollakin tavalla epätyypillinen (mts. 252–253).

Heuristiikat 3.3 ja 3.4 keskittyvät jo tapahtuneiden virheiden käsittelyyn. Dokumentaation tulisi tarjota virheiden tunnistamista ja korjaamista tukevaa tietoa ja tiedon tulisi olla tarjolla silloin, kun käyttäjä sitä tarvitsee: mahdollisimman lähellä sitä toimintoa, johon virhe liittyy. Kun

virhetieto on saatavilla heti, kun virhe tapahtuu, käyttäjää huomaa virheen ennen kuin mitään pahempaa ehtii tapahtua. Se myös auttaa käyttäjää ymmärtämään, mistä virhe johtuu. Jos virhetieto sijaitsee jossain erillään (esimerkiksi erillisessä vianetsintäosiossa), kontekstin antaminen voi olla vaikeaa. Van der Meij ja Carroll (mt.) huomauttavat myös, että virhetieto voi myös herättää käyttäjän uteliaisuuden: käyttäjä saattaa virhetietoa hyväksi käyttäen kokeilla, miten virhe tapahtuu ja miten sitä korjataan. Virhetieto voi siten auttaa käyttäjää oppimaan myös virheistä palautumista sitä kokeilemalla. (mts. 253–254.)

Mobiilisovelluksissa painottuu virheiden ennaltaehkäisy hyvällä käyttöliittymäsuunnittelulla. Kuten jo tämän tutkielman luvussa 2.3 mainitsin, mobiilisovelluksista pyritään tekemään niin intuitiivisia, ettei käyttäjädokumentaatiota ensinkään tarvittaisi. On kuitenkin käytännössä mahdotonta ennakoita aivan kaikkia virhetilanteita edes kattavan käytettävyydestauksen avulla. Mobiilisovelluksissa virhetiedon olisi suurimmilta osin oltava sovelluksen käyttöliittymässä, sillä käyttäjä ei voi samanaikaisesti käyttää sovellusta ja lukea dokumentaatiota, joka sijaitsee omassa näkymässään sovelluksen sisällä tai mahdollisesti jopa sovelluksen ulkopuolella. Mobiilisovelluksissa saattaakin korostua se, että käyttäjälle kerrotaan käyttöliittymässä, mikä merkitys hänen parhaillaan suorittamallaan toiminnolla on. Tähän liittyy esimerkiksi toimintojen varmistaminen. Vaikka van der Meij ja Carroll (mts. 254) kehottavatkin sijoittamaan virheitä koskevan tiedon lähelle sitä toimintoa, jonka yhteydessä virhe saattaa tapahtua, mobiilisovellusten käyttäjädokumentaatioissa erillinen vianetsintäosio saattaa olla paikallaan: jos mobiilisovellusten käyttäjä lähtee etsimään dokumentaatiota vasta siinä vaiheessa, kun jotain on jo mennyt pieleen, voi olla helpompaa, että virhetilanteiden vastaukset on kerätty yhteen paikkaan. Ihanteellista tässäkin tapauksessa toki olisi, jos sovellus tunnistaisi ongelmatilanteen ja tarjoaisi käyttäjälle apua suoraan.

Neljäs periaate koskee dokumentaation rakennetta. Myös rakenteen tulisi tukea käyttäjän toimintaa, oppimista ja uuden löytämistä. Heuristiikka 4.1 ohjeistaa ilmaisemaan vain tarpeellisen: kaikkea ei tarvitse selittää. Tehtävien tulisi olla lyhyehköjä kokonaisuuksia, jotka käyttäjä saa suoritettua nopeasti. Jos selityksiä tarvitaan, niiden olisi oltava lyhyitä (van der Meij & Carroll 1995, 256–257). Van der Meij ja Carroll (mts. 256) myös suosittelevat, että selitykset pitäisi sijoittaa sen tehtävän jälkeen, johon ne liittyvät. Itse olen kuitenkin osittain eri mieltä tästä: jos selitykset ovat tehtävän suorittamisen kannalta tarpeellisia, ne olisi mielestäni sijoitettava joko ennen tehtävää tai sen vaiheen yhteyteen, johon ne liittyvät, sillä käyttäjän ei voi odottaa lukevan koko tehtävän ohjeistusta ennen toimeen ryhtymistä.

Kun kaikkea ei selitetä, käyttäjä tukeutuu olemassa oleviin tietoihinsa ja omaan ajatteluunsa, mikä edistää oppimista. Heuristiikka 4.2 koskee käyttöohjeen lukujen rakennetta. Tämä liittyy dokumentaation modulaarisuuteen: hyvin suunniteltu rakenne tukee sitä, ettei käyttäjän tarvitse lukea koko käyttöohjetta saadakseen tarvitsemansa tiedon. Lukujen tulisi alkaa samasta lähtöpisteestä, jotta käyttäjä ei joudu lukemaan aiempia lukuja päästäkseen tarvittavaan paikkaan, kun taas lukujen lopetusten tulisi vetää asiat yhteen. Tekninen viestijä joutuu pohtimaan, minkä verran käyttäjän voidaan olettaa jo osaavan, jottei tämä joudu etsimään tietoa aiemmista luvuista. (Van der Meij & Carroll 1995, 254–257.)

Mobiilisovellusten käyttäjädokumentaatio eroaa rakenteeltaan perinteisten työpöytäsovellusten dokumentaatiosta, mutta ytimekkyys ja modulaarisuus ovat valttia siinäkin. Asioiden ilmaiseminen lyhyesti on jo käytettävissä olevan tilan rajallisuuden vuoksi tärkeää: käyttäjä tuskin viitsii lukea useampaa ruudunmittaa ohjeistavaa tekstiä.

Van der Meij'n ja Carrollin (1995) minimalistiheuristiikat vaikuttavat siis monilta osin soveltuvan mobiilisovellusten käyttäjädokumentaation tarkasteluun, vaikka jotkin asiat painottuvat mobiilisovelluksissa enemmän ja jotkin taas eivät tunnu niin merkittävilä kuin työpöytäsovellusten kohdalla. Esimerkiksi käyttäjän kannustaminen kokeilemaan sovelluksen toimintoja saattaa toimia paremminkin kuin muissa tuotteissa, kun taas välitön mahdollisuus toimia ei aina toteudu, jollei dokumentaatio ole käytönaikaista. Minimalismin toiminta- ja käyttäjäkeskeisyys on hyvä lähtökohta mobiilisovellusten käyttäjädokumentaatiolle. Seuraavaksi siirrynkä tarkastelemaan minimalismia erityisesti mobiilisovellusten käyttäjädokumentaation näkökulmasta.

### **3.4 Minimalismi mobiilisovellusten käyttäjädokumentaatiossa**

Tässä luvussa tarkastelen minimalismin sovellettavuutta mobiilisovellusten käyttäjädokumentaatioon käyttäen tarkastelun pohjana luvussa 2 esittelemääni tutkimusta mobiilisovellusten käytettävyydestä ja käyttäjäkokemuksesta. Kuten tämän tutkielman johdannossa mainitsin, mobiilisovellusten käyttäjädokumentaatiota on tutkittu hyvin vähän. Vaikka mobiilisovellusten käytettävyyden arviointiin on laadittu monia eri heuristiikkalistoja, olen itse löytänyt ainoastaan yhden heuristiikan, joissa käyttäjädokumentaatio eksplisiittisesti mainitaan: Rosa Yáñez Gómezin, Daniel Cascado Cabaelleron ja José-Luis Sevillanon (2014, 5) mobiilisovellusten käytettävyyshauristiikoissa yhtenä kolmestatoista pääkohdasta ovat ohjeet ja dokumentaatio. Alaheuristiikkoja on heuristiikkojen ensimmäisessä versiossa 158,



joista 16 sijoittuu ohjeiden ja dokumentaation alle. Lopullisessa versiossa Yáñez Gómez ym. (mts. 13) tiivistävät ohjeistukseen ja dokumentaatioon liittyvät alaheuristiikat yhdeksi: ”Focus on one single feature at a time. Present only those instructions that are necessary for the user to get started.”

Mielestäni dokumentaatioon liittyvien käytettävyysskysymysten tiivistäminen yhteen alaheuristiikkaan ei kuitenkaan aivan riitä. Yáñez Gómezin ym. (mt.) heuristiikkojen ensimmäisessä versiossa on useita asioita, jotka on hyvä ottaa huomioon dokumentaatiota suunniteltaessa, ja siksi tarkastelen juuri niitä minimalismin näkökulmasta. Osassa niistä onkin nähtävissä selkeitä yhteyksiä minimalismiin: esimerkiksi heuristiikat 130: *Seuraako ohjeistusten järjestys käyttäjän toimintajärjestystä?*, 135: *Navigaatio: onko käyttäjän helppo löytää tarvitsemansa tieto dokumentaatiosta?* ja 138: *Onko tieto merkityksellistä?* liittyvät asioihin, joita minimalismikin käsittelee. Se, että ohjeistusten järjestys vastaa sitä järjestystä, jossa käyttäjä asioita tekee, on osa minimalismin käyttäjä- ja toimintakeskeisyyttä ja liittyy läheisesti van der Meij’n ja Carrollin (1995, 250) heuristiikkaan 2.2: *Ohjeen tulee heijastaa tehtävän todellista rakennetta*. Tiedon löytämisen helppous taas vähentää lukemisen tarvetta, kun käyttäjä pääsee suoraan tarvitsemansa tiedon äärelle, ja heuristiikka 2.2 liittyy myös tähän, sillä sekä kuvaavat otsikot että luonteva etenemisjärjestys helpottavat tiedon löytämistä.

Myös kysymys dokumentaatioissa tarjotun tiedon merkityksellisyydestä liittyy vahvasti minimalismiin. Merkityksellisyys riippuu aina kontekstista: jos käyttäjää opastetaan tietyn tehtävän tekemisessä, merkityksellisyys on esimerkiksi tehtävä- ja käyttäjäkeskeisyyttä. Tässä kontekstissa merkityksellistä tietoa ei yleensä ole vaikkapa tuotteen tarkka tekninen kuvailu, joka jossain toisessa kontekstissa voikin olla tärkeää. Myös Draperin (1998, 360) ajatus siitä, että tuotteiden jokaista toimintoa ei tarvitse erikseen esitellä käyttäjälle järjestelmien käytettävyyden jatkuvasti kehittyessä, on osa merkityksellisyyden periaatetta: ohjeistetaan käyttäjää vain niissä keskeisissä tehtävissä, joita hän haluaa tuotteella suorittaa.

Myös mobiilisovellusten rajallinen näyttötila ja sen käyttäjälle mahdollisesti aiheuttamat kognitiiviset rasitteet kuten työmuistin kuormittuminen (ks. Budiu 2015; Meyer 2016) asettavat vaatimuksen siitä, että dokumentaatioon valitaan vain käyttäjän kannalta kaikista merkityksellisistä tieto. Aiemmin tarkastelemani kerrostaminen voi myös mobiilisovellusten käyttäjädokumentaatioissa toimia keinona, jolla voidaan huomioida erilaisia käyttäjiä, esimerkiksi aloittelevia ja edistyneempiä. Ihanteellista olisi myös, jos käyttäjä voisi itse säätää

tarvitsemansa dokumentaation määrää, kuten myös Yáñez Gómezin ym. (2014, 10) heuristiikassa 140 todetaan.

Tärkeä kysymys mobiilisovellusten käyttäjädokumentaatiossa on myös sovelluksen ja dokumentaation samanaikaisen käytön helppous. Yáñez Gómez ym. (2014, 10) nostavat tämän esiin heuristiikoissaan 141–143 kysyen muun muassa, onko ohjeistusten ja tehtävän välillä vaihtelu helppoa ja voiko käyttäjä palata dokumentaatiosta suoraan käsillä olevan tehtävän pariin. Kuten todettua, jos käyttäjädokumentaatio ei ole käytönaikaista, käyttäjä ei voi yhtä aikaa lukea dokumentaatiota ja tehdä sitä, mitä on sovelluksella tekemässä, sillä puhelimen näytöllä voi olla vain yksi näkymä kerrallaan. Toisin sanoen käyttäjän täytyy muistaa, mitä dokumentaatiossa sanotaan ja näytetään, kun hän palaa tehtävänsä pariin. Tämä edellyttää sitä, että dokumentaatio on helposti löydettävissä ja että tietoa on helppo löytää dokumentaation sisältä.

Tiedon löydettävyyttä helpottavat esimerkiksi Draperin (1998, 366–367) mainitsemat, dokumentaation eri tasoilla toimivat keinot: hyvä otsikointi ja esimerkiksi sisällysluettelo auttavat käyttäjää löytämään tarvitsemansa ohjeen, kun taas tehtäväsarjan sisällä tiedon löytämistä ja sisäistämistä tukee yksityiskohtien karsiminen ja eri tyyppisen tiedon, kuten tehtävän vaiheiden ja varoitusten, selkeä merkitseminen. Draper (mt.) kehottaa käyttämään hyödyksi käyttöliittymän ja dokumentaation yhteyttä, ja mobiilisovellusten käyttäjädokumentaation sisäistettävyyttä voidaankin parantaa esimerkiksi käyttämällä käyttöliittymästä otettuja kuvia: esimerkiksi tietyn painikkeen ulkonäkö ja sijainti käyttöliittymässä voidaan näyttää kuvassa. Tämä voi auttaa käyttäjää muistamaan ohjeen paremmin sitten, kun hän palaa takaisin tehtävänsä ääreen.

Yáñez Gómezin ym. (2014) heuristiikkojen lisäksi myös luvussa 2.3 esittelemistäni Hoehlen ja Venkateshin (2015, 442–444) keräämistä mobiilisovellusten käytettävyySPIIRTEISTÄ muutamassa on nähtävissä selkeitä yhteyksiä käyttäjädokumentaatioon ja minimalismiin. Kun sovellus toimii kuten käyttäjä odottaa sen toimivan, tarvittavan dokumentaation määrä vähenee ja sen painopiste muuttuu: kuten Draper (1998, 368) toteaa, nykyisten käyttöliittymien helpokäyttöisyys siirtää dokumentaation huomiota pois toimintojen selittämisestä ja kohti tehtäväkeskeisyyttä. Myös mobiilisovelluksissa dokumentaation tulisi keskittyä yksittäisten toimintojen sijaan niihin tehtäviin, joita käyttäjä haluaa suorittaa, joskin käyttäjälle on esiteltävä myös konventioista poikkeavat toiminnot, kuten sovelluskohtaiset eleet (ks. luku 2.2). Hoehlen

ja Venkateshin (mt.) mainitsemista käytettävyySPIIRTEISTÄ myös kielen ytimekkyys ja käyttäjakeskeinen termistö ovat suoraan liitettävissä minimalismiin.

Minimalismi vaikuttaa sopivan mobiilisovellusten käyttäjädokumentaatioon varsin hyvin, sillä sen keskeiset periaatteet, kuten ytimekkyys, käyttäjakeskeisyys ja tiedon löydettävyys ovat hyvin linjassa käsittelemieni käyttäjäkokemuksen ja käytettävyyden piirteiden kanssa. Sitä, kuinka hyvin nämä periaatteet käytännössä sopivat mobiilisovellusten käyttäjädokumentaation analysoimiseen, arvioin tämän tutkielman päätelmissä. Seuraavaksi esittelen tutkimusmenetelmäni, heuristisen asiantuntija-arvioinnin.

## 4 Heuristinen asiantuntija-arviointi

Käytän tässä tutkimuksessa tutkimusmenetelmänä heuristista asiantuntija-arviointia. Koska mobiilisovellusten käyttäjädokumentaatiota ei ole juuri tutkittu eikä näin ollen ole myöskään olemassa juuri tähän tarkoitukseen luotuja heuristiikkoja, luon heuristiikat itse käyttäen hyväksi aineksia luvuista 2 ja 3 sekä Purhon (2000) teknisen dokumentaation käytettävyyshauristiikkoja ja älypuhelinien käyttöjärjestelmäkohtaisia ohjeistuksia, jotka esittelen tässä luvussa. Ensin kuitenkin tarkastelen heuristista asiantuntija-arviointia käytettävyystudkimuksen menetelmänä. Sitten esittelen luvussa 4.2 Purhon (mt.) heuristiikat ja luvussa 4.3 olemassa olevia käyttöjärjestelmäkohtaisia ohjeistuksia pohtien samalla, mitkä osat näistä heuristiikoista ja ohjeistuksista sopivat oman tutkimukseni tarpeisiin. Lopuksi luvussa 4.4 esittelen omat heuristiikkani.

### 4.1 Heuristinen asiantuntija-arviointi menetelmänä

Heuristinen asiantuntija-arviointi on käytettävyystudkimuksen menetelmä, jossa pieni joukko asiantuntijoita arvioi tuotteen käytettävyyttä tiettyjä heuristiikkoja apuna käyttäen. Heuristiikat ovat erilaisia käytettävyyshauristeita, sääntöjä ja oletuslistoja, jotka kuvaavat yleisellä tasolla käytettävän järjestelmän tai tuotteen piirteitä ja joita voidaan käyttää sekä tuotteen ja käyttöliittymän suunnittelussa että asiantuntija-arvioinnin ohjeistuksina. Asiantuntija-arviointeja voidaan siis tehdä missä tahansa tuotekehityksen vaiheessa. Eniten hyötyä arvioinnista on kehitysvaiheessa, jolloin voidaan välttää käytettävyyshauristeiden päätyminen valmiiseen tuotteeseen, mutta valmiiden tuotteiden arvioinnissa tehtyjä löytöjä voidaan käyttää esimerkiksi uusien tuotteiden suunnittelussa. (Korvenranta 2005, 111–113.)

Heuristisessa asiantuntija-arvioinnissa arvioijat käyvät käyttöliittymän tai muun arvioitavan tuotteen läpi ensin jokainen yksin valittuja heuristiikkoja apuna käyttäen. Tuote suositellaan käymään läpi ainakin kahdesti: ensimmäisellä kerralla arvioija saa yleiskuvan arvioitavasta tuotteesta ja toisella kerralla voidaan keskittyä yksityiskohtiin. Arvioinnin jälkeen arvioijat vertaavat löydöksiään. Arvioijat voivat myös käyttää TAP-menetelmää (*talk aloud protocol*), jolloin kirjuri kirjaa arvioijien löydökset. (Nielsen 1995b.) Löydöksistä kirjataan ylös heuristiikka, jota on rikottu ja se, miten sitä on rikottu. Usein kirjataan myös heuristiikkarikkomuksen vakavuusaste käyttäen esimerkiksi Nielsenin (1993, 103) asteikkoa sekä korjausehdotus. Tässä tutkimuksessa en käytä vakavuusluokitusta, sillä tarkoitukseni ei ole kehittää tarkastelemieni sovellusten dokumentaatiota paremmaksi eikä minulla siksi ole

tarvetta tietää, mitkä rikkomukset täytyy korjata kiireellisesti. Tarkoitukseni on ainoastaan tutkia sitä, millaista dokumentaatio ja sen käytettävyys on.

Vaikka menetelmän nimi onkin asiantuntija-arviointi, ei arvioijien välttämättä aina tarvitse olla käytettävyysasiantuntijoita: termiä käytetään erottamaan heuristisen arvioinnin tekijät tuotteen käyttäjistä (Yáñez Gómez ym. 2014, 1). Arvioijien määräksi suositellaan 3–5:ä, sillä yksi arvioija ei löydä kaikkia käytettävyysongelmia, mutta toisaalta yli viiden arvioijan käyttö ei välttämättä tuo lisäarvoa (Korvenranta 2005, 114).

Tässä tutkimuksessa minulla ei ole mukana muita arvioijia, joten on otettava huomioon, etten tule löytämään aineistosta kaikkia käytettävyysongelmia. Nielsen (1994, Korvenrannan 2005, 112 mukaan)<sup>2</sup> toteaa, että keskimääräinen arvioija havaitsee vain 35 prosenttia ongelmista. Sen sijaan viiden kaksoisasiantuntijan – henkilöiden, joilla on käytettävyysasiantuntijuuden lisäksi osaamista ja tietoa arvioitavasta tuotteesta – ryhmä voi löytää jopa 98 prosenttia käytettävyysongelmista. Uskon kuitenkin pystyväni saamaan tämän tutkielman tarpeisiin riittävän kattavan kuvan aineistoni ongelmista ja onnistumisista.

Eräät tunnetuimmista käytettävyysheuristiikoista julkaisivat Jakob Nielsen ja Rolf Molich vuonna 1990, ja Nielsen (1995a) julkaisi niistä myöhemmin hiotun version. Nämä heuristiikat koskevat erityisesti käyttöliittymien käytettävyyttä, mutta yleisesti ottaen heuristiikkoja voidaan käyttää muidenkin asioiden, kuten dokumentaation käytettävyyden arviointiin. Myös tuotekohtaisia heuristiikkoja voidaan laatia (Nielsen 1995b). Vaikka Nielsenin heuristiikat ovat urauurtavat, en esittele niitä tarkemmin, sillä niitä ei ole suunniteltu dokumentaation arvioimiseen. Seuraavassa luvussa esittelemissäni Purhon (2000) heuristiikoissa voidaan kuitenkin selkeästi nähdä Nielsenin vaikutus.

Heuristisen asiantuntija-arvioinnin vahvuuksia ovat sen nopeus ja edullisuus. Tuloksia voidaan saada jo samana päivänä. Wei-siong Tan, Dahai Liu ja Ram Bishu (2009, 626) mainitsevat, että heuristisessa asiantuntija-arvioinnissa esiin nousevat ongelmat ovat usein yleisluontoisempia, rakenteellisia ongelmia kuin käyttäjätestauksessa löytyvät. Heuristinen arviointi myös todennäköisemmin tuo esiin ongelmien mahdollisia aiheuttajia, jolloin niiden korjaaminen voi olla helpompaa: arvioijat voivat tarkastella järjestelmän toimintoja eri näkökulmista, kun taas

---

<sup>2</sup> Nielsen, Jakob 1994. Heuristic Evaluation. Teoksessa Nielsen, Jakob, & Mack, Robert. L. (toim.), *Usability Inspection Methods*, 25–62. New York, NY: John Wiley & Sons.

käyttäjätestauksessa käytetään yleensä ennalta määrättyjä skenaarioita. Heuristista arviointia voidaan käyttää myös sellaisten tuotteiden arvioimiseen, joita ei ole vielä olemassa: koska arvioijat eivät varsinaisesti käytä tuotetta eli suorita sillä aitoja tehtäviä, jo paperilla olevaa konseptia voidaan arvioida heuristiikkoja käyttäen (Nielsen 1995b). Tan ym. (2009, 626) toteavat, että heuristista asiantuntija-arviointia ja käyttäjätestausta kannattaa käyttää rinnakkain täydentämään toisiaan.

Heuristisen asiantuntija-arvioinnin olennaisin heikkous on se, ettei loppukäyttäjä ole mukana arvioinnissa (Nielsen 1994, Korvenrannan 2005, 121 mukaan) ja asiantuntija-arvioijan käyttäjäkokemus voi erota kohdekäyttäjän kokemuksesta: arvioijan tarkoitusperät ovat erilaiset kuin ”oikean” käyttäjän. Käytettävyys ei myöskään ole objektiivista, vaan käsitys siitä riippuu paljolti arvioijan omista kokemuksista ja mieltymyksistä (Yáñez Gómez ym. 2014, 1). Heuristiikat toimivat kuitenkin hyvänä apuna arvioinnissa, sillä ne perustuvat aiempaan tutkimukseen ja mahdollisesti – kuten tämän tutkimuksen tapauksessa – aiempiin heuristiikkoihin. Tämä ei tietenkään silti kokonaan poista arvioijan subjektiivista kokemusta, mutta vähentää sen kokonaisvaikutusta.

## 4.2 Purhon käytettävyysheuristiikat tekniselle dokumentaatiolle

Teknistä viestintää varten on kehitetty useita heuristiikkalistoja. Esittelin niistä van der Meij’n ja Carrollin (1995) minimalismiheuristiikat luvussa 3.3 ja Yáñez Gómezin ym. (2014) mobiilisovellusten dokumentaatioon liittyviä heuristiikkoja luvussa 3.4. Jotta saisin muodostettua mahdollisimman kattavan pohjan omien heuristiikkojeni laatimiseen, tarkastelen lisäksi tässä luvussa Purhon (2000) teknisen dokumentaation käytettävyysheuristiikkoja. Heuristiikkalista perustuu ajatuksiin siitä, millaista hyvä dokumentaatio on (mt.). Heuristiikkoja on kymmenen:

1. **Dokumentaatio vastaa todellisuutta.** Dokumentaatiossa käytetty kieli on käyttäjälle tuttua ja informaatiojärjestys on luonteva ja looginen.
2. **Dokumentaatio vastaa tuotetta.** Dokumentaatiossa käytetty termistö vastaa käyttöliittymässä olevaa.
3. **Dokumentaatio on tarkoituksenmukaista.** Dokumentaation eri osien käyttötarkoitukset ovat selviä ja dokumentaatio on tarkoituksenmukaisessa muodossa.

4. **Dokumentaatio tukee erilaisia käyttäjiä.** Dokumentaatio tukee eritasoisia käyttäjiä ja käyttäjiä, joilla on erilaisia rooleja. Tietylle käyttäjäroolille tarkoitettua tietoa ei näytetä muille käyttäjille tai se on helposti ohitettavissa.
5. **Dokumentaation informaatorakenne on toimiva.** Tieto on helposti löydettävissä ja ymmärrettävissä. Käyttäjää puhutellaan suoraan. Kuvia, taulukoita ja listoja käytetään tarkoituksenmukaisesti ja ne ovat helposti luettavia.
6. **Dokumentaatio tukee erilaisia tiedonhakutapoja.** Dokumentaation sisällysluettelot ja hakemistot ovat toimivia ja dokumentaatio tukee selaamista: osioiden alut sekä huomiot ja varoitukset ovat helposti löydettävissä.
7. **Dokumentaatio on tehtäväorientoitunutta.** Dokumentaatio rakentuu niiden tehtävien ympärille, joita käyttäjä haluaa saada tehtyä tuotteen avulla.
8. **Dokumentaatio tukee virheistä palautumista.** Dokumentaatioissa on vianetsintäosio, joka antaa käyttäjälle neuvoja yleisiin virhetilanteisiin ja siihen, miten harvinaisempia virheitä voi analysoida.
9. **Dokumentaatio on johdonmukaista ja noudattaa standardeja.** Dokumentaatioissa käytetty sanasto on johdonmukaista. Jos dokumentaatioissa on useampi dokumentti, ne ovat rakenteeltaan johdonmukaisia.
10. **Dokumentaatiolla on lukuohje.** Jos dokumentaatio on laaja, sen käyttöön annetaan ohjeet.

Purhon heuristiikat vastaavat varsin hyvin yleistä käsitystä hyvän dokumentaation piirteistä ja niissä on nähtävissä yhtymäkohtia myös van der Meij'n ja Carrollin (1995) minimalismi-heuristiikkoihin: molemmat esimerkiksi painottavat virheistä palautumista. Purhon heuristiikat on laadittu aikana, jolloin mobiilisovelluksia oli vielä varsin vähän eikä älypuhelimia ollut ollenkaan. Siitä huolimatta näen niiden olevan yleispätevydessään sovellettavissa myös mobiilisovellusten dokumentaatioon.

Purhon (2000) heuristiikoissa on dokumentaation kieleen liittyviä ohjeita: käyttäjää puhutellaan suoraan ja ohjeissa käytetään imperatiivia. Kielen tulisi myös olla käyttäjälle tuttua, ei siis järjestelmäkeskeistä tai jargonia, jollei kohdekäyttäjä ole asiantuntija. Dokumentaation kielen tulisi olla johdonmukaista dokumentoitavan tuotteen kanssa, jolloin esimerkiksi käyttöliittymässä esiintyvät termit kirjoitetaan dokumentaatioissa samalla tavalla kuin ne ovat itse tuotteessa. Mikäli dokumentaatioon kuuluu useampi dokumentti, kielen tulee olla johdonmukaista myös eri dokumenttien välillä. Nämä kaikki pätevät myös mobiilisovellusten käyttäjädokumentaatioon: kielen on oltava käyttäjakeskeistä ja johdonmukaista sekä käyttäjälle

tuttua. Tässä on selkeä yhteys myös Android- ja iOS-käyttöjärjestelmien kehittäjädokumentaatioon (Android Developers 2017; Apple Developers 2017), jotka korostavat samoja piirteitä.

Informaation suunnitteluun liittyvät ohjeistukset Purhon heuristiikoissa ovat myös varsin hyvin sovellettavissa mobiilisovellusten dokumentaatioon, joskin nämä ohjeistukset tuntuvat olevan suunnattu laajemmalle dokumentaatiolle kuin mitä mobiilisovelluksilla yleensä on. Purho (2000) ohjeistaa esittämään informaation järjestyksessä, joka on käyttäjälle luonteva. Dokumentaation eri osien käyttötarkoitusten olisi oltava selviä ja tiedon olisi oltava helposti löydettävissä ja ymmärrettävissä tiedonhakatavasta riippumatta: dokumentaation sisällysluettelot ja hakemistot ovat toimivia ja dokumentaatio tukee selaamista, kun esimerkiksi otsikot, varoitukset ja huomautukset erottuvat selvästi. Kuvat, taulukot ja listat ovat tarkoituksenmukaisia ja helposti luettavissa. (mt.) Mobiilisovellusten käyttäjädokumentaatioissa voidaan hyödyntää hyperlinkkejä, mikä auttaa informaatorakenteen suunnittelussa: dokumentaation alussa voi esimerkiksi olla sisällysluettelo, josta käyttäjä pääsee otsikkoa painamalla tarvitsemansa tiedon äärelle ja dokumentaation eri osien välillä voi olla linkkejä mahdollisesti tarpeelliseen tietoon. Otsikot lienevät tärkein tiedonhakuun tukeva tekijä mobiilisovellusten käyttäjädokumentaatioissa. Myös kuvien merkitys voi korostua, sillä ne voivat auttaa käyttäjää paremmin hahmottamaan ja muistamaan ohjeita, jos tämä joutuu siirtymään dokumentaation ja käsillä olevan tehtävän välillä. Purho (mt.) mainitsee myös dokumentaation lukuohjeiden tarpeellisuuden, mutta uskoakseni mobiilisovellukset vaativat harvoin niin laajaa käyttäjädokumentaatiota, että lukuohje olisi tarpeen.

Purho (2000) myös ohjeistaa tukemaan eri tasoisia käyttäjiä, mutta tämä ei luullakseni ole mobiilisovelluksissa niin olennaista: perustoiminnot esitellään usein ensimmäisellä käyttökerralla, minkä jälkeen kaikki käyttäjät ovat suurin piirtein samalla tasolla. Mobiilisovelluksilla on harvemmin erilaisia käyttäjärooleja, mutta joissakin sovelluksissa voi olla esimerkiksi erilliset roolit ylläpitäjille ja tavallisille käyttäjille: esimerkiksi Slack-projektiviestintäsovelluksessa on erilliset roolit Slack-alustan omistajille, ylläpitäjille, jäsenille ja vierailijoille (Slack 2018). Tällöin eri roolien huomioiminen dokumentaatioissa on ilman muuta tarpeellista. Purho (mt.) mainitsee myös, että tieto, joka koskee vain tiettyä käyttäjäryhmää, ei saisi näkyä muille käyttäjille. Monista mobiilisovelluksista on olemassa sekä ilmais- että niin kutsuttu premium-versio, joista jälkimmäinen on maksullinen ja sisältää enemmän toimintoja tai sisältöä. Ilmaisversion käyttäjälle ei tulisi näyttää maksullisia ominaisuuksia koskevaa dokumentaatiota, sillä tämä voi hämmentää käyttäjää.



Purhon (2000) heuristiikoissa on siis nähtävissä yhteneväisyyksiä van der Meij'n ja Carrollin (1995) minimalismiheuristiikkoihin. Purhon heuristiikat toimivat myös hyvänä siltana minimalismiheuristiikkojen ja mobiilisovellusten käyttöjärjestelmäkohtaisten ohjeistusten välillä: käsittelenkin Android- ja iOS-käyttöjärjestelmien sovelluskehittäjille tarjoamia ohjeita seuraavaksi.

### 4.3 Mobiilisovellusten käyttöjärjestelmäkohtaiset ohjeistukset

Android- ja iOS-käyttöjärjestelmille on olemassa omia ohjeistuksia, jotka määrittävät muun muassa sovelluksissa käytettäviä tyylejä ja painikkeita. Googlella (Android Developers 2017) on Androidille myös erityisesti käyttäjädokumentaatiota koskevia ohjeistuksia. Appllella ei sen sijaan iOS:lle tällaisia erikseen ole, mutta käyttöliittymän ohjeistuksista (Apple Developers 2017) löytyy myös dokumentaatiolle relevantteja ohjeita, jotka koskevat esimerkiksi sitä, millaista kieltä sovelluksissa tulisi käyttää. Tässä alaluvussa käsittelen näitä ohjeistuksia ja pohdin, onko niistä hyötyä heuristiikkojen luomisessa.

Käsittelyn helpottamiseksi olen kerännyt keskeisimmät ohjeistukset alla olevaan taulukkoon (taulukko 1). Koska ohjeistuksia on melko paljon ja ne menevät varsin yksityiskohtaiselle tasolle, olen pyrkinyt nostamaan taulukkoon eräänlaisia teemoja, joita ohjeistuksissa esiintyy, ja esittelen yksityiskohtia tuonempana siinä määrin, kuin ne ovat tarpeen havainnollistamaan ohjeistusten sisältöä. Olen myös taulukossa jakanut ohjeistukset kahteen kategoriaan: dokumentaation ulkoisiin ohjeistuksiin, jotka liittyvät esimerkiksi dokumentaation sijaintiin, ja dokumentaation sisäisiin ohjeistuksiin, jotka liittyvät esimerkiksi tekstin sävyyn ja oikeinkirjoitukseen.

*Taulukko 1. Käyttöjärjestelmäkohtaiset ohjeistukset*

<b>Dokumentaation ulkoiset ohjeistukset</b>		
<b>Ohjeistus</b>	<b>Android</b>	<b>iOS</b>
Käyttäjädokumentaation tulee olla sovelluksen sisällä ja helposti löydettävissä. Dokumentaation tulee löytyä aina samasta paikasta.	x	
Anna käyttäjälle ohjeita pyytämättä ainoastaan näyttääksesi keskeisen toiminnon, jonka käyttämiseen tarvitaan ele.	x	
Älä näytä esittely-slideshow'ta tai vastaavaa ensimmäisellä käyttökerralla.	x	

<b>Dokumentaation sisäiset ohjeistukset</b>		
<b>Ohjeistus</b>	<b>Android</b>	<b>iOS</b>
Kirjoita yleiskielellä, vältä erikoiskieltä ja jargonia.	x	x
Kirjoita lyhyesti ja ytimekkäästi.	x	x
Käytä ystävällistä ja epämuodollista kieltä.	x	x
Puhuttele käyttäjää suoraan, vältä me-pronominin käyttöä.	x	x
Viittaa painikkeisiin ja muihin käyttöliittymäelementteihin johdonmukaisesti.	x	x
Käytä preesensia.	x	
Käytä numeroita numeraalien sijaan. Viittaa päivämääriin ja kellonaikoihin tarkasti (ei ”huomenna”).	x	x
Ota huomioon oikealta vasemmalle luettavat kielet: älä viittaa oikeaan tai vasempaan, sillä käyttöliittymä saattaa joissakin kieliversioissa olla peilattu.	x	
Älä käytä sukupuolitettuja pronomineja: englanniksi ”they” on suositeltavin.	x	

Googlen ja Applen ohjeistuksissa on yhteneväisyyksiä. Ne koskevat enimmäkseen käytettävää kieltä. Molemmat kehottavat käyttämään epämuodollista yleiskieltä ja välttämään jargonia: tekstin sävyn tulisi olla ystävällinen ja sellaisia lyhenteitä kuin *you are* → *you're* voidaan käyttää. Käyttöliittymätekstien lyhyttä ja ytimekkyyttä painotetaan. Yleiskielen käyttö linkittyy Purhon heuristikkaan 1, joka kehottaa käyttämään käyttäjälle tuttua kieltä. Apple (Apple Developers 2017) neuvoo välttämään *me*-pronominia, sillä se voitaisiin tulkita alentuvana, mutta tämä vaikuttaa käytännössä epätodennäköiseltä: *me*-pronominin käytölle ei minusta vaikuttaisi olevan kovin montaa perusteltua käyttöyhteyttä, ja niissä tapauksissa käyttäjä tuskin tekee tällaista tulkintaa. Lähinnä *me*-pronominia esiintyisi tilanteissa, joissa kerrotaan esimerkiksi, että sovelluksen asiakaspalvelu käsittelee käyttäjän lähettämän viestin. Google neuvoo välttämään *me*-pronominia käyttäjakeskeisyyden vuoksi: tulisi keskittyä siihen, mitä käyttäjä haluaa tehdä ja on tekemässä sen sijaan, mitä sovellus tai sen kehittäjä tekee. Pronominien käyttöön liittyy myös Androidin ohjeistus sukupuolitettujen pronomien

välttämisestä. Näissä kieleen liittyvissä ohjeissa on nähtävissä yhteyksiä minimalismiin, joka on perusluonteeltaan käyttäjäkeskeistä (ks. esim. Draper 1998).

Molempien ohjeistuksissa nousee esiin myös painikkeiden kuvaava ja johdonmukainen nimeäminen. Googlella on olemassa lista käytettävistä käyttöliittymäelementtien nimistä, millä pyritään varmistamaan johdonmukaisuus. Apple (mt.) huomauttaa myös, että käytettävien termien tulisi olla kontekstiin sopivia: esimerkiksi kosketusnäyttölaitteella käytetään verbejä *swipe* ja *tap*, joita tietokoneella ei käytettäisi. Näissä molemmissa on yhteneväisyyksiä Purhon heuristiikkaan 2: *Dokumentaatio vastaa tuotetta*.

Google (Android Developers 2017) ohjeistaa sijoittamaan käyttäjädokumentaation aina sovelluksen sisälle. Dokumentaation tulisi aina löytyä samasta paikasta, saman valikon takaa. Kun eri sovellusten käytännöt ovat yhteneväisiä, käyttäjän on aina helppo löytää tarvitsemansa apu. Google myös neuvoa ohjeistamaan käyttäjää ainoastaan silloin, jos käyttäjä sitä ilmaisee tarvitsevansa. Ainoa poikkeus on sellaisten keskeisten toimintojen esittely, joiden käyttämiseen tarvitaan ele. Tällainen voi olla esimerkiksi sovelluskuvakkeen poistaminen puhelimen aloitusnäytöstä: käyttäjä ei välttämättä koskaan itse keksi, miten se tehdään. Järjestelmän laajuisia toimintoja ei kuitenkaan pidä selittää sovelluksessa: puhelimen käyttäjä osaa kyllä vierittää tekstiä, sillä se toimii samalla tavalla joka paikassa.

Googlella on myös tyyliopas (Material Design 2017), jossa käydään varsin yksityiskohtaisesti läpi esimerkiksi oikeinkirjoitusta ja pronominiin käyttöä. Tyylioppaassa on myös ohjeet siitä, missä tilanteessa kutakin välimerkkiä tulee käyttää. Useimmat taulukkoon 1 keräämistäni Android-sovellusten kieltä koskevista ohjeistuksista ovatkin peräisin tästä tyylioppaasta. Kieleen liittyvät ohjeistukset muodostavat ylipäätään suurimman osan käyttöjärjestelmäkohtaisista ohjeistuksista ja ne ovat pitkälti linjassa aiemmin tarkastelemiäni dokumentaatioheuristiikkojen kanssa. Google ja Apple eivät ota kantaa esimerkiksi kuvien käyttöön dokumentaatioissa tai virheiden ehkäisyyn, joten näihin asioihin liittyvät valinnat jäävät dokumentaation kirjoittajan vastuulle.

#### **4.4 Heuristiikat mobiilisovellusten käyttäjädokumentaation arvioimiseen**

Edellä käsiteltyjen heuristiikkojen ja kirjallisuuden pohjalta olen koostanut heuristiikat mobiilisovellusten käyttäjädokumentaation arvioimiseen. Olen ottanut huomioon myös luvuissa 2.2 ja 2.3 käsiteltyjä mobiilisovellusten käyttäjäkokemus- ja käytettävyyssuunnittelun erityispiirteitä. Heuristiikat ovat seuraavat (taulukko 2):

Taulukko 2. Heuristiikat mobiilisovellusten käyttäjädokumentaation arvioimiseen

Numero	Heuristiikka
<b>1.</b>	<b>Dokumentaation löydettävyys</b>
<b>2.</b>	<b>Dokumentaation informaatorakenteen toimivuus</b>
2.1.	Tiedon löydettävyys
2.2.	Tehtävien modulaarisuus
2.3.	Visuaalisten elementtien tarkoituksenmukainen käyttö
<b>3.</b>	<b>Dokumentaation ja tuotteen samanaikaisen käytön helppous</b>
<b>4.</b>	<b>Tehtäväorientaatio</b>
<b>5.</b>	<b>Dokumentaation ja tuotteen vastaavuus</b>
5.1.	Dokumentaatioissa ja käyttöliittymässä käytetyn termistön vastaavuus ja johdonmukaisuus
5.2.	Dokumentaation käyttöliittymän ja tuotteen käyttöliittymän johdonmukaisuus
<b>6.</b>	<b>Virheiden ehkäiseminen</b>
6.1.	Vianetsintäosio, joka antaa käyttäjälle neuvoja yleisiin virhetilanteisiin
6.2.	Varmistuksen kysyminen ennen lopullisten tai mahdollisesti haitallisten toimintojen suorittamista
<b>7.</b>	<b>Dokumentaation kielen tarkoituksenmukaisuus</b>
7.1.	Yleiskielisyys
7.2.	Käyttäjän suora puhuttelu
7.3.	Tekstin ytimekkyys
<b>8.</b>	<b>Käytönaikaisen dokumentaation merkityksellisyys</b>
<b>9.</b>	<b>Käyttöjärjestelmän konventioiden seuraaminen</b>

Heuristiikkojen järjestys perustuu arvioni siitä, miten niitä olisi helpointa käyttää analyysissäni. Järjestys etenee yleisestä yksityiskohtaiseen siinä määrin, kuin näitä heuristiikkoja on mahdollista siten järjestää – esimerkiksi informaatorakenne (heuristiikka 2) koskee dokumentaation kaikkia tasoja, minkä vuoksi olen sijoittanut sen listalla korkealle, vaikka jotkin osat siitä menevätkin yksityiskohtiin. Seuraavaksi perustelen, miksi olen valinnut juuri nämä heuristiikat ja yhdistän niitä edellä käsiteltyihin.

**1. Dokumentaation löydettävyys.** Olen valinnut tämän ensimmäiseksi heuristiikakseni siksi, että jos käyttäjä ei löydä dokumentaatiota tai muusta syystä pääse käsiksi siihen, hän ei voi käyttää sitä: tällöin dokumentaation muilla ansioilla ei ole juuri väliä. Android-

käyttöjärjestelmän ohjeistuksissa (Android Developers 2017) neuvotaan sijoittamaan dokumentaatio aina saman valikkopolun taakse. Dokumentaation olisi hyvä myös olla sovelluksen sisällä, sillä tällöin sen lukeminen ei vaadi verkkoyhteyttä, toisin kuin jos dokumentaatio olisi erillisellä verkkosivulla. Vaikka nykyään nopeat verkkoyhteydet ovat käytettävissä ainakin Suomessa lähes missä tahansa, kaikkialla näin ei ole. Käyttäjän on päästävä dokumentaatioon käsiksi silloin, kun hän sitä tarvitsee: jos dokumentaatio ei ole saatavilla nopeasti, käyttäjä saattaa lopettaa sovelluksen käytön kokonaan (Android Developers 2017).

Android-käyttöjärjestelmän ohjeistuksissa (Android Developers 2017) todetaan, että kaikista paras tapa tehdä dokumentaatiosta löydettävää olisi se, että sovelluksen jokaisesta näkymästä pääsisi suoraan juuri sitä näkymää koskevan ohjeistuksen luokse. Tätä kutsutaan kontekstisensitiiviseksi dokumentaatioksi (*contextual help*). Tiettyyn näkymään liittyvästä ohjeistuksesta tulisi myös päästä helposti käsiksi muuhun dokumentaatioon.

**2. Dokumentaation informaatorakenteen toimivuus.** Heuristiikka 2 koskee dokumentaation informaatorakennetta kahdella eri tasolla: koko dokumentaation tasolla tämä heuristiikka edellyttää, että käyttäjän on helppo löytää dokumentaatiosta tarvitsemansa tieto (alaheuristiikka 2.1) ja yksittäisten tehtävien tasolla modulaarisuutta (alaheuristiikka 2.2) sekä visuaalisten elementtien tarkoituksenmukaista käyttöä (alaheuristiikka 2.3). Yáñez Gómez ym. (2014, 10) nostavat omissa heuristiikoissaan esiin tiedon helpon löydettävyyden, kuten tekevät myös van der Meij ja Carroll (1995, 250) heidän heuristiikoissaan: van der Meij'n ja Carrollin heuristiikka 2.2 koskee muun muassa sitä, kuinka proseduurien otsikot auttavat käyttäjää löytämään tarvitsemansa tiedon. Tiedon löydettävyyteen vaikuttavatkin monet seikat.

Van der Meij'n ja Carrollin (1995, 257) heuristiikoista myös heuristiikka 4.2 liittyy informaatorakenteeseen: se käsittelee dokumentaation lukujen rakennetta. Lukujen tulisi olla itsenäisiä kokonaisuuksia ja oma heuristiikkani 2.2 perustuukin tähän ajatukseen. Kun luvut toimivat yksinään, käyttäjän ei tarvitse etsiä tietoa useista eri paikoista.

Visuaalisia elementtejä käsittelevä heuristiikkani 2.3 sen sijaan perustuu Purhon (2000) heuristiikkaan 5, joka myös nostaa esiin visuaalisten elementtien tarkoituksenmukaisen käytön ja ymmärrettävyyden, ja Yáñez Gómezin ym. (2014, 10) heuristiikkaan 136, *Onko dokumentaation visuaalinen jäsentely hyvin suunniteltu?*. Visuaalisilla elementeillä tarkoitan tässä yhteydessä sellaisia elementtejä, joita käytetään dokumentaatioissa tekstin tukena, kuten

esimerkiksi kuvituksia ja taulukoita. Sen sijaan jätän pois tarkasteluni piiristä esimerkiksi typografian. Visuaalisia elementtejä tulisi käyttää ainoastaan tarpeeseen ja niiden tulisi täydentää tekstiä. Purhon (2000) mukaan ylimääräiset visuaaliset elementit vain hidastavat käyttäjää ja pidentävät esimerkiksi verkkosivujen latausaikoja. Karen Schriver (1997, 412–430) jakaa erilaiset kuvien ja tekstin suhteet viiteen kategoriaan: toisteiseen (*redundant*), täydentävään (*complementary*), tukevaan (*supplementary*), rinnastavaan (*juxtapositioned*) ja pohjustavaan (*stage-setting*). Jokaisella kategorialla on omat käyttötarkoituksensa, ja niiden avulla onkin hyvä analysoida, millaisia tekstin ja kuvan suhteet valitsemissani aineistossa ovat, ja onko kuvien käyttö onnistunutta.

**3. Dokumentaation ja tuotteen samanaikaisen käytön helppous.** Kolmas heuristiikkani koskee tuotteen ja dokumentaation samanaikaista käyttöä: käyttäjän tulisi olla helppo vaihdella dokumentaation ja sovelluksen käytön välillä. Yáñez Gómezin ym. (2014, 10) heuristiikoissa todetaan, että dokumentaatiosta pitäisi olla helppo palata käsillä olevan tehtävän pariin ja että dokumentaatiosta pitäisi päästä palaamaan samaan paikkaan, jossa käyttäjä oli ennen dokumentaation lukemista. Mobiilisovelluksen käyttäjä joutuukin vaihtelevaan sovelluksen ja dokumentaation välillä, jollei dokumentaatio ole käytönaikaista eli sijaitse suoraan sovelluksen käyttöliittymässä. Älypuhelinien näytöt ovat suhteellisen pieniä, joten eri näkymässä olevaa dokumentaatiota ja sovellusta ei voi käyttää yhtä aikaa. Samanaikaista käyttöä vaikeuttaa myös se, että sama sovellus ei voi olla useammassa ikkunassa kerrallaan: vaikka Android 7 - käyttöjärjestelmää käyttävissä puhelimissa on mahdollista käyttää kahta sovellusta yhtä aikaa siten, että näyttö on jaettu kahtia (Bunton 2016), samaa sovellusta ei voi käyttää kahdessa eri näkymässä yhtä aikaa. Lisäksi kaikki sovellukset eivät tue jaetun näytön tilaa.

Koska käyttäjä ei voi lukea dokumentaatiota samalla kun käyttää sovellusta, hänen täytyy painaa dokumentaatiossa mainitut asiat mieleensä, jotta hän muistaisi ne palattuaan takaisin sen tehtävän pariin, jota oli tekemässä. Dokumentaation on siis oltava ytimekästä ja helposti muistettavaa, mikä linkittyy myös muihin heuristiikkoihini: esimerkiksi heuristiikat 2 ja 7.3 ja liittyvät myös muistettavuuteen.

**4. Tehtäväorientaatio.** Neljäs heuristiikkani kuvaa yhtä teknisen viestinnän keskeisistä periaatteista. Sillä on juurensa sekä Purhon (2000) että van der Meij'n ja Carrollin (1995, 248–250) heuristiikoissa. Dokumentaation tulee keskittyä niihin keskeisiin tehtäviin, joita sovelluksella voi suorittaa. Sen määrittäminen, mitkä tehtävät ovat keskeisiä sovelluksen kohdekäyttäjälle, vaatisi käyttäjien tuntemista ja käyttäjätestausta. Tämä ei kuitenkaan oman

tutkimukseni puitteissa ole mahdollista, joten joudun perustamaan päätelmäni itse sovelluksiin: mitkä tehtävät vaikuttavat keskeisimmiltä. Tehtäväorientaatio sisältää myös ajatuksen käyttäjäorientaatiosta: dokumentaatiota kirjoitetaan käyttäjän näkökulmasta. Käyttäjälle tulee myös antaa mahdollisuus ryhtyä heti toimeen ilman pitkäjäisiä johdantoja.

**5. Dokumentaation ja tuotteen vastaavuus.** Viides heuristiikkani koskee dokumentaation johdonmukaisuutta tuotteeseen nähden. Sama heuristiikka löytyy myös Purholta (2000), mutta olen laajentanut ja tarkentanut sen merkitystä: Purho mainitsee lähinnä termistön johdonmukaisuuden, mutta olen itse ottanut mukaan Android-ohjeistuksista ajatuksen siitä, että käyttöliittymäelementteihin viitataan johdonmukaisesti siten, kuin käyttöjärjestelmän konventiona on (Android Developers 2017). Näistä muodostuu alaheuristiikkani 5.1. Alaheuristiikka 5.2 taas tulee Yáñez Gómezin ym. (2014, 10) mobiiliheuristiikkojen dokumentaatiota käsittelevästä alaheuristiikasta 134: *Dokumentaation käyttöliittymä on johdonmukainen tuotteen käyttöliittymän kanssa*. Dokumentaation käyttöliittymän olisi siis hyvä olla visuaalisesti ja toiminnollisesti yhdenmukainen itse sovelluksen käyttöliittymän kanssa.

**6. Virheiden ehkäiseminen.** Viides heuristiikkani keskittyy virheiden ehkäisemiseen ja niistä palautumiseen, jotka ovat keskeinen osa minimalistista dokumentaatiota (van der Meij & Carroll 1995, 250–254). Kuitenkin minimalismista poiketen esitän (alaheuristiikka 6.1), että mobiilisovellusten dokumentaatiossa olisi hyvä olla erillinen vianetsintäosio, joka tarjoaa apua yleisiin ongelmiin. Tämä siksi, että uskon mobiilisovellusten käyttäjien turvautuvan dokumentaatioon yleisesti vasta sitten, kun jotain on jo mennyt pieleen: virhetiedon tarjoaminen dokumentaatiossa olevien tehtävien yhteydessä ei siis auta käyttäjää, joka ei lue käyttöohjeita saadakseen tehtäviä suoritettua. Erillisestä vianetsintäosioista tieto löytyy helpommin. Alaheuristiikka 6.2 koskee varmistuksen kysymistä käyttäjältä, kun ollaan suorittamassa lopullisia tai mahdollisesti vahingollisia toimintoja: esimerkiksi ”Tallentamaton työ menetetään. Haluatko varmasti poistaa?” Voidaan pohtia, ovatko nämä tekstit käyttöliittymätekstejä vai dokumentaatiota. Kallistuisin pitämään niitä käytönaikaisena dokumentaationa, sillä ne ovat ohjeistavia tekstejä, jotka kertovat käyttäjälle riskeistä.

**7. Dokumentaation kielen tarkoituksenmukaisuus.** Tässä heuristiikassa nostan esille dokumentaation kielen. Aiemmin käsittelemissäni heuristiikoissa ei ole kielelle erillistä heuristiikkaa, vaan erilaiset kieleen liittyvät ohjeet ovat hajallaan eri heuristiikkojen osina, kuten esimerkiksi Purhon (2000) heuristiikassa 1, jossa kehoitetaan käyttämään käyttäjälle

tuttua kieltä ja heuristiikassa 2, joka koskee dokumentaatioissa ja käyttöliittymässä käytetyn termistön vastaavuutta. Mielestäni on kuitenkin hyvä keskittää kieleen liittyvät heuristiikat yhteen, sillä se tekee kielen roolin dokumentaatioissa eksplisiittiseksi ja helpottaa analyysia. Purhon lisäksi myös sekä Androidin että iOS:n ohjeistuksissa kehoitetaan käyttämään yleiskieltä ja välttämään jargonia (Android Developers 2017; Apple Developers 2017). Olenkin valinnut alaeuristiikaksi 7.1 *Kieli on käyttäjälle tuttua yleiskieltä*. Yleiskielen käyttö tekee sovelluksista ja niiden dokumentaatiosta helposti ymmärrettäviä mahdollisimman laajalle käyttäjäryhmälle: myös sellaisille käyttäjille, jotka käyttävät englanninkielisiä sovelluksia, vaikka heidän oma äidinkieltensä on jotain muuta.

Myös alaeuristiikka 7.2, *Käyttäjää puhutellaan suoraan*, on yleisesti teknisessä viestinnässä tunnistettu periaate (ks. esim. Lehrner-Mayer 2016) ja sitä painotetaan myös Androidin (Android Developers 2017) ja iOS:n (Apple Developers 2017). Alaeuristiikka 7.3, *Teksti on ytimekästä*, on hieman vaikeammin määriteltävissä, mutta sen voi liittää minimalismin neljänteen periaatteeseen (van der Meij & Carroll 1995, 254–257) ja erityisesti sen alaiseen heuristiikkaan 4.1: dokumentaatioissa kuvattujen tehtävien tulisi olla lyhyitä kokonaisuuksia ja käyttäjälle tulisi antaa ainoastaan välttämätön määrä tietoa. Kuten jo aiemmin totesin, on vaikeaa vetää rajaa, mikä määrä tietoa on tarpeeksi ja mikä on liikaa, mutta pyrin arvioimaan tätä käyttäjän näkökulmasta.

**8. Käytönaikaisen dokumentaation merkityksellisyys.** Käyttäjälle ei esitellä jokaista toimintoa, vaan ainoastaan keskeisimmät ja varsinkin sellaiset, joita käytetään eleillä. Tässäkin rajanveto siitä, mitä kaikkea kannattaa ja tarvitsee esitellä pyytämättä, voi olla vaikeaa. Vaikka Android-sovellusten dokumentaatioon liittyvissä ohjeistuksissa (Android Developers 2017) kielletään antamasta käyttäjälle pyytämättä ohjeita muihin kuin ennestään tuntemattomilla eleillä tehtäviin toimintoihin, mielestäni voi olla hyödyllistä esitellä ensimmäisellä käyttökerralla keskeisiä toimintoja, jolleivät ne ole aivan itsestään selviä.

**9. Käyttöjärjestelmän konventioiden seuraaminen.** Yhdeksäs ja viimeinen heuristiikkani koskee käyttöjärjestelmäkohtaisia konventioita. Sen sisältö siis vaihtelee käyttöjärjestelmästä riippuen. Koska tässä tutkielmassa aineistoni koostuu Android-sovelluksista, tämä heuristiikka sisältää käytännössä Android-sovellusten dokumentaatioon liittyviä ohjeistuksia. Osa ohjeistuksista sisältyy jo muihin valitsemini heuristiikkoihin: esimerkiksi kieleen liittyvät heuristiikat 7.1, 7.2 ja 7.3 ovat linjassa Android-ohjeistusten (Android Developers 2017) kanssa. Sen sijaan omista heuristiikoistani heuristiikka 8 on ristiriidassa Androidin omien



ohjeistusten kanssa, sillä en rajaa käytönaikaista dokumentaatiota pelkästään eleillä suoritettaviin toimintoihin. Yhdeksättä heuristiikkaa sovellan siis niiltä osin, kuin muut heuristiikkani eivät aja sen ohi. Keskeisimpiä Android-sovellusten dokumentaatioon liittyviä ohjeistuksia kokosin taulukkoon 1 (luku 4.3).

On mahdollista, ettei aineistossa ole kaikkiin näihin heuristiikkoihin liittyviä elementtejä, mutta tässä tutkielmassa käsittelemäni teorian ja aiempien heuristiikkojen pohjalta pidän näitä todennäköisinä ja relevantteina mobiilisovellusten käyttäjädokumentaatioissa. Odotukseni on, että mobiilisovellusten käyttäjädokumentaatioissa on laajaa variaatiota, ja eri heuristiikat nousevat vahvemmin esiin eri sovellusten kohdalla. Laadinkin lopuksi näistä heuristiikoista parannellun version, joka aineiston analyysin perusteella toimisi paremmin: heuristiikkoja onkin hyvä kehittää iteratiivisesti, kun huomataan, miltä osin ne eivät sovellukaan esimerkiksi tietyn tyyppisten sovellusten arvioimiseen.

## 5 Aineisto

Aineistokseni olen valinnut kolmen mobiilisovelluksen käyttäjädokumentaation: nämä sovellukset ovat Pocket, Evernote ja Trello. Älypuhelin, jolla tarkastelen näitä sovelluksia, on Samsung Galaxy S7, jonka näyttö on 5,1-tuumainen ja jonka käyttöjärjestelmä on Android 7.0. Aineistoni koostuu siis ainoastaan Android-sovelluksista, sillä iOS-sovellusten ottaminen mukaan tekisi aineistosta liian laajan. Valitsin sovelluksia, jotka ovat suhteellisen suosittuja: Pocket on ladattu Google Play -sovelluskaupasta yli 10 miljoonaa kertaa (Read It Later 2018), Evernote yli 100 miljoonaa kertaa (Evernote Corporation 2018) ja Trello yli viisi miljoonaa kertaa (Trello, Inc 2018). Suosituimmilla Android-sovelluksilla (joista monet, kuten Facebook ja Gmail, tulevat yleensä valmiiksi asennettuina uuden puhelimen mukana) on yli miljardi latausta (Google Play 2018). Päätin heti aluksi, etten valitsisi aineistooni pelisovelluksia, sillä pelien käyttäjäkokemuksen arviointi on erilaista kuin niin kutsuttujen hyötysovellusten: pelien kohdalla puhutaankin pelaajakokemuksesta, jonka arviointiin on omia periaatteitaan (ks. esim. Lucero ym. 2013).

Pocket on vuonna 2007 julkaistu sovellus, joka antaa käyttäjän tallentaa artikkeleita, videoita ja muuta sisältöä internetistä. Näin kaikki tallennettu sisältö löytyy samasta sovelluksesta ja on käytettävissä myös ilman verkkoyhteyttä. (Pocket 2018.) Sovelluksen keskeisin käyttötarkoitus on erilaisten sisältöjen tallentaminen yhteen paikkaan, josta ne ovat helposti löydettävissä ja käytettävissä jopa ilman verkkoyhteyttä. Muut keskeiset tehtävät sijoittuvat tämän päätehtävän alle: ne ovat esimerkiksi sisältöjen hallintaa. Tallennettuihin sisältöihin voi lisätä aihetunnisteita, niitä voi lisätä suosikeiksi ja esimerkiksi luetut artikkelit tai katsotut videot voi siirtää arkistoon. Sovelluksesta voi myös jakaa sisältöjä kavereille, ja muiden käyttäjien suosittelemaa sisältöä voi nähdä seuraamalla heitä. Pocketin käyttäjädokumentaatio on kerätty 8.1.2018 sovelluksen versiosta 6.7.5.1.

Evernote on vuonna 2008 julkaistu muistiinpanosovellus (Libin 2008), jolla voi luoda monenlaisia eri medioita ja keinoja yhdisteleviä muistiinpanoja yksin tai vaikkapa projektiryhmän kesken. Keskeisimpiä käyttötarkoituksia ovatkin muistiinpanojen luominen ja jakaminen sekä erilaisten sisältöjen leikkaaminen ja tallentaminen esimerkiksi internetistä tai pdf-tiedostoista. Evernoten käyttäjädokumentaatio on kerätty 9.2.–22.3.2018 sovelluksen versiosta 7.17.

Trello on vuonna 2011 julkaistu projektinhallintasovellus (Trello 2018), jossa käyttäjä voi luoda eri projekteilleen ”tauluja”. Taulut ovat tiloja, joihin käyttäjä mahdollisine

projektiryhmineen voi luoda teemoiteltuja listoja, joihin taas voidaan lisätä kortteja: näin käyttäjä saa pilkottua projektinsa pienempiin osiin ja jaettua työtä projektiryhmän kesken. Projekti voisi olla vaikkapa tämän viikon askareet, joita varten käyttäjä luo oman taulun: eri listat käyttäjä voisi luoda vaikkapa opiskeluun, töihin ja kotitöihin liittyville askareille. Kortit taas sisältäisivät yksittäisiä askareita tai pieniä kokonaisuuksia – esimerkiksi siivous, jonka käyttäjä voisi kortilla jakaa yhä pienempiin osiin käyttämällä tarkistuslistaa (*checklist*). Trello tarjoaa monenlaisia toimintoja projektienhallintaan: kortteja voi esimerkiksi luokitella kategorioihin, niille voi asettaa määräaikoja ja käyttäjä voi myös tarkastella tietyn taulun käyttäjien viimeisimpiä toimintoja. Trello käyttäjädokumentaatio on kerätty 21.3.2018 sovelluksen versiosta 4.8.0.10622-candidate.

Mobiilisovellusten käyttäjädokumentaation määrän arvioiminen on haastavaa, mutta sitä voi kuvata moduuleina: moduuleina voidaan pitää kaikkia dokumentaation pienimpiä, kokonaisuuden muodostavia osioita, esimerkiksi käytön aikana ilmestyviä vinkkejä tai FAQ-dokumentin yksittäisiä kysymys ja vastaus -kokonaisuuksia. Pocketista keräämäni dokumentaation laajuus on 48 moduulia, Evernotesta 39 moduulia ja Trellosta 51 moduulia. Kaikki dokumentaatio on englanninkielistä ja tarkastelemani sovellukset ovat ilmaisversioita. Kaikista kolmesta sovelluksesta on olemassa myös maksullinen versio, Evernotesta jopa kolme erilaista maksullista käyttäjätasoa. Maksullisten versioiden toiminnoissa ja siten myös dokumentaatioissa voi olla eroja ilmaisversioihin.

Aineiston kerääminen oli yllättävän hankalaa. Suljin tarkasteluni ulkopuolelle pelit, mutta myös sellaiset sovellukset, joita olen aiemmin käyttänyt, sillä liian tutun sovelluksen tarkasteleminen olisi voinut vaikuttaa analyysiini, minkä lisäksi jo aiemmin käyttämäni sovellus ei tietenkään enää näyttäisi minulle sitä dokumentaatiota, joka näkyy vain ensimmäisillä käyttökertoilla. Monessa sovelluksessa, jota harkitsin, olikin dokumentaatiota joko todella vähän tai se oli kokonaan sovelluksen ulkopuolella, esimerkiksi sovelluksen verkkosivuilla. Verkossa sijaitsevan dokumentaation tarkastelu ei olisi tutkimukseni kannalta tarkoituksenmukaista. Pyrin myös välttämään liian samankaltaisia sovelluksia: en esimerkiksi halunnut aineistooni kahta muistiinpanosovellusta. Tästä huolimatta valitsemani sovellukset ovat luonteeltaan jossain määrin samankaltaisia. Ne ovat kaikki eräänlaisia hyötysovelluksia, joiden tarkoitus on auttaa käyttäjää hallitsemaan joko erilaisia sisältöjä tai omia tekemisiään. Erityisesti Evernote ja Trello ovat käyttötarkoituksiltaan varsin yhteneväisiä, vaikka toiminnoissa onkin jonkin verran eroja. On kuitenkin myös sinänsä mielenkiintoista nähdä, millaisia eroja ja yhteneväisyyksiä kahden samankaltaisen sovelluksen käyttäjädokumentaatioissa on.

## 6 Mobiilisovellusten käyttäjädokumentaation heuristinen asiantuntija-arviointi

Tässä luvussa esittelen aineiston analyysini. Esittelen ensin jokaisen sovelluksen analyysin omissa luvuissaan jäsentäen jokaisen luvun niiden teemojen ympärille, jotka kustakin osa-aineistosta nousevat keskeisiksi. Lopuksi vedän löydökseni yhteen ja esitän korjauksia heuristiikkoihini analyysin pohjalta. Koska tutkimusmenetelmäni on heuristinen asiantuntija-arviointi, analyysissäni saattavat korostua dokumentaation käytettävyyteen liittyvät ongelmat, mutta pyrin tuomaan esiin myös sitä, missä tarkastelemani dokumentaatio on onnistunut.

Tässä analyysissä olen kursivoinut käyttöliittymäelementtien ja -näkymien nimet. Käyttöliittymässä olevat muut tekstit, kuten väliotsikot ja viestit käyttäjälle, olen merkinnyt lainausmerkein.

### 6.1 Pocket

Analysoin Pocket-sovelluksen käyttäjädokumentaatiota kolmessa vaiheessa: ensin kävin läpi dokumentaatiosta ottamani kuvankaappaukset, sillä osa dokumentaatiosta ei ole saatavilla enää ensimmäisen käyttökerran jälkeen. Kuvankaappausten perusteella tein huomioita esimerkiksi dokumentaation otsikoista ja siitä, millaista sisältöä dokumentaatioissa ylipäätään on. Seuraavaksi tarkastelin dokumentaatiota itse sovelluksessa muun muassa siltä kannalta, miten dokumentaation ja sovelluksen samanaikainen käyttö toimii. Lopuksi tarkastelin vielä dokumentaatiota lähemmin muun muassa kieleen ja informaatorakenteeseen liittyvien piirteiden kannalta. Löytämäni heuristiikkarikkomukset olen koonnut taulukkoon 3:

*Taulukko 3. Heuristiikkarikkomukset: Pocket*

<b>Käytettävyysongelman kuvaus</b>	<b>Heuristiikka</b>
Dokumentaation otsikoiden muoto vaihtelee, joten on vaikea päätellä, mitä tietoa minkäkin otsikon takaa löytyy.	2.1 Tiedon löydettävyys
Tehtävät on vaikea erottaa muun tekstin seasta. Ei vaihenumerointia.	2.1 Tiedon löydettävyys
Vuorovaikutteisia käyttöliittymäelementtejä on muotoiltu epäjohdonmukaisesti.	2.1 Tiedon löydettävyys
”Then, follow the same steps as above”. Tieto ei ole kovin kaukana, mutta se ei silti löydy ihan samasta paikasta.	2.2 Tehtävien modulaarisuus

Dokumentaatioon ei pääse kaikista näkymistä: tarvittava valikko on näkyvissä ainoastaan kahdessa viidestä päänäkymästä ( <i>My List</i> ja <i>Profile</i> ).	3. Dokumentaation ja tuotteen samanaikainen käyttö
Otsikot ovat useammin toiminto- kuin tehtäväkeskeisiä, esim. ”Pocket Basics”, ”Data and Battery Usage”.	4. Tehtäväorientaatio
Ensimmäisellä käyttökerralla näkyvä <i>This Is Article View</i> -tooltip on tuotokeskeinen: ”Pocket displays this optimized view for articles, and saves them so they are viewable offline.”	4. Tehtäväorientaatio
Sovelluksessa ei ole erillistä vianetsintäosiota.	6.1 Virheiden ehkäiseminen: vianetsintä
Sovellus ei pyydä varmistusta esimerkiksi ennen sisältöjen poistamista.	6.2. Virheiden ehkäiseminen: varmistus
Käyttöliittymätermistöä, jota käyttäjä ei välttämättä ymmärrä. Esim. ”overflow menu”.	7.1 Yleiskielisyys
<i>Tags</i> -näkyvässä kieli on passiivimuotoista: ”Tags are used to describe...” – käyttäjää ei puhutella suoraan.	7.2 Dokumentaation kielen tarkoituksenmukaisuus: käyttäjän puhuttelu
Kun käyttäjä avaa sovelluksen ensimmäisen kerran, hänelle näytetään sovelluksen esittely.	9. Käyttöjärjestelmän konventioiden seuraaminen

Tämän sovelluksen käyttäjädokumentaatiosta tekemäni huomiot voidaan karkeasti jakaa kolmen pääteeman alle: dokumentaation aiheisiin ja asiasisältöön, informaatorakenteeseen ja kieleen liittyviin löytöihin. Seuraavaksi esittelenkin löydöksiäni näistä näkökulmista aloittaen aiheista ja asiasisällöstä.

### 6.1.1 Käyttäjädokumentaation aiheet ja asiasisältö

Tämän osa-aineiston aiheiden ja asiasisällön kannalta keskeisimmät heuristiikat ovat *4. Tehtäväorientaatio*, *6. Virheiden ehkäiseminen* alaheuristiikkoineen ja *8. Käytönaikaisen dokumentaation merkityksellisyys*. Tehtäväorientaatio liittyy dokumentaation aiheiden valintaan, sillä on tärkeää, että käyttäjää ohjeistetaan keskeisimpien tehtävien suorittamisessa. Myös virheiden ehkäisy on itsessään yksi mahdollinen dokumentaation aihe, mutta se voi esiintyä myös esimerkiksi osana yksittäisiä tehtäviä. Käytönaikaisen dokumentaation merkityksellisyys taas liittyy tietenkin erityisesti käytönaikaisen dokumentaation aiheisiin, sillä käytönaikainen dokumentaatio näkyy käyttäjälle riippumatta siitä, tarvitseeko tai haluaako hän ohjeistusta vai ei: tämän dokumentaation on siis oltava sovelluksen käytön kannalta kaikista keskeisintä. Aiheiden ja asiasisällön valinta liittyy myös informaation suunnitteluun ja siten

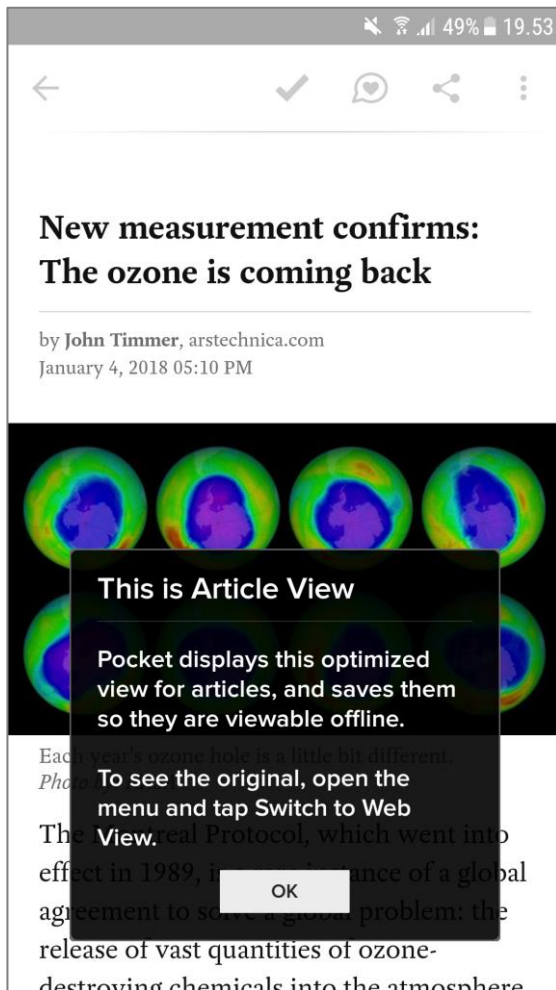
informaatorakenteeseen, joten tässä luvussa tarkastelemillani asioilla on yhteyksiä myös seuraavaan alalukuun, ja nämä menevät osittain limittäin.

Pocket-sovelluksen käyttäjädokumentaatio kattaa keskeisimmät tehtävät varsin hyvin. Sovelluksessa on sekä käytönaikaista että erillistä dokumentaatiota, joka löytyy valikosta *Help*-painikkeen takaa. Käytönaikainen dokumentaatio jakautuu kahteen ryhmään: dokumentaatioon, joka näytetään vain ensimmäisellä näyttökerralla, ja dokumentaatioon, joka näkyy tietyin ehdoin jokaisella käyttökerralla. Kun käyttäjä käynnistää sovelluksen ensimmäistä kertaa, aivan aluksi sovellus näyttää eräänlaisen slideshow'n. Ensimmäisessä näkymässä käyttäjää pyydetään rekisteröitymään tai kirjautumaan. Tämän jälkeiset neljä näkymää ovat eräänlaisia käyttäjädokumentaation ja markkinoivan tekstin risteymiä: ne esittelevät Pocketin perusajatuksen eli sisältöjen tallentamisen, kertovat sovelluksen toimivan kaikilla laitteilla ja minkä tahansa sovelluksen kanssa. Lopuksi todetaan vielä, että Pocket on ”easy on the eyes”, esteettisesti miellyttävä ja helposti luettava. Tällaiset ensimmäisen käyttökerran slideshow't ovat sovelluksissa yleisiä, vaikka Androidin kehittäjäohjeistuksissa (Android Developers 2017) niitä kehoitetaan välttämään. Niiden tarpeellisuudesta voi olla montaa mieltä: yleensä käyttäjällä on jonkinlainen käsitys siitä, mitä sovelluksella tehdään, jos hän on ylipäättään päätenyt lataamaan sen, mutta toisaalta alun tiivis esittely voi myös antaa kattavamman kuvan sovelluksen käyttötarkoituksesta ja -mahdollisuuksista sekä motivoida käyttäjää todella ottamaan sovelluksen käyttöön.

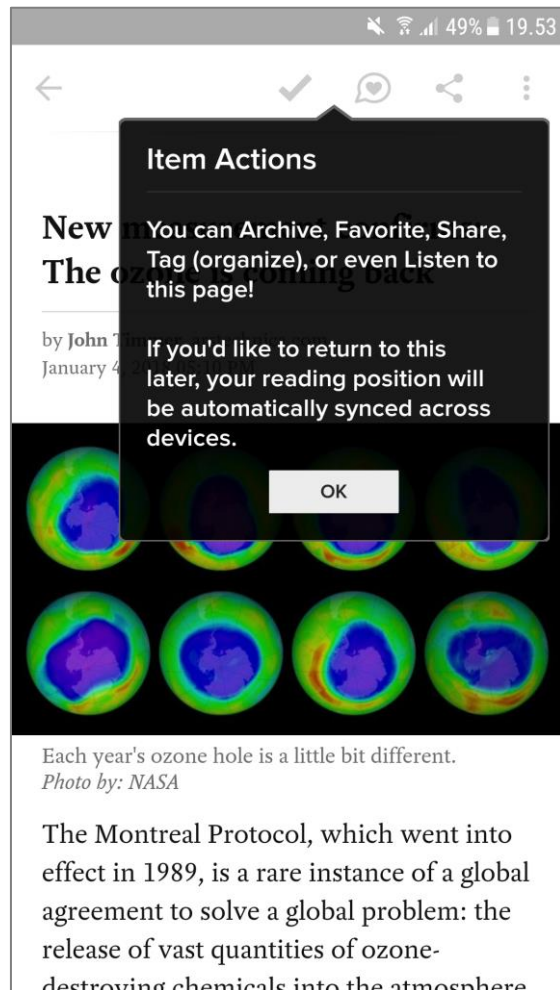
Ensimmäisellä käyttökerralla Pocketissa on myös joitakin vihjetekstejä (*tooltip*), jotka näytetään käyttäjälle automaattisesti ja joista käyttäjä pääsee jatkamaan sovelluksen käyttöä painamalla *OK*. Itselleni näitä tuli vastaan ainoastaan kaksi ja on todennäköistä, ettei niitä enempää olekaan, sillä nämä vihjetekstit ovat tyypillisiä siinä vaiheessa, kun käyttäjä vasta aloittelee sovelluksen käyttöä. Vihjetekstit näkyvät kuvissa 3 ja 4 (seuraava sivu).

*This is Article View* -vihjeteksti (kuva 3) esittelee *Article*-näkyvän käyttötarkoituksen. Tämä selviäisi käyttäjälle varsin nopeasti ilman vihjettäkin, ja herää kysymys, kuinka moni käyttäjä ylipäättään lukee tällaista vihjettä. Käyttäjä saattaa painaa *OK* lukematta vihjettä, jotta pääsisi mahdollisimman nopeasti käyttämään sovellusta. Myös minimalismin periaatteisiin kuuluu, että käyttäjä pääsee tekemisen pariin heti. Vihjetekstissä myös kerrotaan, kuinka käyttäjä pääsee lukemaan tallentamansa sisällön sen alkuperäisestä lähteestä. *Item Actions* -vihjeteksti (kuva 4) luettelee joitakin toimintoja, joilla tallennettuja sisältöjä voi käyttää ja hallita, sekä mainitsee, että sovellus tallentaa kohdan, johon käyttäjä on sisällössä jäänyt ja synkronoi sen

eri laitteiden välillä. Molemmissa vihjeteksteissä annetut vinkit ovat melko keskeisiä. Joillekin käyttäjille ne voivat olla tärkeitä, toiset taas pärjäisivät ilman: niitä on kuitenkin vain kaksi ja ne on helppo ohittaa, jos niitä ei koe tärkeiksi. Kuten useimmilla muillakin tuotteilla, mobiilisovelluksilla on eritasoisia käyttäjiä, ja tässä on menty enemmän ohjeistusta tarvitsevien käyttäjien ehdoilla. Tämä voisi kuitenkin olla myös esimerkki kerrostamisesta, sillä nämä vihjetekstit on hyvin helppo ohittaa halutessaan.

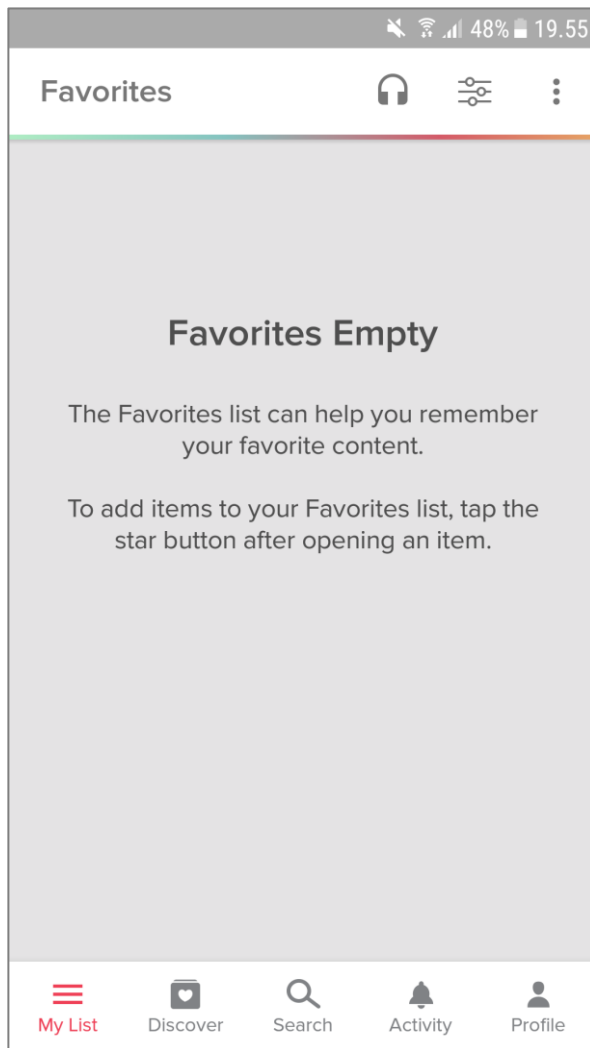


Kuva 3. This is Article View



Kuva 4. Item Actions

Muuta käyttöliittymässä sijaitsevaa dokumentaatiota ovat Pocketissa eri näkymissä olevat esittelevät ja ohjeistavat tekstit. Tällaisia tekstejä on ainakin *Tags*-, *Highlights*-, *Archive*-, *Favorites*-, *Shared to Me*-, *Search*- ja *Profile*-näkymissä. Nämä tekstit näkyvät niin kauan, kun näkymissä ei ole muuta sisältöä, eli esimerkiksi *Favorites*-näkyvän (kuva 5, seuraava sivu) ohjeistavat tekstit näkyvät siihen asti, että käyttäjä lisää jotain sisältöä suosikiksi. Tämä on mielestäni paitsi hyödyllistä käyttäjälle, myös hyvää tilankäyttöä: näkymät olisivat aluksi varsin tyhjiä ja ankeita, jos ei niissä olisi mitään. Saattaisi jopa vaikuttaa siltä, että kyseessä on jokin virhe, eikä sisältö vain näy oikein. Esimerkiksi Apple (Apple Developers 2018) kehottaa



Kuva 5. Tyhjä Favorites-näkymä

Lisäksi dokumentaatiosta voi odottaa löytävänsä hieman harvinaisempiakin aiheita. Pocket-sovelluksen dokumentaation aiheet ovat melko kattavia: *Help*-näkömön etusivulla näkyvät linkit on otsikoitu ”How to Save”, ”Pocket Basics”, ”Read in Pocket”, ”Send to Friend”, ”Favorites, Archive, Bulk Edit”, ”Tags”, ”Sync and Download”, ”Data and Battery Usage”, ”Listen (TTS)”, ”Search Online Support” ja ”Email Us”. Näiden aiheiden järjestystä ja ryhmittelyä käsittelem seuraavassa alaluvussa, sillä se liittyy informaatorakenteeseen.

Pocket-sovelluksen sisällä olevaan käyttäjädokumentaatioon on kerätty keskeisimpiä ohjeistuksen aiheita ja pari sellaista, jotka ovat enemmän lisätietoa kuin ohjeistusta: ”Sync and Download” ja ”Data and Battery Usage”. Nämä kertovat käyttäjälle, miten sovellus synkronoi ja lataa sisältöjä sekä miten sovellus käyttää tiedonsiirtoa ja virtaa. Ne myös antavat käyttäjälle vinkkejä siihen, kuinka sovelluksen toimintaa voi optimoida omalle laitteelle sopivaksi, joten ne sisältävät myös ohjeistavaa tekstiä. Aivan kaikkea dokumentaatiota ei ole kuitenkaan sijoitettu sovelluksen sisälle, vaan lisää dokumentaatiota löytyy Pocketin internetsivuilta

sovelluskehittäjiä käyttämään kaiken tarjolla olevan tilan hyväksi. Nick Babich (2017) huomauttaa Garrettin (2002) tapaan, että käyttäjät käyttävät sovelluksia niiden tarjoaman sisällön vuoksi, ja että siitäkin syystä tällaisia tyhjiä tiloja olisi hyvä käyttää hyödyksi: muodostaa käyttäjälle odotuksia siitä, mitä näkymässä voisi olla, ja kertoa, mitä siinä voi tehdä. Tällaisen dokumentaation voisi myös jollain tapaa nähdä van der Meij’n ja Carrollin (1995, 247) tarkoittamana passiivisena ohjeistuksena, sillä se on kontekstikohtaista ja käyttäjä voi halutessaan jättää sen huomiotta. Passiivinen ohjeistus voi siis joissakin tilanteissa toimia myös mobiilisovelluksissa, vaikka tätä aluksi juuri tilanpuutteesta johtuen epäilin.

*Help*-painikkeen takaa löytyvän dokumentaation tulisi kattaa keskeisimmät tehtävät ja aiheet, joissa käyttäjä saattaa tarvita apua.



”Search Online Support” -linkin takaa. Tämä on aivan järkevää: jos kaikki mahdolliset aiheet löytyisivät sovelluksen sisäisestä dokumentaatiosta, tiedon löytämisestä voisi tulla vaikeaa.

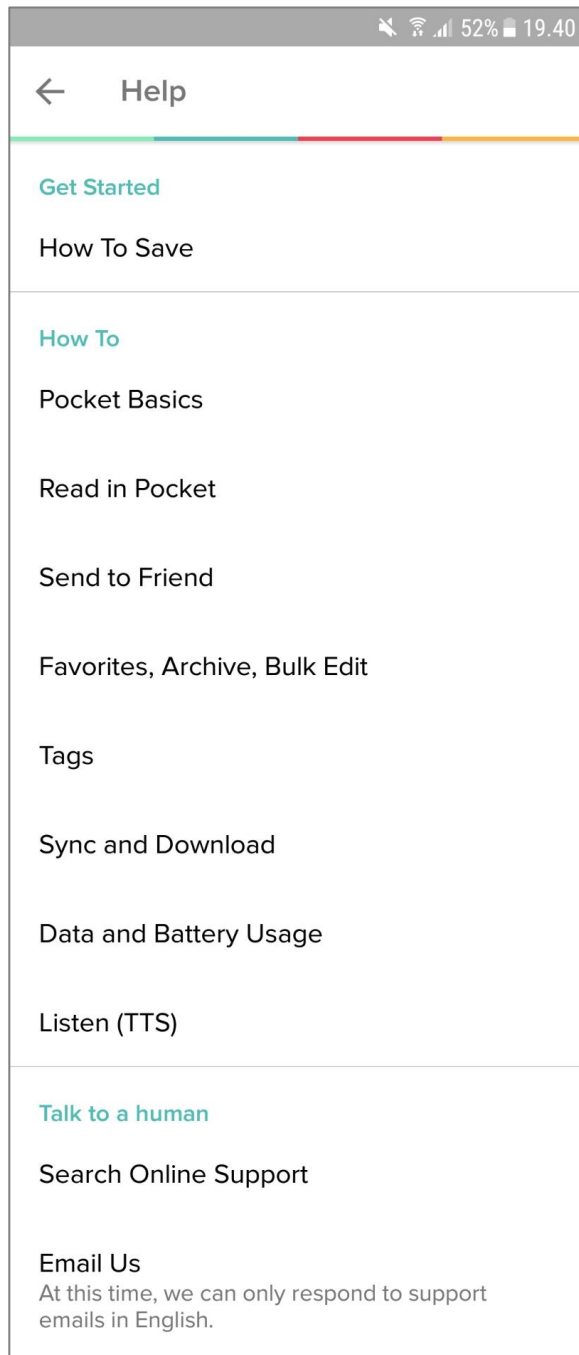
Sovelluksesta ei löydy erillistä vianetsintäosiota, eikä mahdollisista vikatilanteista ylipäättään mainita dokumentaatioissa mitään. Mikäli puhelimesta ei ole verkkoyhteyttä, sovellus ilmoittaa käyttöliittymän eri näkymissä, että esimerkiksi profiilia ei voida ladata. Syytä ongelmaan ei mainita: ”Pocket had trouble loading this profile. Please try again.” Ainoastaan Search-näkymässä kerrotaan ongelman syyksi verkkoyhteyden puuttuminen: ”Not Connected. You must be online to search All Items. You can still search My List while offline.” Ehkä on epätodennäköistä, että sovelluksen käytössä tulisi vastaan muita kuin verkkoyhteyteen liittyviä ongelmia. Mitä tulee virheiden ehkäisyyn, sovellus ei kysy varmistusta esimerkiksi ennen sisältöjen poistamista. Poistamisen jälkeen näytöllä näkyy *Undo*-painike noin viiden sekunnin ajan, minkä jälkeen kumoaminen ei enää ole mahdollista. Virheiden ehkäisy ja korjaaminen voisivat Pocketissa olla paremmalla tolalla.

Pocketin käyttäjädokumentaatiosta aiheet ja asiasisältö on siis valittu varsin onnistuneesti. Käytönaikainen dokumentaatio sijaitsee enimmäkseen tyhjiä käyttöliittymänäkymissä täyttämässä tilaa, mutta tämä on mielestäni aivan järkevä ratkaisu. Ylipäättään käytönaikainen dokumentaatio esittelee muutaman keskeisen asian kahdessa vihjetekstissä ensimmäisellä käyttökerralla, minkä jälkeen nämä eivät enää ole häiritsemässä käyttäjää. Ne on myös halutessaan helppo ohittaa. Virheiden ehkäisyyn ja ratkaisemiseen ei Pocketissa ole juuri panostettu.

### **6.1.2 Käyttäjädokumentaation informaatorakenne**

Pocketin käyttäjädokumentaation informaatorakenteen kannalta keskeisimmät heuristiikkani ovat heuristiikat *2. Dokumentaation informaatorakenteen toimivuus* alaheuristiikkoineen, *3. Dokumentaation ja tuotteen samanaikaisen käytön helppous* ja *4. Tehtäväorientaatio*. Dokumentaation informaatorakenteen toimivuus sisältää tiedon löydettävyyden, tehtävien modulaarisuuden ja visuaalisten elementtien tarkoituksenmukaisen käytön. Toimiva informaatorakenne edesauttaa dokumentaation ja tuotteen samanaikaista käyttöä muun muassa vähentämällä käyttäjän kognitiivista kuormitusta. Tehtäväorientaatio taas vaikuttaa informaatorakenteeseen: millainen informaatorakenne tukee parhaiten tehtäväkeskeisyyttä ja siten myös käyttäjäkeskeisyyttä? Seuraavaksi tarkastelen Pocket-sovelluksen käyttäjädokumentaatiota erityisesti näistä näkökulmista.

Hyvin suunniteltu informaatorakenne auttaa käyttäjää löytämään tarvitsemansa tiedon. Esimerkiksi looginen otsikointi ja aiheiden ryhmittely helpottavat etsintää. Pocketin *Help*-näkyvän (kuva 6) jäsentelyssä on jonkin verran ongelmia: ensinnäkin otsikointi on melko epäloogista. Dokumentaation aiheet on jaettu kolmeen ryhmään, joiden otsikot ovat ”Get Started”, ”How to” ja ”Talk to a Human”.



Kuva 6. *Help*-näkyvä

optimoida sovelluksen toimintaa, ne eivät silti täytä tehtäväkeskeisyyden vaatimusta.

”Talk to a human”-otsikko taas on melkein pä humoristinen, kun heti sen alla onkin otsikko ”Search Online Support”, jonka kautta käyttäjä pääsee selaamaan lisää dokumentaatiota

Ensimmäisen otsikon alla on vain yksi aihe, ”How to Save”. Sisällön tallentaminen onkin Pocketin keskeisin käyttötarkoitus, joten on sinänsä aivan järkevää, että tämä aihe on ensimmäisenä. Mielenkiintoista kuitenkin on, että tämä ainoa ”How to”-alkuinen aihe ei ole ”How to”-osiossa – ja että sen sijaan ”Pocket Basics”, jonka kuvittelisi sopivan hyvin ”Get Started”-otsikon alle, onkin ”How to”-osiossa.

”How to”-otsikon alla on myös varsin sekalainen valikoima dokumentaatiota. Osa sopii tähän osioon hyvin: esimerkiksi ”Read in Pocket”, ”Send to Friend”, ”Listen (TTS)” ja sisältöjen hallintaan liittyvät ”Favorites, Archive, Bulk Edit” ja ”Tags” liittyvät keskeisiin tehtäviin, joita käyttäjä voi sovelluksella suorittaa. Sen sijaan jo aiemmin mainitsemani ”Sync and Download” ja ”Data and Battery Usage” koskevat enemmän sovelluksen itsensä toimintaa kuin käyttäjän mahdollisuuksia. Ne sisältävät tietoa siitä, kuinka sovellus esimerkiksi synkronoi ja lataa sisältöä tai käyttää tiedonsiirtoa ja virtaa: ne ovat hyvin tuotekeskeisiä. Vaikka ne lisäksi sisältävät vinkkejä siitä, kuinka käyttäjä voi

Pocketin verkkosivuilta – tällä ei ole mitään tekemistä ihmiskontaktin kanssa. *Help*-näkyvän rakennetta ja jäsentelyä ei selvästikään ole mietitty loppuun asti.

Android-käyttöjärjestelmän dokumentaatiota koskevissa ohjeistuksissa (Android Developers 2017) kehoitetaan järjestämään dokumentaation aiheet siten, että ensimmäisenä ovat ohjeet, sillä käyttäjä, joka dokumentaatiota katsoo, tarvitsee todennäköisesti ohjeistusta ja haluaa löytää sen nopeasti. Sellaiset lisätiedot kuten tekninen informaatio ja vaikkapa käyttäjäehdot taas voivat olla alempana tai eri valikon takana. Pocketin käyttäjädokumentaatioissa tämä toteutuukin suhteellisen hyvin: *Help*-näkyvässä on lähinnä linkkejä ohjeisiin, joskin myös eräänlaista teknistä tietoa löytyy ”Sync and Download” ja ”Data and Battery Usage”-otsikoiden takaa.

Itse dokumentaation rakenne vaihtelee. Useimpia otsikoita painamalla käyttäjä pääsee suoraan dokumentaatioon. Poikkeuksena on ”How to Save”, jota painamalla käyttäjä pääsee eräänlaiseen välivaiheeseen: tässä näkyvässä käyttäjä valitsee, mistä hän haluaa sisältöä tallentaa. Vaihtoehtoina ovat ”From Your Apps and Browsers”, ”Save Via Email”, ”From Your Computer” ja ”From Your Clipboard”. Kahdessa ensimmäisessä vaihtoehdossa käyttäjälle annetaan pikaohje, esimerkiksi ”Save Via Email”:

Add any page from a Browser or your favorite app to Pocket.

Simply email the link to [add@getpocket.com](mailto:add@getpocket.com).

Lisäksi kahdessa ensimmäisessä on ”Learn How”-painikkeet, joita painamalla käyttäjä pääsee lukemaan yksityiskohtaisempaa ohjetta aiheesta. ”From Your Computer”-otsikon alla ei sen sijaan ole pikaohjetta vaan pikemminkin lyhyt pohjustava teksti sekä ”Get Instructions”-painike, jota painamalla käyttäjä saa ohjeet sähköpostiinsa. ”From Your Clipboard”-otsikon alla on ainoastaan pikaohje – ilmeisesti pidemmän ohjeen kirjoittamista ei ole katsottu tarpeelliseksi. Tätä välimuotoa voisi pitää esimerkkinä kerrostuksesta: joillekin käyttäjille pikaohje voi riittää, ja enemmän tietoa tarvitseva voi painaa ”Learn How”-painiketta. Muissa aiheissa ei tällaista välivaihetta ole, vaikka esimerkiksi ”Pocket Basics”-dokumentin kohdalla se voisi olla järkevää, sillä kyseinen ohjesivu on kaikista pisin.

Kuvia Pocketin käyttäjädokumentaatioissa käytetään melko runsaasti. Suurin osa niistä on käyttötarkoitukseltaan havainnollistavia, mutta muutamalla tuntuu olevan enemmän kuvituksenomainen tarkoitus: ne lähinnä elävöittävät näkymää, jossa ne ovat. Schriverin (1997, ks. luku 4.4) luokituksessa havainnollistavat kuvat sijoittuvat Pocketin tapauksessa enimmäkseen täydentävään tai tukevaan kategoriaan, kun taas kuvituksenomaiset kuvat

sijoittuvat parhaiten tukevaan tai pohjustavaan kategoriaan. Kuvituksenomaiset kuvat luovat kontekstia dokumentaatiolle, johon ne liittyvät: esimerkiksi aiemmin mainitsemissani *How to Save* -näkyvässä jokaisen vaihtoehdoisen tallennustavan yhteydessä on kuva, joka kuvaa sisällön lähdettä ja Pocket-sovellusta, joiden välillä on nuoli. Nämä kuvat voivat paitsi kuvata tallentamisen helppoutta, myös toimia selaamisen tukena, kun käyttäjä näkee kuvastakin, mistä kussakin ohjeessa on kyse.

Havainnollistavat kuvat ohjeteksteissä auttavat käyttäjää hahmottamaan paremmin, kuinka tehtävä suoritetaan. Tämä on erityisen tärkeää mobiilisovellusten käyttäjädokumentaatioissa, sillä kuten todettua, dokumentaatiota ja sovellusta ei yleensä voi käyttää yhtä aikaa. Kuvat sovelluksen käyttöliittymästä auttavat käyttäjää muodostamaan tarkemman mielikuvan tehtävästä ja siten myös muistamaan sen paremmin, kun hän palaa dokumentaatiosta tehtävän pariin. Hyvin valitut kuvat keventävät käyttäjän kognitiivista kuormitusta, mikä parantaa käyttäjäkokemusta: työmuistin rasitus on yksi mobiilisovellusten keskeisimmistä käyttäjäkokemukseen liittyvistä haasteista (Budi 2015). Mielestäni Pocketin dokumentaation kuvitukset toimivatkin hyvin: ne näyttävät käyttäjälle ne käyttöliittymäelementit, joita tehtävän suorittamiseen tarvitaan, ja niiden sijainnin käyttöliittymässä. Näin käyttäjä tietää, mitä hän tarvitsee ja missä se sijaitsee.

Sovelluksen käyttäjädokumentaatioissa on joitakin muotoon liittyviä ongelmia. Kuten yllä olevista esimerkeistä voi huomata, dokumentaation otsikoiden muoto vaihtelee: esimerkiksi ”How to Save”, ”Send to Friend” ja ”Data and Battery Usage” ovat kaikki epäjohdonmukaisia keskenään. Johdonmukaisesti muotoillut otsikot helpottaisivat selaamista, minkä lisäksi otsikko ”Data and Battery Usage” ei oikeastaan kerro, mitä sen takana tarkalleen on – ohjeita vai jotain muuta? Se ei myöskään ole käyttäjä- vaan tuotekeskeinen.

Toinen muotoon liittyvä ongelma on se, ettei sovelluksen käyttäjädokumentaatioissa ole käytetty ollenkaan vaihenumeroitusta, vaikka vaiheita olisikin. Tiettyssä järjestyksessä suoritettavien vaiheiden esittäminen numeroituna listana on teknisessä viestinnässä yleisesti suositeltu käytäntö (ks. esim. McMurrey s. d.). Pocketin dokumentaatioissa ohjeet ovat tavallisena leipätekstinä, ja vaikka kappaleet ovatkin melko lyhyitä, vaiheita on silti hieman vaikea seurata. Sen sijaan numeroitua käytetään yhden kerran ilmaisemaan kahta eri vaihtoehtoa. Pohdin, voisiko syynä numeroinnin pois jättämiselle olla tilan säästäminen. Kun numeroitua ei käytetä, ohjeessa on vähemmän rivejä, jolloin puhelimen näytölle mahtuu

enemmän tietoa kerralla. Mikään Pocketin käyttäjädokumentaation yksittäisistä tehtävistä ei kuitenkaan ole niin monivaiheinen, ettei se numeroitunakin mahtuisi yhdelle ruudulliselle.

Vielä yksi luettavuutta ja tiedon löydettävyyttä hankaloittava tekijä on se, että käyttöliittymäelementtejä on dokumentaatioissa korostettu tekstin muotoilulla epäjohdonmukaisesti. Usein näiden elementtien nimiä ei ole korostettu millään tavalla, mutta välillä ne on kursivoitu. Muotoilussa ei näytä olevan mitään logiikkaa. Yleinen käytäntö teknisessä viestinnässä on lihavoida käyttöliittymäelementtien nimet dokumentaatioissa: esimerkiksi *Microsoft Manual of Style* (Microsoft 2012, 70) kehottaa lihavoimaan käyttöliittymäelementtien nimet.

Dokumentaation modulaarisuudessa Pocket on mielestäni onnistunut pääosin varsin hyvin. Useimmiten tarvittava tieto löytyy samasta paikasta. Ainoa huomaamani ongelma löytyy ”Send to Friend” -ohjeen alta kohdasta ”Adding a Quote”. Ohjeessa neuvotaan, kuinka sisältöä kaverille jakaessa voi viestiin lisätä lainauksen kyseisestä sisällöstä, ja tämän jälkeen kehoitetaan katsomaan varsinaiseen lähettämiseen liittyvät ohjeet yltä: ”Then, follow the same steps as described above.” Ohje on kyllä hyvin lähellä, joten sen löytäminen ei ole vaikeaa, mutta se on myös niin lyhyt, että sen olisi yhtä hyvin voinut toistaa. Jo mainitsemani vaihenumeroinnin puuttuminen ja muotoilujen epäjohdonmukaisuus myös vaikeuttavat ylimääräistä selailua, joten sitä olisi hyvä välttää.

Pocketin käyttäjädokumentaation informaatorakenteessa on siis sekä onnistumisia että ongelmia. Modulaarisuudessa on onnistuttu pääosin hyvin, kun taas esimerkiksi tiedon löydettävyydessä on puutteita esimerkiksi epäjohdonmukaisten otsikoiden ja tekstin muotoilun vuoksi. Tehtäväorientaatioon liittyviä ongelmia ilmenee dokumentaation ajoittaisessa tuotekeskeisyydessä. Sovelluksen ja dokumentaation samanaikaista käyttöä vaikeuttaa se, ettei dokumentaatioon pääse käsiksi kaikista näkymistä, mutta toisaalta dokumentaation onnistunut kuvitus usein helpottaa ohjeiden muistamista.

### **6.1.3 Käyttäjädokumentaation kieli**

Pocketin käyttäjädokumentaation kielen kannalta keskeisin heuristiikka on luonnollisesti heuristiikka 7. *Dokumentaation kielen merkityksellisyys* alaheuristiikkoineen, mutta myös heuristiikka 4. *Tehtäväorientaatio*. Pocketissa dokumentaation kieli on yleiskieltä: helposti lähestyttävää ja neutraalia. Kuitenkin dokumentaatioissa esiintyy myös yks erikoistermi, *overflow menu*, joka viittaa valikkoon, joka yleensä sijaitsee sovellusnäkyvän yläreunassa ja jota Android-sovelluksissa tyypillisesti merkitään kolmella päällekkäisellä pisteellä (ks. esim.

kuva 5): tähän valikkoon sijoitetaan valikkoelementit, jotka eivät mahdu yläreunan valikkonauhaan tai joita kehittäjä ei jostain syystä ole halunnut sijoittaa siihen (Android Developers 2018). Tämä on selkeästi kehittäjäsanastoa, eikä käyttäjälle välttämättä ole selvää, mitä hänen tulee tehdä, kun ohjeessa sanotaan ”Just select *Bulk Edit* in the overflow menu.”

Se, puhutellaanko käyttäjää suoraan vai ei, vaihtelee. Varsinaisissa käyttöohjeissa käyttäjää yleensä puhutellaan imperatiivimuodossa: ”To mark an item as a Favorite, long-press it and tap the star button.” Välillä toimintoja kuitenkin esitellään myös passiivissa: ”Tags are used to describe your saved items and make them easy to be found again using search or while filtering from the tags menu.” Käyttäjän suora puhuttelu on teknisessä viestinnässä yleisesti hyväksytty suositus (ks. luku 4.4), ja se tekee käyttäjälle selväksi, että tämä voi itse tehdä kyseisen asian. Pocketin käyttäjädokumentaatiossa näkökulma on yleisesti ottaen varsin tehtäväkeskeinen, mutta välillä eksytään tuotekeskeisyyden puolelle, esimerkiksi kuvailemaan sitä, mitä sovellus tekee. Osittain tämä voi olla perusteltuakin: esimerkiksi ”Pocket displays this optimized view for articles, and saves them so they are viewable online” korostaa sitä, että Pocket tekee joitakin asioita käyttäjän puolesta ja helpottaa näin käyttäjän elämää.

Pocketin käyttäjädokumentaatiosta löytyy myös suhteellisen paljon pieniä kielivirheitä, kuten virkkeessä ”Many apps like your Browser or Internet app, places the share button in your device’s menu.”: subjekti on monikossa ja predikaatti yksikössä eli virkkeessä on kongruenssivirhe. Nämä eivät kuitenkaan merkittävästi vaikeuta dokumentaation ymmärtämistä. Yleisesti ottaen Pocketin käyttäjädokumentaation kielessä ei ole suurta huomautettavaa, vaan ongelmat ovat melko pieniä.

## 6.2 Evernote

Evernoten kohdalla analyysiprosessini eteni hieman eri tavalla kuin Pocketin analyysissä. Aloitin käymällä läpi kuvankaappaukset saadakseni kokonaiskuvaa dokumentaation informaatorakenteesta ja sisällöstä. Samaan aikaan tarkastelin kuvankaappausten rinnalla myös itse sovellusta, sillä pelkkien kuvankaappausten perusteella oli hieman vaikea saada tarkkaa kuvaa siitä, missä kontekstissa mikäkin osa dokumentaatiosta esiintyy. Myös Evernotessa keskityin lopuksi pienempiin yksityiskohtiin, kuten kieleen.

Analyysiä tehdessäni tuntui aluksi, ettei heuristiikkarikkomuksia tahdo löytyä kuin muutama, ja mietin, eivätkö heuristiikat sovellu käyttäjädokumentaation käytettävyyden arviointiin aivan yhtä sujuvasti kuin Pocketista keräämääni aineistoon. Sitten pohdin, onko kyse kuitenkin siitä,

että Evernote vain noudattaa niitä erityisen hyvin. Lopulta heuristiikkarikkomuksia kyllä löytyi, mutta niiden löytäminen vaati enemmän työtä kuin Pocketin tapauksessa, ja kävinkin tätä osaineistoa lopulta läpi heuristiikka heuristiikalta. Tarkastelen heuristiikkojen soveltuvuuden kysymystä myöhemmin tässä luvussa. Löytämäni heuristiikkarikkomukset näkyvät taulukossa 4:

*Taulukko 4. Heuristiikkarikkomukset: Evernote*

<b>Käytettävyysongelman kuvaus</b>	<b>Heuristiikka</b>
Sovelluksen dokumentaatio ei ole konventionaalisesti <i>Help</i> -valikossa, vaan valikon nimi on <i>Explore Evernote</i> .	1. Dokumentaation löydettävyys 9. Käyttöjärjestelmän konventioiden seuraaminen
Suuri osa käytönaikaisesta dokumentaatiosta näkyy käyttäjälle jokseenkin sattumanvaraisesti.	1. Dokumentaation löydettävyys
Dokumentaation otsikot eivät ole erityisen kuvaavia, joten käyttäjän voi olla vaikea löytää tiettyä ohjeen aihetta.	2.1 Dokumentaation informaatorakenne: tiedon löydettävyys
Dokumentaation ja tietyn ohjeenaiheen löytäminen voi olla vaikeaa. Tiedon löytäminen vaatii useamman siirtymän.	3. Dokumentaation ja tuotteen samanaikaisen käytön helppous
Kaikki otsikot eivät ole tehtäväkeskeisiä.	4. Tehtäväorientaatio
Käyttäjälle ehdotetaan valmista tehtävää: ”Let’s start by making a list of things you need to do tomorrow.”	4. Tehtäväorientaatio
Sovelluksessa ei ole erillistä vianetsintäosiota.	6.1 Virheiden ehkäiseminen: vianetsintä
Kaikki dokumentaatio ei ole varsinaisesti ohjeistavaa, vaan osa on enemmän ideoivaa.	7.3 Tekstin ytimekkyys
Sovelluksen käyttöliittymässä on jopa häiritsevä määrä ohjeistusta/vinkkejä.	8. Käytönaikaisen dokumentaation merkityksellisyys
Käyttäjältä ei kysytä, haluaako hän nähdä ohjeita ja vinkkejä käyttöliittymässä.	9. Käyttöjärjestelmän konventioiden seuraaminen

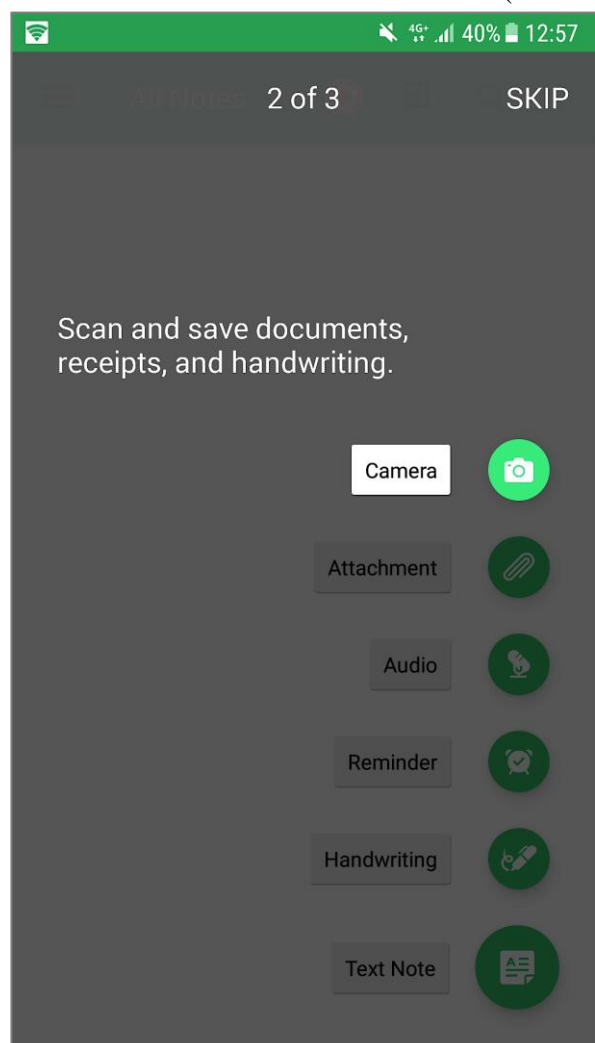
Evernoten käyttäjädokumentaation heuristisesta asiantuntija-arvioinnista nousevat keskeisinä teemoina esiin käytönaikaiseen dokumentaatioon liittyvät kysymykset: mikä määrä dokumentaatiota on tarpeeksi ja mikä on liikaa, miten dokumentaatio on jäsennelty ja miten käytönaikainen dokumentaatio vaikuttaa käyttäjäkokemukseen. Myös informaatorakenne nousee analyysissäni esiin. Evernoten analyysin teemat eivät ole aivan yhtä selkeitä kuin Pocketin, sillä heuristiikkarikkomukset eivät koskeneet niin johdonmukaisesti tiettyjä aihepiirejä. Tämän vuoksi myös tämän analyysin alaluvut ovat laveampia.

## 6.2.1 Käytönaikainen dokumentaatio

Käytönaikaisen dokumentaation tarkasteluun liittyvät erityisesti heuristiikat 3. *Dokumentaation ja tuotteen samanaikaisen käytön helppous* ja 8. *Käytönaikaisen dokumentaation merkityksellisyys*. Valtaosa Evernote-sovelluksen sisältä löytyvästä dokumentaatiosta onkin käytönaikaista dokumentaatiota: se sijaitsee sovelluksen käyttöliittymässä, samassa tilassa sovelluksen toimintojen kanssa. Tällaisella dokumentaatiolla on potentiaalisesti sekä hyötyjä että haittoja. Se vähentää käyttäjän kognitiivista kuormitusta erillään sijaitsevaan dokumentaatioon verrattuna ja sopii käyttäjän ohjeistamiseen ensimmäisellä kerralla, mutta toisaalta se voi myös häiritä käyttäjää (Lakatos 2016). Erityisesti mobiilisovelluksissa sekä hyödyt että haitat korostuvat: kognitiivisen kuorman keventäminen on tärkeä tavoite (Budiu 2015), kun käyttäjä ei voi käyttää erillistä dokumentaatiota ja sovellusta rinnakkain vaan joutuu vaihtelemaan niiden välillä, ja käyttöliittymässä sijaitseva ohjeistus vastaakin tähän haasteeseen hyvin – jos käyttöliittymään mahtuu ohjeistusta. Juuri tilan puute älypuhelimien pienellä näytöllä aiheuttaa haasteita käyttöliittymässä sijaitsevan dokumentaation toteuttamiselle.

Evernotessa käytönaikaista dokumentaatiota onkin paljon – jopa niin paljon, että se saattaa häiritä käyttäjää. Dokumentaation liian suuri määrä rikkoo heuristiikkaa 8. *Käytönaikaisen dokumentaation merkityksellisyys*, sillä käytönaikaiseen dokumentaatioon tulisi valita vain keskeisimmät ohjeen aiheet. Osa Evernoten käytönaikaisesta dokumentaatiosta onkin perusteltua. Kun käyttäjä ensimmäisen kerran avaa sovelluksen, sovellus korostaa +-näppäimen, jolla muistiinpanoja lisätään, ja

sanoo: ”Take better notes. Tap the plus button.” Kun näppäintä painaa, sovellus aloittaa kolmiosaisen esittelyn, joka avautuu käyttöliittymän päälle (ks. kuva 7). Se esittelee sovelluksen keskeisimmät käyttötarkoitukset eli muistiinpanojen tekemisen, erilaisten



Kuva 7. Esittelynäkyvä 2/3



sisältöjen tallentamisen ja erilaiset tavat tehdä muistiinpanoja. Tämä voi olla hyvä tapa antaa käyttäjälle kattavampi mielikuva siitä, mitä kaikkea sovelluksella voi tehdä: muissa tarkastelemissani sovelluksissa samaan tarkoitukseen on käytetty sovelluksen ensimmäisellä käynnistyksellä näytettävää slideshow'ta, jollaisia Androidin ohjeistuksissa (Android Developers 2017) neuvotaan välttämään. Evernoten esittely onkin slideshow'ta parempi ratkaisu sikäli, että se sijaitsee samassa tilassa sovelluksen toimintojen kanssa eli käyttäjä näkee ohjeet niiden todellisessa kontekstissa. Kuitenkin vasta ohjeistusten toisen näkymän kohdalla käyttöliittymään ilmestyy *Skip*-painike, jolla esittelyn voi ohittaa, jos ei koe tarvitsevänsä sitä.

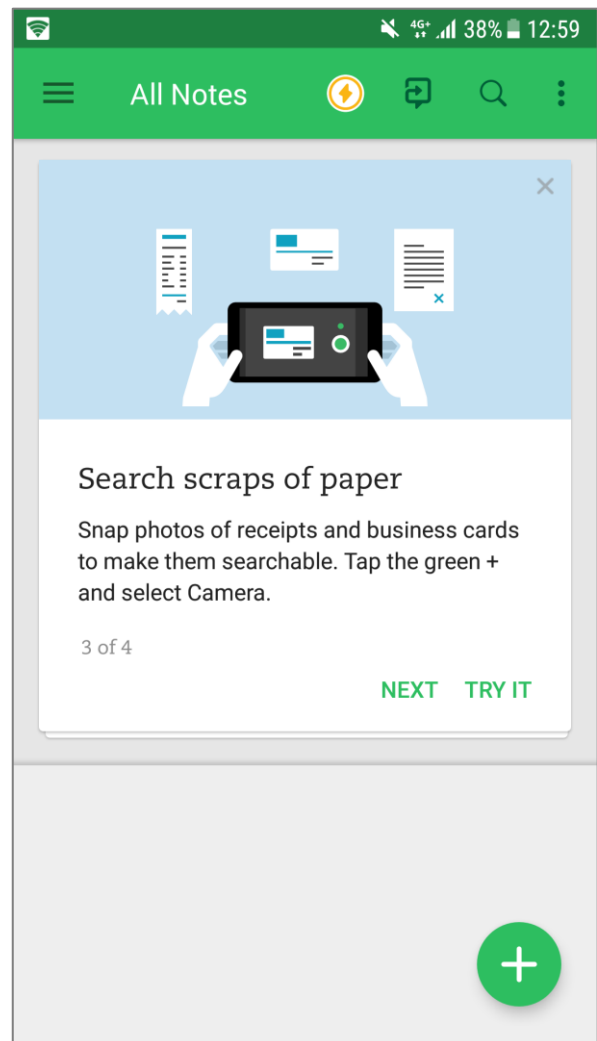
Tämä esittely ei ole ainoa osa dokumentaatiosta, joka avautuu käyttöliittymän päälle siten, että käyttäjä joutuu painamaan sen pois ennen kuin pääsee jatkamaan sitä, mitä oli tekemässä. Kun käyttäjä aloittaa ensimmäisen muistiinpanonsa luomisen, näytölle ilmestyy viestiruutu, joka kehottaa käyttäjää kokeilemaan käsinkirjoitettua tekstiä. Käyttäjän on painettava *Got it!* -painiketta päästäkseen takaisin tehtävän pariin. Sovellus myös näyttää käyttäjälle ”Where Do I Start?”-viestiruudun, joka kertoo, että muistiinpanoja voi tehdä mistä tahansa aiheesta ja kehottaa aluksi tekemään listan asioista, jotka käyttäjän pitäisi tehdä seuraavana päivänä. Tässä on varsin kädestä pitelevä sävy, joka voi ärsyttää käyttäjää: ehkä käyttäjä haluaa jo päästä käyttämään sovellusta siihen, mitä hänen on tarkoitus tehdä, sen sijaan, että kokeilisi harjoituksen vuoksi tehdä listoja, joita ei edes välttämättä tarvitse. Tällainen valmiiden tehtävien antaminen käyttäjälle on vastoin minimalismin ensimmäistä heuristiikkaa, jonka mukaan käyttäjän tulisi antaa ryhtyä toimeen heti (van der Meij & Carroll 1995, 244, ks. luku 3.3). Vaikka käyttäjän ei tässä tapauksessa ole pakko suorittaa ehdotettua tehtävää vaan hän voi jättää ehdotuksen huomiotta, sen lukeminen on kuitenkin yksi ylimääräinen välivaihe käyttäjän ja tekemisen välissä. Minimalismi myös korostaa käyttäjän toiminnan merkityksellisyyttä ja antaa tälle tilaa tehdä omia asioitaan (Farkas 1998, 261), eikä valmiiden, tarkkaan määriteltyjen tehtävien antaminen käyttäjälle sovi tähän ajatukseen.

Suurimman osan Evernoten käyttöliittymässä olevasta dokumentaatiosta muodostavat kuitenkin eräänlaiset vinkit ja vinkkisarjat (kuva 8, seuraava sivu): nämä vinkit ilmestyvät *All Notes* -näkymään, joka on sovelluksen päänäkymä, ja jokainen sarja tarjoaa tiettyyn aihepiiriin liittyviä vinkkejä. Vinkkien aiheita ovat muun muassa sovelluksen keskeiset toiminnot, inspiraatio ja ideointi, yhteistyö ja Evernoten työpöytäversio. Kun käyttäjä on painanut *Done*-painiketta merkiksi siitä, että on lukenut vinkit, sovellus saattaa näyttää seuraavat vinkit heti

perään tai esimerkiksi seuraavalla käyttökerralla. Vinkkien ilmestyminen tuntuu melko sattumanvaraiselta. Toisin kuin käyttöliittymän päälle aukeavat näkymät ja viestiruudut, nämä vinkit eivät välttämättä vaadi käyttäjältä toimenpiteitä, vaan ne on mahdollista jättää huomiotta. Nämä vinkit eivät kuitenkaan kaikki ole ohjeistavia, vaan osa niistä ideoivia tai inspiroivia, esimerkiksi:

*Collaborate with Work Chat. Chat about the things you are working on. Share notes and get feedback without leaving your workspace.*

Jos tätä tarkastelee nimenomaan käyttäjädokumentaationa, se voi tuntua tarpeettomilta: eihän tämä ohjeista käyttäjää mitenkään ja rikkoo selvästi heuristiikkaa 7.3: *Tekstin ytimekkyys*. Silti tällaiset vinkit voivat motivoida käyttäjää antamalla erilaisia näkökulmia eri toimintoihin. Kuvassa 8 näkyy myös esimerkki siitä, kuinka Evernoten käyttäjädokumentaatio rohkaisee käyttäjää kokeilemaan toimintoja itse – aivan kuten van der Meij ja Carroll (1995, 246, ks. luku 3.3) kannustavat tekemään. Kun käyttäjä painaa *TRY IT* -painiketta, sovellus avaa kameranäkymän, jossa käyttäjä pääsee heti kokeilemaan asiakirjojen skannaamista.



Kuva 8. Esimerkki vinkkisarjasta

Evernote myös näyttää käyttäjälle vinkkejä kameranäkymässä. Kameranäkymän alareunassa vuorottelevat vinkit ”Try snapping documents, whiteboards and business cards”, ”Place documents on a contrasting background” ja ”Tilt camera to avoid glare on glossy cards and documents”. Vinkit vaihtuvat sen mukaan, mitä sovellus kameran kautta näkee: jos kamera osoittaa kiiltävään tai hohtavaan pintaan, kuten tietokoneen näyttöön, sovellus neuvoa kallistamaan puhelinta. Jos taas vaikkapa valkoinen kuitti on valkoisen paperin päällä, sovellus kehottaa asettamaan asiakirjan suurempikontrastiselle taustalle. Tämä on mielestäni varsin hyvin toteutettua ja merkityksellistä käytönaikaista dokumentaatiota: sovellus kertoo käyttäjälle heti, miten tämä voisi onnistua skannaamisessa paremmin. Tämä on myös esimerkki

siitä, kuinka mobiilisovelluksissa voidaan hyödyntää puhelimen ominaisuuksia, tässä tapauksessa kameraa, jotta sovelluksen käyttö olisi mahdollisimman helppoa (ks. luku 2.2).

Käyttäjältä ei Evernotessa kysytä, haluaako hän nähdä ohjeita ja vinkkejä, mutta ne saa laitettua pois päältä sovelluksen asetuksista. Käyttäjä siis voi itse säätää dokumentaation määrää, mikä on positiivista, mutta hän ei välttämättä osaa etsiä tätä vaihtoehtoa. Saattaisi siis olla hyvä ensimmäisellä käyttökerralla kysyä, haluaako käyttäjä nähdä ohjeita varsinkin, kun Evernote on toimintaperiaatteiltaan suhteellisen yksinkertainen sovellus. Aiemmin mainitut sovelluksen käyttöliittymässä näytettävät vinkit löytyvät myös sovelluksen *Explore Evernote* -valikosta, josta käyttäjä voi tarkastella niitä myöhemmin.

Evernoten käyttäjädokumentaation käytönaikaisuus helpottaa jossain määrin sovelluksen ja dokumentaation samanaikaista käyttöä (heuristiikka 3): erityiskiitoksen ansaitsevat sovelluksen perustoimintojen esittely näiden toimintojen todellisissa kontekstissa, vaikka nämä toiminnot ovatkin varsin helppotajuisia eivätkä siten välttämättä vaatisi ohjeistusta, ja kameranäkymän vinkit asiakirjojen skannaamiseen. Kameranäkymässä olevat ohjeet ovat erityisen oivaltava lisä, sillä ne myös vaihtuvat sen mukaan, mitä sovellus näkee kameran kautta. Käyttäjä esimerkiksi oppii helposti kokeilemalla, millainen kontrasti asiakirjan ja taustan välillä on riittävä. Sovelluksen ja dokumentaation samanaikaista käyttöä puolestaan vaikeuttavat tietyt dokumentaation informaatorakenteeseen liittyvät seikat, joita käsittelen seuraavassa luvussa.

### **6.2.2 Informaatorakenne**

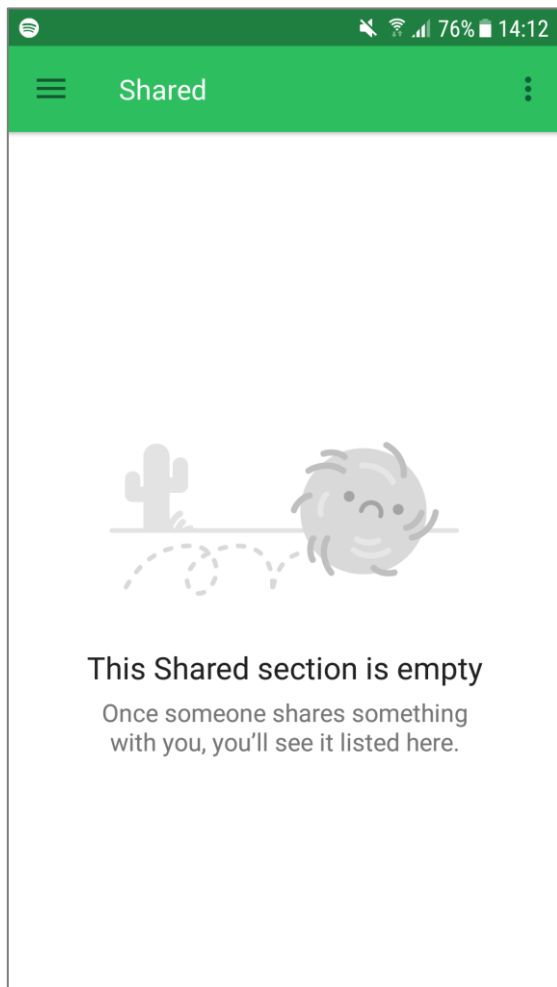
Evernoten käyttäjädokumentaation informaatorakenteeseen liittyy luonnollisesti heuristiikka 2. *Dokumentaation informaatorakenteen toimivuus* alaheuristiikkoineen, mutta sen lisäksi erityisesti heuristiikat 4. *Tehtävöorientaatio* ja 6. *Virheiden ehkäiseminen* alaheuristiikkoineen.

Käytännössä kaikki Evernoten käyttäjädokumentaatio on käytönaikaista, mutta suurin osa siitä löytyy myös koottuna yhteen paikkaan. Tämän voisi ajatella helpottavan tiedon löytämistä, mutta ongelmana on se, ettei käyttäjä välttämättä osaa etsiä sitä: sovelluksessa ei ole *Help*-valikkoa, jota esimerkiksi Android-käyttöjärjestelmän ohjeistukset (Android Developers 2017) suosittelevat, vaan dokumentaatio löytyy *Explore Evernote* -valikosta, kuten juuri aiempana mainitsin. Tämä rikkoo siis paitsi heuristiikkaa 1. *Dokumentaation löydettävyys*, myös heuristiikkaa 9. *Käyttöjärjestelmän konventioiden seuraaminen*. Tästä valikosta löytyvät vinkit, joista osa ei välttämättä tarkalleen ottaen ole dokumentaatiota: ne eivät ohjeista käyttäjää ja ovat enemmän ideoivia kuin tehtäväkeskeisiä. Valikosta löytyvien ohjeiden otsikot ja

kuvaukset eivät myöskään ole kovin kuvaavia tai tehtäväkeskeisiä, joten käyttäjän voi olla vaikea löytää etsimäänsä: otsikoita ovat esimerkiksi ”In a hurry?” ja ”Get out of your inbox”. Tämä taas rikkoo heuristiikkaa 4, joka koskee tehtäväorientaatiota. Tieto on myös usean painalluksen päässä, mikä tekee sovelluksen ja dokumentaation samanaikaisesta käytöstä hankalaa (heuristiikka 3).

Vaikka otsikot ovatkin epämääräisiä, käyttäjän tiedonhakuja voi helpottaa se, että joitakin samoja dokumentaation aiheita löytyy eri otsikoiden alta: esimerkiksi teksti ”Jot down ideas on the go and edit them on your computer later. Tap the green + and select Text” löytyy sekä otsikon ”Ditch the notepad” että otsikon ”Napkins are overrated” alta. Tämä voi tukea erilaisia tiedonhakustrategioita ja tavoitteita, mikä on sinänsä positiivista, mutta pidän tämän vaikutusta kuitenkin tässä tapauksessa vähäisenä.

Evernoten käyttäjädokumentaatioissa on hyvin vähän havainnollistavia kuvia: lähes kaikilla kuvilla on enimmäkseen pohjustava (Schriver 1997, ks. luku 4.4) tarkoitus. Ne eivät auta



käyttäjää sisäistämään ohjetta, vaan toimivat enemmän esteettisessä ja kontekstia luovassa tarkoituksessa (ks. esim. kuva 8, luku 6.2.1). Mahdollisesti ainoa poikkeus löytyy kameranäkymästä, jossa ”Tilt camera to avoid glare on glossy cards and documents”-vinkin yhteydessä on kuva, jossa henkilö kallistaa puhelintaan tietyllä tavalla ottaessaan valokuvaa. Puhelintahan voisi kallistaa myös eri tavalla, mutta tästä kuvasta käyttäjälle selviää oikea tapa. Havainnollistavia kuvia ei kuitenkaan Evernoten tapauksessa välttämättä tarvita, sillä dokumentaatio sijaitsee käyttöliittymässä ja tehtävät ovat varsin yksinkertaisia. Pohjustavilla kuvilla taas on selkeä funktio, sillä ne täyttävät tyhjää tilaa ja hyvin suunniteltuina ja tyylikkäänä parantavat käyttäjäkokemusta (ks. esim. Ismailov 2015).

Kuva 9. Tyhjä Shared -näkyvä

Sen lisäksi, että kuvia käytetään täyttämään tyhjää tilaa, Evernotessa on myös hyvin samankaltaista tyhjän tilan hyödyntämistä kuin Pocketissa: jos jokin näkymä on toistaiseksi tyhjä, siinä saatetaan näyttää dokumentaatiota, joka kertoo, mitä varten näkymä on. Samankaltaisuus näkyy hyvin vertaamalla kuvia 5 (luku 6.1.1) ja 9. Tällä tavalla täytettyä tilaa on miellyttävämpää katsella kuin tyhjää, ja miellyttävyyttä lisää Evernoten hyvin suunniteltu ja hieman humoristinenkin graafinen ilme.

Evernotessa ei ole vianetsintäosiota eikä virhetietoa löydy myöskään ohjeiden yhteydestä. Käyttäjältä kuitenkin kysytään varmistusta esimerkiksi ennen muistiinpanojen poistamista, ja poistamisen jälkeenkin toiminnon kumoaminen on vielä hetken aikaa mahdollista. Sovellus myös ilmoittaa ”Cannot save an empty note”, jos käyttäjä yrittää tallentaa tyhjän muistiinpanon. Näin ollen Evernote rikkoo heuristiikkaa *6.1 Vianetsintäosio*, mutta noudattaa varsin hyvin heuristiikkaa *6.2 Varmistuksen kysyminen*.

### **6.2.3 Heuristiikkojen soveltuvuus**

Kuten mainitsin, minulla oli analyysini alkupuolella epäilyksiä heuristiikkojeni soveltuvuudesta Evernoten käyttäjädokumentaation arvioimiseen. Lähemmällä tarkastelulla kävi ilmi, että asia on varsin monitahoinen: Evernote täyttää osan heuristiikoista varsin hyvin, joitakin ei, ja osaa heuristiikoista tuntuu olevan hankala soveltaa tähän osa-aineistoon. Rikkomuksia olen käsitellyt edellä, ja ne kohdistuvat erityisesti heuristiikkoihin *1. Dokumentaation löydettävyys*, *2.1 Tiedon löydettävyys*, *3. Dokumentaation ja tuotteen samanaikaisen käytön helppous* ja *6.1 Vianetsintäosio, joka antaa käyttäjälle neuvoja yleisiin virhetilanteisiin*, joskin heuristiikkaan 3 liittyen Evernoten käyttäjädokumentaatioissa on myös onnistumisia.

Heuristiikkoja, jotka Evernote täyttää hyvin, ovat lähinnä *6.2 Varmistuksen kysyminen ennen lopullisten tai mahdollisesti haitallisten toimintojen suorittamista*, *7.1 Yleiskielisyys* ja *7.2 Käyttäjän suora puhuttelu*. Evernoten käyttäjädokumentaation kielessä ei juuri ole huomautettavaa: se on selkeää yleiskieltä, ja sovelluksessa käytettävän kielen tyyli on selvästikin huolellisesti mietitty. Kieli on enimmäkseen neutraalia, enkä havainnut oikeinkirjoitusvirheitä. Käyttäjää puhutellaan suoraan erittäin johdonmukaisesti eikä passiivista tai tuotekeskeistä kieltä juuri esiinny. Ainoana kieleen liittyvänä ongelmana voidaan pitää sitä, ettei kieli aina ole näissä heuristiikoissa tarkoittamallani tavalla ytimekästä, vaan mukana on välillä asiaa, joka ei ole käyttäjän ohjaamisen kannalta tarpeellista. Silläkin vaikuttaa kuitenkin olevan oma motivoiva ja inspiroiva tarkoituksensa.

Heuristiikkoja, joiden soveltaminen Evernoten käyttäjädokumentaatioon oli haastavaa, taas olivat erityisesti 2.2 *Tehtävien modulaarisuus* ja 8. *Käytönaikaisen dokumentaation merkityksellisyys*. Syitä sille, miksi osa heuristiikoista ei tuntunut sopivan Evernoteen, ovat muun muassa se, että odotukseni mobiilisovellusten käyttäjädokumentaatiosta oli ilmeisesti kuitenkin jossain määrin konservatiivinen, ja ajatus siitä, ettei käyttöliittymään mahdu dokumentaatiota älypuhelimien pienestä näytöstä johtuen. Heuristiikoissani on selkeästi nähtävissä oletus, että mobiilisovelluksissa on erillinen osio dokumentaatiolle, joka vastaa muodoltaan perinteisempää käyttäjädokumentaatiota: esimerkiksi odotus siitä, että dokumentaatio on selkeästi aiheittain jäsenneiltyä ja selvästi ohjeistavaa ilman lisämausteita. Osittain oletukseni erillisestä dokumentaatiosta johtuikin siitä, etten oletanut mobiilisovellusten käyttöliittymässä olevan tilaa dokumentaatiolle: tällöin dokumentaation olisi pakko sijaita erillisessä näkymässä. Selvästikin ainakin joissain mobiilisovelluksissa ja niiden näkymissä voi olla jopa ylimääräistä tilaa, jota on syytä täyttää jollakin. On myös mahdollista, että mobiilisovellusten käyttö on itsessään sen verran yksinkertaista ja helposti opittavaa, että varsinaisen ohjeistuksen sijaan tai lisäksi käyttäjää voidaan yrittää myös vaikkapa inspiroida tai viihdyttää.

Luvussa 6.4 vedän löydöksiäni yhteen ja esittelen ehdotukseni parannelluista heuristiikoista analyysini pohjalta: Evernoten analyysin perusteella vaikuttaa siltä, että kehitettävää onkin. Ensin kuitenkin esittelen seuraavassa luvussa vielä viimeisen osa-aineistoni analyysin.

### 6.3 Trello

Tästäkin osa-aineistosta kävin aluksi läpi kuvankaappaukset, minkä jälkeen tarkastelin itse sovellusta. Trellon käyttäjädokumentaatio on varsin vuorovaikutteista, minkä takia sitä oli erityisen tärkeää tarkastella sovelluksessa itsessään. Sitten kävin tätä osa-aineistoa läpi vielä heuristiikka heuristiikalta saadakseni kattavan kuvan siitä, kuinka hyvin heuristiikkojen vaatimukset toteutuvat. Olen koonnut löytämäni heuristiikkarikkomukset taulukkoon 5:

*Taulukko 5. Heuristiikkarikkomukset: Trello*

<b>Käytettävyysongelman kuvaus</b>	<b>Heuristiikka</b>
<i>Help!</i> -valikkoon pääsee vain sovelluksen päänäkymästä, ei esimerkiksi yksittäisestä taulunäkymästä.	1. Dokumentaation löydettävyys
Dokumentaatio on kahdessa eri paikassa, joten käyttäjä joutuu myös etsimään eri paikoista.	2.1 Tiedon löydettävyys

Käyttöliittymäelementtien korostus dokumentaatioissa on epäjohdonmukaista.	2.1 Tiedon löydettävyys
Dokumentaation kahteen eri sijaintiin pääsee käsiksi vain sovelluksen päänäköymästä, mikä aiheuttaa ylimääräisiä siirtymiä.	3. Dokumentaation ja tuotteen samanaikaisen käytön helppous
Dokumentaatioissa käytetään eräästä painikkeesta yhdessä paikassa ”Archive Board”-nimitystä, vaikka todellisuudessa se on ”Close Board”.	5.1 Dokumentaatioissa ja käyttöliittymässä käytetyn termistön vastaavuus ja johdonmukaisuus
Käyttäjältä ei kysytä varmistusta peruuttamattomien toimintojen kohdalla (mutta esimerkiksi sisältöjen lopullinen poistaminen sovelluksessa ei ole mahdollista).	6.2 Virheiden ehkäiseminen: varmistuksen kysyminen
”You can archive a card from the overflow menu in the top right”: viitataan, että käyttöliittymäelementti on oikealla.	9. Käyttöjärjestelmän konventioiden seuraaminen
Kun käyttäjä avaa sovelluksen ensimmäisen kerran, hänelle näytetään sovelluksen esittely.	9. Käyttöjärjestelmän konventioiden seuraaminen

Trellon käyttäjädokumentaation analyysistä ei nouse esiin selkeitä teemoja esimerkiksi heuristiikoittain, minkä vuoksi käsittelyn jäsentely oli haastavaa eivätkä alaluvut ole aiheiltaan niin tarkkaan rajattuja kuin Pocketin analyysissä. Analysoidessani tätä osa-aineistoa mielessäni oli sama kysymys heuristiikkojen soveltuvuudesta kuin Evernotenkin kohdalla. Tästä osa-aineistosta keskeisimmiksi teemoiksi nousevat se tapa, jolla dokumentaatio on toteutettu, sekä käyttäjädokumentaation informaatorakenne ja tyylillinen toteutus eli esimerkiksi kieleen ja visuaalisiin elementteihin liittyvät kysymykset.

### 6.3.1 Käyttäjädokumentaation toteutus

Pocketin ja Evernoten käyttäjädokumentaatioon verrattuna Trellon käyttäjädokumentaation toteutus on varsin erilainen, vaikka samankaltaisuuksiakin löytyy. Kuten edellä Evernoten käytönaikaisen dokumentaation analyysissä, keskeisimpiä heuristiikkoja Trellon käyttäjädokumentaation tarkastelun kannalta ovat *3. Dokumentaation ja tuotteen samanaikaisen käytön helppous* ja *8. Käytönaikaisen dokumentaation merkityksellisyys*. Keskeinen kysymys on, mitkä osat Trellon käyttäjädokumentaatiosta voidaan katsoa käytönaikaiseksi dokumentaatioksi.

Kuten Pocket, myös Trello näyttää sovellusta ensimmäistä kertaa käynnistäessä slideshow’n, jossa sovelluksen perusajatus esitellään muutamassa vaiheessa. Tämä on Android-käyttöjärjestelmän suosituksia (Android Developers 2017) vastaan, mutta kuten aiemmin totesin, lyhyt esittely voi antaa käyttäjälle selkeämmän kuvan sovelluksen käyttötarkoituksesta ja motivoida käyttäjää aloittamaan sovelluksen käytön. Myös varsinaista käytönaikaista

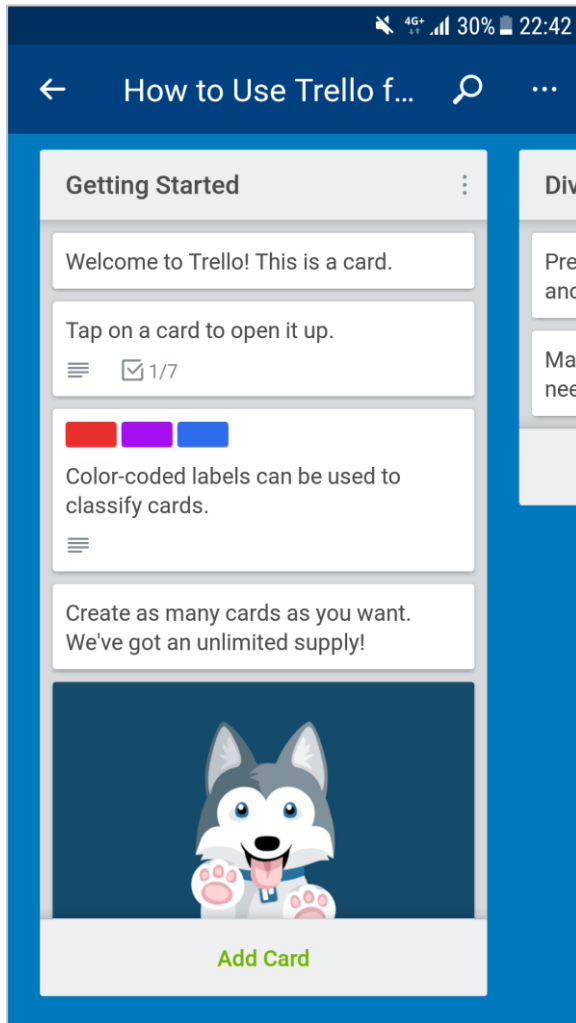
dokumentaatiota on: ensimmäisellä käyttökerralla sovellus ilmoittaa käyttäjäliittymän päälle aukeavassa viestissä, että käyttäjälle on nyt luotu uusi Trello-tili hänen antamallaan sähköpostiosoitteella ja että sähköpostiosoitetta voi käyttää Trellon selainversioon kirjautumiseen. Sovelluksen päänäkymän alareunaan ilmestyy myös viesti, joka pyytää vahvistamaan sähköpostiosoitteen, ja kun tämän on tehnyt, sovellus ilmoittaa, että vahvistus onnistui ja kaikki on nyt valmista sovelluksen käyttöä varten. Babich (2017) toteaaakin, että käyttäjälle kannattaa kertoa, kun tämä on onnistunut tehtävän suorittamisessa, sillä tämä luo käyttäjälle onnistumisen tunteen ja synnyttää siten myös positiivisia tunteita itse sovellusta kohtaan.

Trellon käyttäjädokumentaatio on jaettu kahteen osaan, jotka löytyvät eri paikoista sovelluksen sisältä. Molempiin pääsee sovelluksen päänäkymän kautta: näkymän yläreunassa olevan valikkonauhan valikosta löytyy *Help!*-niminen vaihtoehto, jonka kautta käyttäjä pääsee *Trello Android App* -näkömään, jossa voi muun muassa selata usein kysytyjä kysymyksiä ja vastauksia niihin. Sovelluksen varsinaisessa päänäkymässä käyttäjän silmiin osuu heti *board* eli taulu, jonka nimi on ”How to Use Trello for Android”. Tätä painettuaan käyttäjä pääsee lukemaan ohjeistusta, joka on luotu käyttäen sovelluksen omia työkaluja: ohjeen pääkategoriat ovat listoina, jonka alta yksittäiset ohjeenaiheet löytyvät ”kortteina” (kuvat 10 ja 11, seuraava sivu). Tässä näkymässä käyttäjä voi kokeilla toimintoja itse. Taulut ovat siis eräänlaisia pohjia, joihin kerätään tietyn aiheen ympäriltä listoja, joissa puolestaan on vielä pienempiä osasia eli kortteja. toiminnon suorittamiseen ja voi myös tarjota mahdollisuuden kokeilla kyseistä toimintoa itse. Kuvassa 11 näkyvässä kortissa käyttäjä pääsee kokeilemaan muistiinpanon värikoodausta. Nabeena Mali (2017) toteaa, että kun sovelluksen toimintoja esitellään käyttäjälle siten, että tämä pääsee itse kokeilemaan niitä, käyttäjä paitsi oppii paremmin, myös sitoutuu sovellukseen voimakkaammin kuin jos olisi vain passiivisen lukijan osassa.

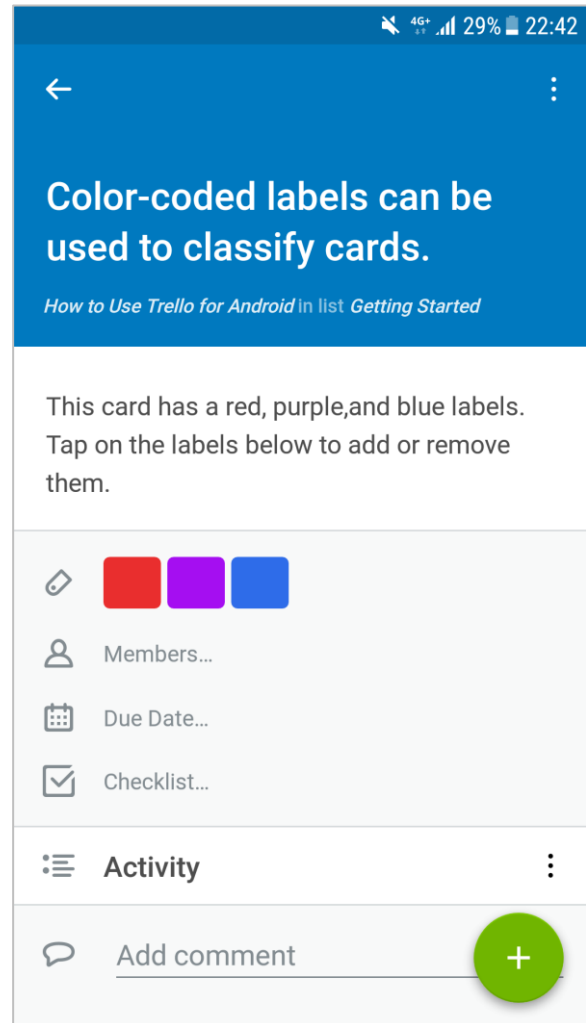
Ovatko ”How to Use Trello for Android”-taululla olevat ohjeet käytönaikaista dokumentaatiota? Ne sijaitsevat kyllä samassa tilassa kuin käyttäjän itse luomat taulut, eivät erillisen valikon takana, mutta ne eivät kuitenkaan ole käytettävissä samaan aikaan, kun käyttäjä esimerkiksi muokkaa projektejaan: käyttäjän on siirryttävä ohjetaululle sovelluksen päänäkymän kautta. Kyseessä ei siis oikeastaan ole käytönaikainen dokumentaatio, vaikka Kuvassa 10 näkyy ”Getting Started”-lista, jonka alla on yksittäisiä ohjeen aiheita: kun jotakin näistä aiheista painaa, se aukeaa kortiksi (kuva 11), joka kertoo aiheesta lisää, antaa ohjeita äkkiseltään olisin niin ajatellut juuri siksi, että dokumentaatio sijaitsee ikään kuin sovelluksen toimintapuolella. Tällainen rajojen hämärtyminen onkin odottamatonta ja kiinnostavaa. *Help!*-painikkeen takaa löytyvä ”Trello Android App”-dokumentaatio sen sijaan ei selvästikään ole



käytönaikaista vaan vastaa enemmän perinteisempää käyttäjädokumentaatiota, joka on erillään sovelluksen toiminnoista: vaikka se onkin toteutettu sovelluksen omia työkaluja käyttäen, käyttäjä voi ainoastaan lukea sitä, ei esimerkiksi muokata. Se sisältää eri kategorioihin jaoteltuja ohjeen aiheita. Myös tässä näkymässä kategoriat ovat listoina, joiden alta esimerkiksi yksittäiset kysymykset löytyvät kortteina, mutta näitä käyttäjä ei voi itse muokata.



Kuva 10. Getting Started -lista



Kuva 11. Color-coded labels -kortti

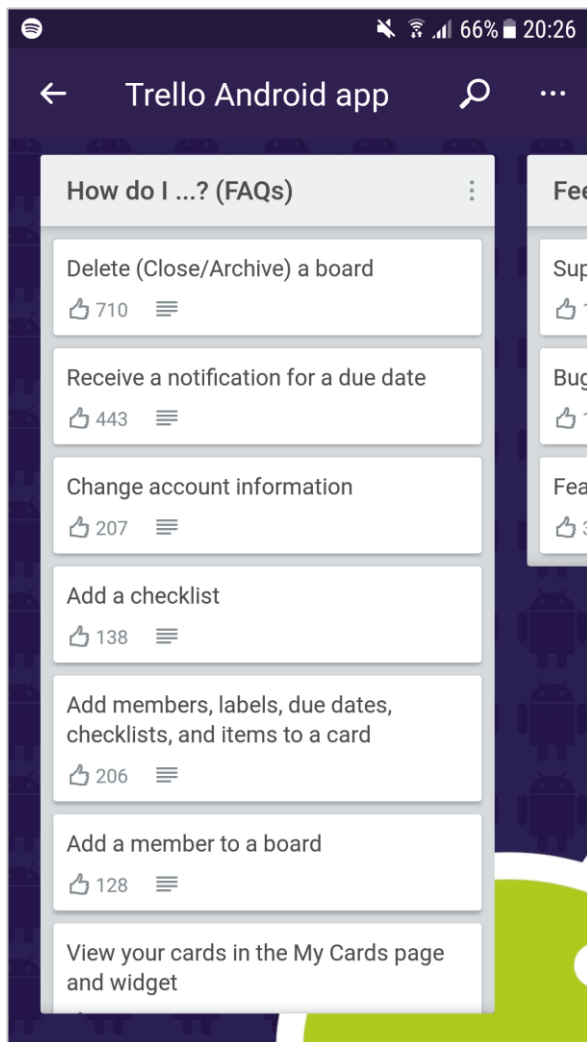
Trellon käyttäjädokumentaatio toteuttaa heuristiikkaani 5.2 *Dokumentaation käyttöliittymän ja tuotteen käyttöliittymän johdonmukaisuus* äärimmäisen hyvin, sillä dokumentaatio on toteutettu käyttäen täsmälleen samoja työkaluja ja toimintoja, joita käyttäjä voi omia projektejaan luodessaan käyttää. Kun käyttäjä pääsee dokumentaatioissa suoraan kokeilemaan erilaisia toimintoja, myös oppiminen ja muistaminen helpottuvat: kun käyttäjä palaa oman tekemisensä pariin, hänellä on käytännön kautta syntynyt mielikuva siitä, mitä hänen pitää tehdä ja miten. Tämä helpottaa dokumentaation ja tuotteen samanaikaista käyttöä. Myös Samuel Hulick (s. d.) toteaa, että ihanteellinen dokumentaatio on erottamaton osa sovelluksen käyttöliittymää.

Toisaalta se, ettei dokumentaatioon pääse käsiksi suoraan esimerkiksi taulu- tai korttinäkymästä, jossa käyttäjä omia asioitaan tekee, puolestaan vaikeuttaa samanaikaista käyttöä. Heuristiikka 3. *Dokumentaation ja tuotteen samanaikaisen käytön helppous* toteutuu siis osittain, mutta siirtymiä tulee tarpeettoman monta, vaikka teknisesti olisi aivan mahdollista sijoittaa kaikkien näkymien ylävalikkoon linkki dokumentaatioon.

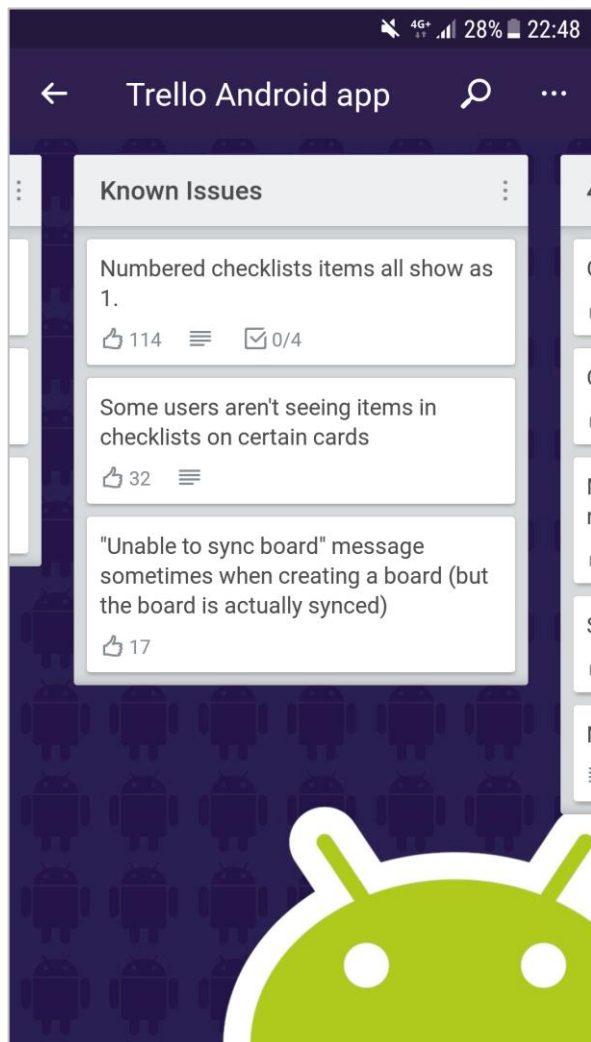
### 6.3.2 Informaatorakenne ja kieli

Trellon käyttäjädokumentaation informaatorakenteeseen liittyvät erityisesti heuristiikat 2. *Dokumentaation informaatorakenteen toimivuus* alaheuristiikkoineen, 4. *Tehtävörientaatio* ja 6. *Virheiden ehkäisy* alaheuristiikkoineen. Trellon käyttäjädokumentaation informaatorakenteessa on sekä ongelmia että onnistumisia. Dokumentaation löydettävyyden ja tiedon löytämisen kannalta on ongelmallista, että dokumentaatio sijaitsee kahdessa eri paikassa. Mistä käyttäjä tietää, mistä mihinkin kysymykseen pitäisi etsiä vastausta? Joitakin asioita tosin käsitellään kyllä molemmissa paikoissa. Vaikka dokumentaation jakautuminen eri paikkoihin voikin olla käyttäjälle hankalaa, se voi mahdollisesti myös tukea erilaisia tiedonhakatapoja: joku voi pitää toimintojen kokeilemisesta, kun taas joku toinen saattaa mieluummin vain lukea ohjetta.

Myös otsikoissa on epätasaisuutta. ”How to Use Trello for Android”-ohjetaulun listojen otsikot ovat hieman epämääräisiä: ”Getting Started”, ”Diving In”, ”Mastering Trello” ja ”More Info”. Kolme ensimmäistä viestittävät etenemistä ja taitojen syvenemistä, mutta ne eivät välttämättä kerro käyttäjälle, missä juuri hänen tarvitsemansa tieto sijaitsee. Korttien otsikot ovat vaihtelevia (ks. kuva 10, luku 6.3.1), ja osasta korteista löytyy niiden avaamisen jälkeen tarkempia ohjeita, kun osassa taas on vain otsikko. Sen sijaan ”Trello Android App”-dokumentaatioissa listojen ja korttien otsikot ovat huomattavasti kuvaavampia. Listojen otsikot ovat ”How do I ...? (FAQs)” (*Frequently Asked Questions* eli usein kysytyt kysymykset, ks. kuva 12, seuraava sivu), ”Feedback” ja ”Known Issues” (kuva 13, seuraava sivu), minkä lisäksi täältä löytyvät myös sovelluksen eri versioiden *release notes* -ilmoitukset, mutta ne olen rajannut tarkasteluni ulkopuolelle. ”How do I ...? (FAQs)”-listalta löytyvät kortit jatkavat otsikon kysymystä, esimerkiksi ”Add a checklist”. Ne ovat hyvin tehtäväkeskeisiä ja viestittävät selkeästi, mitä niiden takana on: tätä koskevat heuristiikat 4. *Tehtävörientaatio* ja 2.1. *Tiedon löydettävyys*.



Kuva 12. ”How do I ...? (FAQs)”-aiheita



Kuva 13. ”Known Issues”-aiheet

Heuristiikka 2.1 koskee tiedon löytämistä yksittäisten tehtävien sisällä. Useimmissa ”How do I ...? (FAQs)”-listalla olevissa korteissa käytetään vaihenumerointia, mikä helpottaa ohjeiden seuraamista. Kuitenkin joistakin ohjeista numerointi puuttuu ja teksti on kappalemuodossa, minkä lisäksi yhdessä ohjeessa käytetään numeroinnin sijaan luettelomerkkejä. Käyttöliittymäelementit on usein korostettu dokumentaatioissa kursiiivilla, mutta toisinaan niitä ei ole korostettu mitenkään, ja tämä epäjohdonmukaisuus voi hankaloittaa tiedon löytämistä: tilanne on samankaltainen kuin Pocketin dokumentaatioissa (luku 6.1.2).

Kuten Evernotessa, myös Trellossa tehtävien modulaarisuutta (heuristiikka 2.2) on osittain vaikea arvioida. ”Trello Android App”-dokumentaatioissa modulaarisuus toteutuu hyvin, sillä yksittäisissä korteissa olevat ohjeet tarjoavat yksinään käyttäjälle kaiken tiedon, jota tämä tehtävän suorittamiseen tarvitsee. Sen sijaan ”How to Use Trello for Android”-ohjetaululla olevien ohjeiden modulaarisuuden arvioiminen on hankalampaa, sillä yksittäiset kortit saattavat olla niin pieniä palasia, etteivät ne ole varsinaisia tehtäviä ollenkaan (ks. kuva 10, luku 6.3.1).

Ehkä tässä moduuleja ovatkin kokonaiset listat, mutta niitäkin on vaikea arvioida kokonaisuuksina, kun ne muodostuvat pienistä paloista, joiden välillä täytyy erikseen siirtyä.

Ohjeisiin liittyviä kuvia Trellossa ei juurikaan ole, mutta ne eivät nähdäkseni ole tarpeenkaan, kun käyttäjällä on apunaan sovelluksen varsinainen käyttöliittymä toimintoineen. Kuvitus kuvat ovat leikkisiä ja sopivat sovelluksen tyyliin: esimerkiksi kuvaliitteiden lisäämiseen ohjeistavassa kortissa on kuva Trelon maskottikoira Tacosta. Heuristiikka 2.3. *Visuaalisten elementtien tarkoituksenmukainen käyttö* ei siis varsinaisesti koske tätä osa-aineistoa. Myöskään varsinaista vianetsintäosiota (heuristiikka 6.1.) Trellostakaan ei löydy eikä virhetietoa näy myöskään tehtäväkohtaisten ohjeiden yhteydessä, mutta ”Known Issues”-listalla näkyvät tekniset ongelmat, jotka ovat tällä hetkellä ratkaisematta ja kehittäjien tiedossa. Kahteen kolmesta aineistonkeruuhetkellä tiedossa olevasta ongelmasta tarjotaan väliaikaisratkaisua, joista toisen pitäisi toimia ja toinen on toiminut joillakin käyttäjillä. Teknisistä vioista johtuviin virhetilanteisiin on siis tarjolla jonkin verran apua ja ainakin käyttäjälle kerrotaan, että viat eivät johdu käyttäjästä itsestään ja niitä ollaan korjaamassa.

Trelon käyttäjädokumentaation tyylliseikkoja ovat vähäisen kuvituksen lisäksi myös kieleen liittyvät asiat. Trelon dokumentaatio on yleiskieltä, mikä vastaa heuristiikkaa 7.1. *Yleiskielisyys*. Ajoittain Trelon kieli on puhekielistä ja leikkisääkin. Kun sovelluksen ilmaisversiota käyttävä käyttäjä aktivoi jollekin taululle jonkin tarjolla olevista *Power-Up*-lisäosista, sovellus ilmoittaa:

*Darn! It looks like you've reached your Power-Ups limit for this board.*

*To unlock more, upgrade your team to Business Class today!*

Samasta esimerkistä on huomattavissa myös markkinoivaa sävyä. Käyttäjää myös puhutellaan johdonmukaisesti suoraan niin ohjeteksteissä kuin tämän kaltaisissa ilmoituksissakin, mikä puolestaan vastaa heuristiikkaa 7.2. *Käyttäjän suora puhuttelu*. Välillä Trelon kehittäjätiimi viittaa myös itseensä: ”Make as many lists and boards as you need. We'll make more!” Tällä luodaan käyttäjälle mielikuvaa siitä, että sovelluksen on tarkoitus helpottaa hänen työtään tai arkeaan, ehkä jopa jonkinlaisesta yhteistyöhengestä. Käyttöjärjestelmien omissa ohjeistuksissa (Android Developers 2017; Apple Developers 2017) me-pronominin käyttöön suhtaudutaan varauksella, mutta Androidin ohjeistuksissa (mt.) mainitaan tämän olevan hyväksyttävää, jos ilmaistaan sovelluksen kehittäjien tai esimerkiksi asiakastuen tekevän joltain käyttäjän puolesta. Tässä on tavallaan kyse tällaisesta tilanteesta, joskin sävy on leikkisä: digitaalisia tauluja ei

tarvitse erikseen tehdä lisää. Trellossa käytetty kieli on ilmeisen hyvin mietittyä ja huoliteltua, sillä kieli- tai kirjoitusvirheitä en löytänyt.

Sovelluksen ja dokumentaation termistö (heuristiikka 5.1.) on Trellossa käyttäjädokumentaatioissa hyvin johdonmukaista yhtä poikkeusta lukuun ottamatta: ohjeissa puhutaan valikkopainikkeesta nimeltä *Archive Board*, kun todellisuudessa painike on *Close Board*. Trellossa käyttäjädokumentaatioissa myös viitataan ainakin kahdesti tiettyyn suuntaan käyttöliittymässä: ”You can archive a card from the overflow menu in the top right” ja ”Hold the top right of a list to drag it around”. Android-käyttöjärjestelmän tyylioppaassa (Material Design 2017) kehoitetaan välttämään viittauksia vasempaan tai oikeaan, sillä suunta saattaa muuttua lokalisaatiossa: oikealta vasemmalle luettavia kieliversioita varten käyttöliittymä saatetaan peilata. Jälkimmäisessä esimerkkilauseessa suunnan määre olisi kuitenkin melko välttämätön, jos listoja voisi raahata vain oikeasta yläkulmasta painamalla. Näin ei kuitenkaan ole, joten suunnan voisi jättää mainitsematta ja puhua esimerkiksi listan yläreunasta. Tällaisenaan dokumentaatio rikkoo heuristiikkaa 9. *Käyttöjärjestelmän konventioiden noudattaminen*.

Kuten Evernoten, myös Trellossa käyttäjädokumentaatioissa on piirteitä, joihin en heuristiikoissani osannut varautua. Osa-aineistojeni analyysi tuntuikin etenevän siten, että ensimmäinen osa-aineisto, Pocket, vastasi varsin hyvin odotuksiani, kun taas seuraavana vuorossa oleva Evernote oli jo hieman kauempana siitä, mitä heuristiikkoja laatiessani odotin, ja Trello yllätti erikoisella toteutuksellaan. Pocketin kohdalla löydökset ovat helposti jaoteltavissa muutaman selkeän teeman alle, mutta Evernoten ja varsinkin Trellossa analyysiä on vaikeampi jäsentää. Seuraavaksi vedän löydökseni yhteen ja esittelen analyysin pohjalta tekemäni korjaukset heuristiikkoihini.

## **6.4 Yhteenveto**

Kuten juuri edellä totesin, heuristiikkojen toimivuus vaihteli eri osa-aineistojen välillä. Pocketin käyttäjädokumentaatioissa sekä ongelmat että onnistumiset vastasivat varsin hyvin odotuksiani, mikä heijastui heuristiikkarikkomusten selkeyteen ja analyysin luontevaan jaotteluun. Evernoten kohdalla heuristiikkojen sopivuus oli jo epätasaisempaa ja Trellossa kanssa oli myös omat haasteensa. Yksi selkeä syy tähän olivat mobiililaitteiden ja -sovellusten erityispiirteisiin liittyvät odotukset kuten se, ettei mobiilisovelluksen käyttöliittymässä ole tilaa dokumentaatiolle, kun näyttötila on rajallista. Näin ei analyysini perusteella selvästikään aina ole, ja välillä käyttöliittymässä on jopa ylimääräistä tilaa, jota on järkevää täyttää esimerkiksi

kyseisen näkymän käyttötarkoituksesta kertomalla. Myös odotukseni käyttäjädokumentaation informaatorakenteesta olivat ilmeisen konservatiivisia.

Osa-aineistojeni variaatiosta voidaan ainakin todeta, että päämääriltään näinkin samankaltaisten sovellusten käyttäjädokumentaatiota voidaan toteuttaa hyvin erilaisin tavoin. Eräs huomionarvoinen seikka on, ettei yhdessäkään tarkastelemassani sovelluksessa ollut kontekstisensitiivistä dokumentaatiota siinä merkityksessä kuin Android-käyttöjärjestelmän ohjeistuksissa tarkoitetaan, eli tietystä näkymästä ei päässyt suoraan juuri sen näkymän tehtäviä koskevaan dokumentaatioon. Käytönaikainen dokumentaatio esiintyi esimerkiksi vihjeteksteinä (*tooltip*) ja muina käyttöliittymän päälle aukeavina ohjeina ja ilmoituksina sekä tyhjiissä näkymissä esiintyvänä ohjeina ja vinkkeinä (ks. kuva 5, luku 6.1.1, ja kuva 9, luku 6.2.2) ja Evernoten tapauksessa vinkkisarjoina, jotka käyttäjä voi halutessaan jättää huomiotta. Trellon käyttäjädokumentaatio oli virkistävän käyttäjälähtöistä, kun käyttäjä pääsi dokumentaatioon heti kokeilemaan kuvailtuja toimintoja, ja sovelluksesta löytyi lisäksi perinteisempiä käyttöohjeita. Perinteisempää dokumentaatiota löytyi myös Pocketista käytönaikaisen dokumentaation lisäksi.

Analyysini perusteella vaikuttaa siltä, että osa valitsemistani heuristiikoista sopi mobiilisovellusten käyttäjädokumentaation käytettävyyden arviointiin varsin hyvin, kun taas osa oli sovellettavissa vaihtelevasti ja osa osoittautui mahdollisesti jopa melko tarpeettomiksi. Analyysini perusteella parannellut heuristiikat näkyvät taulukossa 6. Muualle siirretyt tai poistetut heuristiikat on vedetty yli ja uudet tai uuteen paikkaan siirtyneet heuristiikat on merkitty punaisella.

*Taulukko 6. Parannellut mobiilisovellusten käyttäjädokumentaation käytettävyysh euristiikat*

Numero	Heuristiikka
<b>1.</b>	<b>Dokumentaation löydettävyys</b>
<b>2.</b>	<b>Dokumentaation informaatorakenteen toimivuus</b>
2.1.	Tiedon löydettävyys
<del>2.2.</del>	<del>Tehtävien modulaarisuus</del>
2.2.	Visuaalisten elementtien tarkoituksenmukainen käyttö
<b>3.</b>	<b>Dokumentaation ja tuotteen samanaikaisen käytön helppous</b>
<b>4.</b>	<b>Tehtäväorientaatio</b>
<del>5.—</del>	<del>Dokumentaation ja tuotteen vastaavuus</del>

5.1.	<del>Dokumentaatiossa ja käyttöliittymässä käytetyn termistön vastaavuus ja johdonmukaisuus</del>
5.2.	<del>Dokumentaation käyttöliittymän ja tuotteen käyttöliittymän johdonmukaisuus</del>
<b>5.</b>	<b>Virheiden ehkäiseminen</b>
5.1.	Virhetiedon tarjoaminen käyttäjälle
5.2.	Varmistuksen kysyminen ennen lopullisten tai mahdollisesti haitallisten toimintojen suorittamista
<b>6.</b>	<b>Dokumentaation kielen tarkoituksenmukaisuus</b>
6.1.	Yleiskielisyys
6.2.	Käyttäjän suora puhuttelu
6.3.	<del>Tekstin ytimekkyys</del>
6.3.	Termistön vastaavuus ja johdonmukaisuus
<b>7.</b>	<b>Käytönaikaisen dokumentaation merkityksellisyys</b>
<b>8.</b>	<b>Käyttöjärjestelmän konventioiden seuraaminen</b>

Heuristiikat ovat siis monelta osin ennallaan. Esimerkiksi 1. *Dokumentaation löydettävyys*, 3. *Dokumentaation ja tuotteen samanaikaisen käytön helppous* ja 7. *Käytönaikaisen dokumentaation merkityksellisyys* ovat analyysini perusteella varsin yksiselitteisesti tärkeitä mobiilisovellusten käyttäjädokumentaation käytettävyyden kannalta. Myös heuristiikan 2. *Dokumentaation informaatorakenteen toimivuus* tiedon löydettävyyttä ja visuaalisten elementtien tarkoituksenmukaisuutta koskevat alaeuristiikat ovat ilman muuta tärkeitä: tiedon on löydettävä vaivattomasti ja myös kuvien oppimista ja muistamista tukeva vaikutus kävi analyysissäni ilmi. Sen sijaan tehtävien modulaarisuutta koskevan alaeuristiikan jätin pois, sillä ohjeet saattavat mobiilisovelluksissa olla pienempinäkin osina varsin toimivia eikä moduulien hahmottaminen aineistosta ollut aina aivan helppoa.

Heuristiikka 4. *Tehtäväorientaatio* säilyy ennallaan, sillä vaikka mobiilisovellusten käyttäjädokumentaatio saattaakin sisältää muutakin kuin varsinaista ohjeistavaa tekstiä, on silti tärkeää keskittyä niihin tehtäviin, joita käyttäjä haluaa sovelluksella suorittaa. Dokumentaation ja tuotteen vastaavuutta koskevan heuristiikan 5. päädyin poistamaan, sillä totesin, että termistöä koskeva alaeuristiikka sopisi paremmin kieltä koskevan heuristiikan alle ja dokumentaation ja sovelluksen käyttöliittymien johdonmukaisuutta koskeva alaeuristiikka on melko tarpeeton, koska ei liene perusteltua syytä olettaa, että nämä käyttöliittymät *eivät* vastaisi toisiaan, mikäli dokumentaatio on sovelluksen sisällä.

Virheiden ehkäisemistä koskevan heuristiikan koen edelleen tarpeelliseksi. Olen kuitenkin muuttanut mieltäni yleisestä vianetsintäosiosta, sillä tarkastelemissani sovelluksissa on ylipäätään varsin vähän virheiden mahdollisuuksia ja parasta olisi, että virhetieto annettaisiin käyttäjälle heti siinä kontekstissa, jossa virhe tapahtuu, käytönaikaisena dokumentaationa. Virhetiedon määrä tarkastelemissani sovelluksissa oli ylipäätään varsin vähäinen ja olisi hyvä, että sitä ylipäätään olisi tarjolla tarpeen mukaan, oli se sitten käytönaikaisessa dokumentaatioissa tai tietyn tehtävän ohjeiden yhteydessä. Tästä syystä olen muuttanut tätä heuristiikkaa merkitykseltään laajemmaksi: *5.1 Virhetiedon tarjoaminen käyttäjälle*. Varmistuksen kysymistä lopullisten tai mahdollisesti vahingollisten toimintojen yhteydessä koskevaa alaheuristiikkaa pidän yhä tärkeänä.

Dokumentaation kieleen liittyvät heuristiikat säilyvät pääosin ennallaan, mutta kielen ytimekkyyttä vaativan heuristiikan tilalle siirrän aiempiana mainitsemani termistön johdonmukaisuuden: olen myös laventanut sen koskemaan kaiken termistön johdonmukaisuutta, ei ainoastaan dokumentaation ja sovelluksen välistä johdonmukaisuutta. Kielen ytimekkyys on ylipäätään vaikea määritellä, kuten huomasin jo alkuperäisiä heuristiikkoja laatiessani, minkä lisäksi käytönaikainen dokumentaatio jossain määrin pakottaa ytimekkyteen rajallisen näyttötilan vuoksi.

Myös käyttöjärjestelmäkohtaisten konventioiden seuraamisen säilytän heuristiikoissani, sillä mitä useamman mobiilisovelluksen käyttäjädokumentaatio kyseessä olevan alustan konventioita noudattaa, sitä tutumpaa ja helpommin käytettävää dokumentaatiosta tulee käyttäjälle (ks. Hoehle & Venkatesh 2015, luku 2.3).

Parannellut heuristiikkani heijastavat myös mobiilisovellusten käyttäjädokumentaation ominaispiirteitä, joita ensimmäinen tutkimuskysymykseni koski. Osa-aineistojeni välillä oli paljon eroja, mutta yhteisiäkin piirteitä oli. Analyysini perusteella ainakin seuraavat piirteet vaikuttaisivat mobiilisovellusten käyttäjädokumentaatiolle ominaisilta:

- **Johdonmukaisuus sovelluksen käyttöliittymän tyylin kanssa:** dokumentaatio on kielellisesti ja visuaalisesti yhdenmukaista sovelluksen käyttöliittymän kanssa.
- **Sovelluksen käyttöliittymän ominaisuuksien hyödyntäminen:** dokumentaatio saattaa täyttää käyttöliittymässä olevaa tyhjää tilaa tai se voi jopa olla sovelluksen omilla työkaluilla toteutettu, jolloin se vastaa käyttöliittymää myös toiminnoiltaan.



- **Yleiskielisyys:** dokumentaation kieli on yleiskieltä. Erikoistermistöä ei juuri esiinny. Kielessä voi olla myös esimerkiksi humoristisia piirteitä, jos ne sopivat sovellukseen.
- **Dokumentaatiolle epätyypillisten tyylipiirteiden käyttö:** dokumentaatiossa saattaa ohjeistavan tekstin lisäksi esiintyä esimerkiksi humoristisia, inspiroivia tai markkinoivia piirteitä.
- **Varmistuksen kysyminen lopullisten tai mahdollisesti vahingollisten toimintojen yhteydessä.**

Nämä piirteet esiintyivät vähintään kahdessa kolmesta osa-aineistostani. Vaikka tutkimusaineistoni onkin pieni eikä kolmesta sovelluksesta voi tehdä suoria johtopäätöksiä mobiilisovellusten käyttäjädokumentaation ominaispiirteistä, pidän todennäköisenä, että nämä ovat ainakin jossain määrin yleisiä piirteitä. Esimerkiksi tyyllinen johdonmukaisuus ja yleiskielisyys ovat mobiilisovellusten yleisiä käytettävyySPIIRTEITÄ (ks. luku 2.3), ja siten sovellusten kehityksessä tavoiteltavia asioita. Osa piirteistä sisältyikin jo heuristiikkoihini, kuten juuri yleiskielisyys, johdonmukaisuus ja varmistuksen kysyminen. Heuristiikat saattavat ohjata ajattelua jonkin verran, ja Evernoten ja Trellos dokumentaatiota kävinkin läpi heuristiikkaheuristiikalta, minkä vuoksi heuristiikoissa esiintyneet piirteet ovat saattaneet analyysissä korostua ja jotain niiden ulkopuolelta on saattanut jäädä huomaamatta. Siitä huolimatta uskon, että havaintoni ovat relevantteja, sillä heuristiikkani perustuivat laajaan pohjamateriaaliin.

Toinen tutkimuskysymykseni koski käyttäjädokumentaation roolia käyttäjäkokemuksen luomisessa. Analyysini perusteella mobiilisovellusten käyttäjädokumentaatiolla tosiaan on mahdollisuuksia vaikuttaa käyttäjäkokemukseen. Dokumentaation rooli käyttäjäkokemuksen luomisessa vaikuttaa liittyvän suurimmilta osin käyttäjän kognitiivisen kuormituksen keventämiseen. Tähän voidaan pyrkiä esimerkiksi esittelemällä sovelluksen keskeiset toiminnot ensimmäisellä käyttökerralla, kuten Pocketin ja Evernoten käytönaikaisessa dokumentaatiossa tehdäänkin. Dokumentaation käytönaikaisuus ylipäättään tekee sovelluksen käytöstä vaivattomampaa, sillä tällöin käyttäjän ei tarvitse siirtyä tehtäviensä ja dokumentaation välillä.

Kognitiivisen rasituksen minimoimiseen voidaan pyrkiä myös dokumentaation hyvällä havainnollistavalla kuvituksella, jos dokumentaatio sijaitsee sovelluksen toiminnoista erillään, tai antamalla käyttäjälle mahdollisuus kokeilla toimintoja dokumentaation yhteydessä. Havainnollistavat kuvat, jollaisia Pocketin dokumentaatiossa on käytetty, auttavat käyttäjää hahmottamaan ja muistamaan ohjeen paremmin kuin pelkkä teksti, ja esimerkiksi Trellossa

oleva toiminnallinen dokumentaatio auttaa myös käyttäjää muodostamaan paremman mentaalisen mallin sovelluksen toiminnasta. Myös sovelluksessa olevan näyttötilan hyvällä käytöllä voidaan parantaa käyttäjäkokemusta: tilan rajallisuus on otettava huomioon, eikä käyttäjädokumentaatio saa häiritä käyttäjän toimintaa (Lakatos 2016), mutta toisaalta käyttöliittymässä saattaa toisinaan olla jopa ylimääräistä tyhjää tilaa, joka ei näytä hyvältä. Tällöin voidaan esimerkiksi antaa tietoa kyseisen näkymän käyttötarkoituksesta (Babich 2017), kuten Pocketissa ja Evernotessa onkin tehty. Seuraavassa luvussa esitänkin johtopäätökseni ja mahdollisia jatkotutkimuksen aiheita.

## 7 Päätelmät

Tutkimuskysymykseni koskivat mobiilisovellusten käyttäjädokumentaation ominaispiirteitä ja sen roolia käyttäjäkokemuksen luomisessa. Analyysini perusteella käyttäjädokumentaation ominaispiirteet liittyvät paljolti sovelluksen käyttöliittymän ja dokumentaation väliseen yhteyteen sekä dokumentaation tyyliin. Vaikka kolme sovellusta on varsin pieni aineisto, uskon löytämäni piirteiden olevan yleisiä, sillä osa niistä on suorassa yhteydessä yleisesti hyvinä käytettävyyksiä pidettyihin ominaisuuksiin, kuten yleiskielisyyteen ja käyttöliittymän johdonmukaisuuteen. Analysoimissani sovelluksissa oli melko paljon eroja, mutta yhteisiä piirteitä olivat kielellinen ja visuaalinen yhdenmukaisuus sovelluksen käyttöliittymän kanssa, sovelluksen käyttöliittymän ominaisuuksien hyödyntäminen, yleiskielisyys, dokumentaatiolle epätyypillisten tyylipiirteiden käyttö ja varmistuksen kysyminen, kun käyttäjä on suorittamassa lopullista tai mahdollisesti vahingollista toimintoa.

Myös käyttäjäkokemuksen luomisessa käyttäjädokumentaatiolla on paikkansa: yksi suurimmista mobiilisovellusten käyttäjäkokemuksen ongelmista on käyttäjän kokema kognitiivinen kuormitus, ja analyysini perusteella tähän voidaan jossain määrin vaikuttaa tekemällä käyttäjädokumentaatiosta helposti omaksuttavaa ja dokumentaation ja sovelluksen samanaikaisesta käytöstä helppoa. Kun käyttösessiot ovat tyyppillisesti lyhyitä, käyttäjän huomio sirpaloitunutta ja näyttötila rajallinen, kuten esimerkiksi Budiu (2015, luku 2.2) toteaa, hyvin toteutettu käyttäjädokumentaatio voi auttaa käyttäjää suorittamaan tehtävänsä vaivattomammin. Huomion sirpaleisuuteen voidaan vastata ytimekkäällä käytönaikaisella dokumentaatiolla ja tekemällä tarvittavista siirtymistä sovelluksen ja dokumentaation välillä mahdollisimman helppoja ja nopeita: ihanteellista olisi, että dokumentaatio olisi joko suoraan samassa tilassa sovelluksen toimintojen kanssa, mikäli tilaa on, tai näkymäkohtaista, jotta käyttäjä pääsee suoraan tarvitsemansa tiedon pariin (Lakatos 2016). Myös havainnollinen kuvitus dokumentaatiossa auttaa käyttäjää sisäistämään ja muistamaan lukemaansa. Lisäksi positiivista käyttäjäkokemusta voidaan vahvistaa käyttämällä dokumentaatiossa samoja käyttäjäkokemuksen parantamisen keinoja kuin sovelluksessa muutenkin, esimerkiksi laadukkaita kuvia ja hyvin suunniteltua kielellistä tyyliä.

Tämän tutkimuksen perusteella vaikuttaa myös, että hypoteesini minimalismin soveltuvuudesta mobiilisovellusten käyttäjädokumentaation tarkasteluun pitää monilta osin paikkansa. Minimalismin keskeisistä periaatteista esimerkiksi tiedon löydettävyys ja tehtäväorientaatio (ks. van der Meij & Carroll 1995, luku 3.3) sekä kerrostaminen (ks. Farkas 1998, luku 3.2)

sopivat myös mobiilisovellusten käyttäjädokumentaation suunnitteluun. Tiedon on oltava helposti löydettävissä, mieluiten käytönaikaisena dokumentaationa, jotta dokumentaation ja sovelluksen samanaikainen käyttö olisi mahdollisimman saumatonta. Tehtäväorientoitunut dokumentaatio sisältää ohjeet niiden tehtävien tekemiseen, joita käyttäjä todellisuudessa haluaa tehdä, ja kerrostamalla informaatiota voidaan ottaa huomioon erilaisten käyttäjien tarpeet ja tavoitteet. Myös ytimekkyys on keskeistä jo mobiilisovellusten rajallisen näyttötilan vuoksi, ja dokumentaatioon onkin valittava merkityksellisimmät aiheet. Toisaalta mobiilisovellusten käyttäjädokumentaatio voi sisältää myös asioita, jotka eivät ole välttämättömiä tehtävien suorittamisen kannalta, kuten esimerkiksi humoristisia tai markkinoivia piirteitä. Yksi minimalismin keskeisimmistä periaatteista, virheiden ehkäiseminen ja niistä palautumisen tukeminen, on puolestaan aineistoni analyysissä vähemmän esillä, mutta juuri ennaltaehkäiseminen on mobiilisovelluksissa keskeistä, ja tämä voidaan tehdä erityisesti kysymällä käyttäjältä varmistusta lopullisten tai mahdollisesti vahingollisten toimintojen yhteydessä.

Vaikuttaakin siltä, että käyttäjäkokemus ja minimalismi kietoutuvat hyvin yhteen. Minimalismin periaatteista muun muassa tiedon löydettävyys ja merkityksellisyys tukevat mobiilisovellusten käyttäjäkokemusta, sillä ne osaltaan keventävät käyttäjän kognitiivista kuormitusta. Kaiken kaikkiaan mobiilisovellusten käyttäjädokumentaatioissa korostuu vaivattomuus ja keskeisten sisältöjen priorisointi.

Tutkimusaineistoni oli näin perustavan tutkimuksen tarpeisiin suppeahko, mutta pro gradu - tutkielman puitteissa laajemman aineiston tarkastelu ei olisi ollut mahdollista näin syvällisellä tasolla. Vaikka tutkimusaineistokseni valitsemani sovellukset ovat käyttötarkoituksiltaan suhteellisen samankaltaisia, niiden dokumentaatio on kuitenkin toteutettu varsin eri tavoin. Olisi kiinnostavaa tietää, onko mobiilisovellusten käyttäjädokumentaatio yleisesti hyvin vaihtelevaa vai onko aineistoni sattumalta näin monipuolinen. Laajemman, käyttötarkoituksiltaan vaihtelevamman sovellusjoukon tarkastelu voisi olla paikallaan, mutta toisaalta myös esimerkiksi mobiilipelien käyttäjädokumentaatio olisi hedelmällinen tutkimuskohde: kuinka pelaajakokemuksen luominen ulottuu käyttäjädokumentaatioon ja onko dokumentaatio itsessään kenties pelillistä?

Heuristinen asiantuntija-arviointi oli varsin onnistunut menetelmävalinta tämän tutkielman tarkoituksiin. Kun aiempaa tutkimusta aiheesta tai juuri tähän tarkoitukseen luotuja heuristiikkoja ei ollut, mobiilisovellusten käyttäjäkokemus- ja käytettävyydestä sekä

teknisen viestinnän teoriaa laajasti yhdistämällä onnistuin luomaan hyvän viitekehyksen mobiilisovellusten käyttäjädokumentaation ominaisuuksien kartoittamiseen. Menetelmän tehokkuutta rajoitti todennäköisesti jonkin verran se, että arvioijia oli vain yksi, kun yleensä suositellaan 3–5 asiantuntija-arvioijaa (Nielsen 1995b). Analyysissäni kohtaamiani haasteita olivat myös yksittäisten heuristiikkojen ajoittainen heikko soveltuvuus tiettyjen osa-aineistojen arvioimiseen, minkä vuoksi erityisesti Trellon analyysin selkeä jäsentely oli vaikeaa, ja heuristiikkojen mahdollinen ajattelua ohjaileva vaikutus. Myös käyttäjäkokemuksen subjektiivisuus voi vaikuttaa löydösten tulkintaan, mutta olen pyrkinyt pohjaamaan kaikki tulkintani teorialähteisiin. Näistä haasteista huolimatta pidän analyysiäni onnistuneena ja monipuolisena: se luo hyvän pohjan jatkotutkimukselle, jota mielestäni tarvitaan.

Koska osa heuristiikoistani osoittautui vaikeasti sovellettaviksi, paranneltuun versioon nostin joitakin asioita hieman yleisemmälle tasolle, jotta ne soveltuisivat mobiilisovellusten käyttäjädokumentaation arviointiin mahdollisimman laajasti, sillä kuten mainittua, dokumentaation toteutustapojen kirjo vaikuttaa varsin laajalta. Vaikka aineistoni koostuikin vain kolmen sovelluksen käyttäjädokumentaatiosta, uskon heuristiikkojen muokkaamisen olevan järkevää: jos osa heuristiikoista vaikutti liian suppeilta tai yksityiskohtiin puretuilta jo tämän aineiston tarkasteluun, niiden soveltamisessa laajempiin aineistoihin olisi todennäköisesti ongelmia.

Teoriapohjaksi valitsemani aihepiirit sopivat mielestäni erinomaisesti yhteen ja muodostivat vankan ja monitahoisen kokonaisuuden. Tämän laajaan ja monipuolisen teoriapohjan perusteella luomani heuristiikat paikkaavat onnistuneesti teknisen viestinnän tutkimuskentässä olevaa puutetta ja uskon heuristiikkojen olevan sovellettavissa myös jatkotutkimuksen tarpeisiin. Vaikka heuristiikat onkin luotu erityisesti Android-sovelluksia silmällä pitäen, ne todennäköisesti soveltuvat myös iOS-sovellusten tarkasteluun: tällöin käyttöjärjestelmän konventioiden toteutumista arvioitaessa on tietenkin katsottava juuri iOS:n ohjeistuksia. Uskon myös heuristiikkojen soveltuvan mobiilisovellusten käyttäjädokumentaation käytännön toteutuksen tueksi ja arviointityökaluksi.

Mobiililaitteet ja niille kehitetyt sovellukset kehittyvät jatkuvasti. Onkin kiinnostavaa nähdä, kuinka laitteiden ja sovellusten affordanssit ja toiminnallisuudet tulevat lisääntymään ja muuttumaan, ja miten tämä vaikuttaa käyttäjädokumentaatioon. Jos esimerkiksi luvussa 2.2 mainitsemani haptiset eli tuntoaistiin perustuvat palautteet yleistyvät ja monipuolistuvat, tämä voi tuoda uusia haasteita dokumentaation suunnittelulle ja toteutukselle.

Toivon tämän tutkielman tulosten toimivan pohjana jatkotutkimukselle, jotta mobiili-sovellusten käyttäjädokumentaatiota tutkittaisiin laajemmin ja sen luomiseen liittyviä ohjeistuksia ja käytäntöjä saataisiin vankemmalle pohjalle. Kyseessä on hyvin ajankohtainen aihe, ja tutkimuksella voidaan myös tuoda näkyvämmäksi tämänkin teknisen viestinnän alan merkitystä tuotteiden käyttäjille.

## Lähteet

### Aineisto

- Evernote Corporation 2018. *Evernote* v. 7.17. Uusin versio saatavilla: <https://play.google.com/store/apps/details?id=com.evernote> [Luettu 22.3.2018.]
- Read It Later 2018. *Pocket* v. 6.7.5.1. Uusin versio saatavilla: <https://play.google.com/store/apps/details?id=com.ideashower.readitlater.pro> [Luettu 22.3.2018.]
- Trello, Inc 2018. *Trello* v. 4.8.0.10622-candidate. Uusin versio saatavilla: <https://play.google.com/store/apps/details?id=com.trello> [Luettu 22.3.2018.]

### Kirjallisuus

- Android Developers 2017. *Help*. Saatavilla: <https://developer.android.com/design/patterns/help.html> [Luettu 7.12.2017.]
- Android Developers 2018. *Action Bar*. Saatavilla: <https://developer.android.com/design/patterns/actionbar.html> [Luettu 1.4.2018.]
- Apple Developers 2017. *Terminology – Visual Design*. Saatavilla: <https://developer.apple.com/ios/human-interface-guidelines/visual-design/terminology/> [Luettu 7.12.2017.]
- Apple Developers 2018. *Adaptivity and Layout*. Saatavilla: <https://developer.apple.com/ios/human-interface-guidelines/visual-design/adaptivity-and-layout/> [Luettu 29.3.2018.]
- Babich, Nick 2017. *Best Practices for Onboarding*. Saatavilla: <http://babich.biz/best-practices-for-onboarding/> [Luettu 18.4.2018.]
- Benyon, David 2014. *Designing Interactive Systems: A comprehensive guide to HCI, UX and interaction design*. 3. painos. Boston: Pearson.
- Budiu, Raluca 2015. *Mobile User Experience: Limitations and Strengths*. Saatavilla: <https://www.nngroup.com/articles/mobile-ux/> [Luettu 23.11.2017.]
- Bunton, Cam 2016. *Android N quick-tip: How to activate split-screen multitasking mode*. Saatavilla: <https://9to5google.com/2016/05/04/how-to-multiwindow-android-n-split-screen/> [Luettu 29.3.2018.]
- Carroll, John M. 1990. *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. Cambridge, Massachusetts: The MIT Press.
- Carroll, John M. & van der Meij, Hans 1998. Ten Misconceptions about Minimalism. Teoksessa John M. Carroll (toim.), *Minimalism Beyond the Nurnberg Funnel*. Cambridge, Massachusetts: The MIT Press. 55–90.
- Cummings, Michael 2010. *UX Design Defined*. Saatavilla: <http://uxdesign.com/ux-defined> [Luettu 3.1.2018.]
- Dogtiev, Artyom 2018. *App Download and Usage Statistics 2017*. Saatavilla: <http://www.businessofapps.com/data/app-statistics/> [Luettu 21.1.2018.]

- Draper, Stephen W. 1998. Practical Problems and Proposed Solutions in Designing Action-Centered Documentation. Teoksessa John M. Carroll (toim.), *Minimalism Beyond the Nurnberg Funnel*. Cambridge, Massachusetts: The MIT Press. 349–374.
- Ellis, Megan 2017. *5 Mobile Games You Control in Crazy Ways*. Saatavilla: <https://www.makeuseof.com/tag/mobile-games-crazy-control/> [Luettu 18.4.2018.]
- Farkas, David K. 1998. Layering as a Safety Net for Minimalist Documentation. Teoksessa John M. Carroll (toim.), *Minimalism Beyond the Nurnberg Funnel*. Cambridge, Massachusetts: The MIT Press. 247–274.
- Garrett, Jesse James 2002. *The Elements of User Experience: User-Centered Design for the Web*. Indianapolis, Indiana: New Riders.
- Google Play 2018. *Google Play*. Saatavilla: <https://play.google.com/store> [Luettu 22.3.2018.]
- Guren, Leah 2014. Minimalism in documentation. *tcworld* January 2014. Saatavilla: <http://www.tcworld.info/rss/article/minimalism-in-documentation/> [Luettu 24.1.2018.]
- Hoehle, Harmut & Venkatesh, Viswanath 2015. Mobile Application Usability: Conceptualization and Instrument Development. *MIS Quarterly* 39:2. 435–472.
- Hoehle, Hartmut, Ruba Aljafari & Viswanath Venkatesh 2016. Leveraging Microsoft's mobile usability guidelines: Conceptualizing and developing scales for mobile application usability. *Int. J. Human-Computer Studies* 89. 35–53.
- Horton, William 1994. *Designing and Writing Online Documentation: Hypermedia for Self-Supporting Products*. 2. painos. New York: Wiley.
- Huckin, Thomas N. & Olsen, Leslie A. 1991. *Technical Writing and Professional Communication for Nonnative Speakers of English*. 2. painos. New York: McGraw-Hill.
- Hulick, Samuel s. d. *Bulletproof User Onboarding*. Saatavilla: <https://www.useronboard.com/bulletproof-user-onboarding/> [Luettu 18.4.2018.]
- Ismailov, David 2015. *The Future of User Experience is Visual Design*. Saatavilla: <https://uxplanet.org/the-future-of-user-experience-is-visual-design-af2fc818bc9> [Luettu 9.4.2017.]
- ISO 9241-210 2012. *Ihmisen ja järjestelmän vuorovaikutuksen ergonomia. Osa 210: Vuorovaikutteisten järjestelmien käyttäjäkeskeinen suunnittelu*. Helsinki: Suomen Standardisoimisliitto.
- Korvenranta, Heta 2005. Asiantuntija-arvioinnit. Teoksessa Ovaska, Saila, Anne Aula & Päivi Majaranta (toim.), *Käytettävyystutkimuksen menetelmät*. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos B-2005-1. 111–124.
- Kuusinen, Kati & Mikkonen, Tommi 2014. On Designing UX for Mobile Enterprise Apps. *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. Konferenssijulkaisu. Saatavilla: <http://ieeexplore.ieee.org/helios.uta.fi/document/6928815/> [Luettu 24.10.2017.]
- Lakatos, Diána 2016. *An overview of context-sensitive and embedded help formats*. Saatavilla: <https://pronovix.com/blog/overview-context-sensitive-and-embedded-help-formats> [Luettu 21.3.2018.]



- Lehrner-Mayer, Karina 2016. Five tips for creating documentation that focuses on the user. *tcworld*, July 2016. Saatavilla: <http://www.tcworld.info/e-magazine/technical-communication/article/five-tips-for-creating-documentation-that-focuses-on-the-user/> [Luettu 28.2.2018.]
- Libin, Phil 2008. *Evernote Public Launch!* Saatavilla: <https://web.archive.org/web/20120904074223/http://blog.evernote.com/2008/06/24/evernote-public-launch/> [Luettu 22.4.2018.]
- Lucero, Andrés, Jussi Holopainen, Elina Ollila, Riku Suomela & Evangelos Karapanos 2013. The playful experiences (PLEX) framework as a guide for expert evaluation. *DPPI '13 Proceedings of the 6th International Conference on Designing Pleasurable Products and Interfaces*. 221–230.
- Mali, Nabeena 2017. *5 tips to help first-time users fall in love with your app*. Saatavilla: <https://thenextweb.com/contributors/2017/07/15/5-tips-help-first-time-users-fall-love-app/> [Luettu 17.4.2018.]
- Material Design 2017. Saatavilla: <https://material.io/guidelines/style/writing.html> [Luettu 8.12.2017.]
- McMurrey, David s. d. *Bulleted and Numbered Lists: Provide emphasis and sequence*. Saatavilla: <https://www.prismnet.com/~hcexres/textbook/lists.html> [Luettu 18.4.2018.]
- Meyer, Kate 2016. *Reading Content on Mobile Devices*. Saatavilla: <https://www.nngroup.com/articles/mobile-content/> [Luettu 24.11.2017.]
- Microsoft 2012. *Microsoft Manual of Style*. 4. painos. Redmond, Washington: Microsoft Press.
- Moss, Richard 2015. *Haptic technology: The next frontier in video games, wearables, virtual reality, and mobile electronics*. Saatavilla: <https://newatlas.com/haptic-tech-vr-wearables-games-sightlence/35616/> [Luettu 29.3.2018.]
- Nielsen, Jakob & Rolf Molich 1990. Heuristic evaluation of user interfaces. *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256.
- Nielsen, Jakob 1993. *Usability Engineering*. San Francisco (CA): Academic Press.
- Nielsen, Jakob 1995a. *10 Usability Heuristics for User Interface Design*. Saatavilla: <https://www.nngroup.com/articles/ten-usability-heuristics/> [Luettu 22.1.2018.]
- Nielsen, Jakob 1995b. *How to Conduct a Heuristic Evaluation*. Saatavilla: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/> [Luettu 22.1.2018.]
- Nielsen, Jakob 2012. *Usability 101: Introduction to Usability*. Saatavilla: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> [Luettu 3.4.2018.]
- Norman, Don & Nielsen, Jakob s. d. *The Definition of User Experience (UX)*. Saatavilla: <https://www.nngroup.com/articles/definition-user-experience/> [Luettu 7.11.2017.]
- Ovaska, Saila, Anne Aula & Päivi Majaranta 2005. Johdatus käytettävyytutkimukseen. Ovaska, Saila, Anne Aula & Päivi Majaranta (toim.), *Käytettävyytutkimuksen menetelmät*. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos B-2005-1. 1–16.
- Pocket 2018. *About Pocket*. Saatavilla: <https://getpocket.com/about> [Luettu 9.2.2018.]

- Purho, Vesa 2000. Heuristic Inspections for Documentation – 10 Recommended Documentation Heuristics. *Usability Interface, STC Usability SIG Newsletter* 6:4.
- Redish, Janice 1998. Minimalism in Technical Communication: Some Issues to Consider. Teoksessa John M. Carroll (toim.), *Minimalism Beyond the Nurnberg Funnel*. Cambridge, Massachusetts: The MIT Press. 219–246.
- Rosenbaum, Stephanie 1998. Follow-up on Training in Minimalism: How are Technical Communicators Using Minimalism? Teoksessa John M. Carroll (toim.), *Minimalism Beyond the Nurnberg Funnel*. Cambridge, Massachusetts: The MIT Press. 119–148.
- Schriver, Karen A. 1997. *Dynamics in Document Design*. New York: John Wiley & Sons, Inc.
- Sirach, Blake 2017. *Tactile Design for Mobile Apps*. Saatavilla: <https://willowtreeapps.com/ideas/tactile-design-for-mobile-apps> [Luettu 18.4.2018.]
- Slack 2018. *Roles and permissions in Slack*. Saatavilla: <https://get.slack.help/hc/en-us/articles/201314026-Roles-and-permissions-in-Slack> [Luettu 20.1.2018.]
- Statista 2018. *Number of smartphone users worldwide from 2014 to 2020 (in billions)*. Saatavilla: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/> [Luettu 19.4.2018.]
- Tan, Wei-Siong, Dahai Liu & Ram Bishu 2009. Web evaluation: Heuristic evaluation vs. user testing. *International Journal of Industrial Ergonomics* 39. 621–627.
- Tilastokeskus 2017. 2. *Internetin käyttö mobiililaitteilla*. Saatavilla: [https://www.stat.fi/til/sutivi/2017/13/sutivi\\_2017\\_13\\_2017-11-22\\_kat\\_002\\_fi.html](https://www.stat.fi/til/sutivi/2017/13/sutivi_2017_13_2017-11-22_kat_002_fi.html) [Luettu 10.4.2018.]
- Tinder 2018. *What Is Tinder?* Saatavilla: <https://www.help.tinder.com/hc/en-us/articles/115004647686-What-is-Tinder-> [Luettu 15.3.2018.]
- Trello 2018. *About Trello*. Saatavilla: <https://trello.com/about> [Luettu 22.4.2018.]
- Van der Meij, Hans & Carroll, John M. 1995. Principles and heuristics for designing minimalist instruction. *Technical Communication* 42(2), 243–261.
- Wikipedia 2018. *Mobile app*. Saatavilla: [https://en.wikipedia.org/wiki/Mobile\\_app#Distribution](https://en.wikipedia.org/wiki/Mobile_app#Distribution) [Luettu 26.4.2018.]
- Windows Dev Center 2017. *Guidelines for App Help*. Saatavilla: <https://docs.microsoft.com/en-us/windows/uwp/design/in-app-help/guidelines-for-app-help> [Luettu 20.11.2017.]
- Yáñez Gómez, Rosa, Daniel Cascado Caballero & José-Luis Sevillano 2014. Heuristic Evaluation on Mobile Interfaces: A New Checklist. *The Scientific World Journal* 2014.

## **English Summary**

### **Introduction**

The percentage of people using mobile devices and applications developed for these devices has been growing fast in recent years, ever since Apple and Google opened their app stores in 2008 (Wikipedia). The number of smartphone users is projected to reach 2.53 billion worldwide in 2018, that is, 36 percent of world's population (Statista 2018). In Finland, 94 percent of people under 55 years of age used smartphones in 2017 (Tilastokeskus 2018). In 2017, the amount of mobile applications in Google Play and Apple App Store was 3 million and 2.2 million, respectively (Dogtiev 2018). However, despite mobile devices rapidly becoming more common and the need for mobile application documentation increasing accordingly, practically no research exists on user documentation for mobile applications. The aim of this Master's Thesis is to study the characteristics of existing mobile application documentation and the role of user documentation in the user experience of mobile applications, and to create heuristics for evaluating the usability of the documentation.

While mobile usability heuristics exist, very few of them have anything to say about user documentation. There are significant differences between the usage and contexts of use between mobile and desktop applications, and these differences affect documentation design as well as the design of the applications themselves. Notable differences include the restrictions caused by the smaller screen in mobile applications and the tendency of mobile users to multitask: mobile applications are often used while on the move, with more outside interruptions. These differences, among others, need to be considered in designing user documentation for mobile applications.

I base my study on user experience (UX) and usability research as well as the minimalist approach in technical communication. The central link between these theories is user-centeredness: the user is always the key when designing for UX and usability, and minimalism is also centred around the needs and goals of the user. Both are concerned with making the usage of the system as easy, effective and pleasant as possible for the target user.

As there are no existing usability heuristics for user documentation in mobile applications, I will create my own set of heuristics based on the aforementioned UX and usability research and existing heuristics for technical documentation as well as user documentation guidelines for the

mobile operating systems Android and iOS. The heuristics I will utilise in creating my own are the *10 Recommended Documentation Heuristics* by Vesa Purho (2000) and the *Principles and Heuristics for Designing Minimalist Documentation* by Hans van der Meij and John M. Carroll (1995). Additionally, I will utilise documentation-related elements from the mobile application usability heuristics by Rosa Yáñez Gómez, Daniel Cascado Caballero, and José-Luis Sevillano (2014).

Using these heuristics, I intend to find answers to the following questions: what are the characteristics of user documentation of mobile applications and what is their role in creating positive user experience? I will evaluate the user documentation of three mobile applications: Pocket, Evernote, and Trello.

## **User Experience and Usability in Mobile Applications**

The Finnish Standards Association SFS (ISO 9241-210 2012, 17) defines *user experience* or *UX* as “user’s perceptions and responses that result from the use and/or anticipated use of a system, product or service”. According to Don Norman and Jakob Nielsen (s. d.), “‘User experience’ encompasses all aspects of the end-user's interaction with the company, its services, and its products”. *User experience design* is the process of ensuring that every interaction between the user and the product works as intended and that the user’s expectations and needs have been understood and taken into consideration (Garrett 2002, 21). User experience design draws from several different disciplines, such as graphical and industrial design, engineering, and interface design (Norman & Nielsen s. d.).

*Usability* has various definitions. According to The Finnish Standards Association SFS (ISO 9241-210 2012, 17), usability means “extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”. One of the most commonly used definitions is the one by Nielsen (1993, 26), which divides usability into five elements: learnability, efficiency, memorability, errors, and satisfaction. In short, a usable system is easy and pleasant to use.

Jesse James Garrett (2002) presents a model of user experience design called the *Five Planes* model. According to Garrett, the process of user experience design consists of five hierarchical stages called, from top to bottom, the *Surface Plane*, the *Skeleton Plane*, the *Structure Plane*, the *Scope Plane*, and the *Strategy Plane*. *Strategy* is the first plane to consider when designing user experience: it includes the goals of both the developers and users. The other planes are

built on top of the strategy, advancing from abstract to concrete. Michael Cummings (2010) presents his own model based on Garrett's *Five Planes*, but does so from a more user-centred perspective: his model consists of the different aspects of an interactive system that make up the user experience. At the top of his model are the aspects that the user is the most aware of, such as the language and visuals of the system, and at the bottom are the aspects that the user hardly ever thinks about, such as the programming and interaction design. These aspects form a whole that Cummings (ibid.) compares to a language: it is "more than the sum of its parts".

Although Garrett's (ibid.) model is mainly concerned with websites, many of the same principles also apply to mobile user experience. However, mobile devices and their use have their own special characteristics that affect user experience design. The use sessions of mobile applications are typically shorter than those of desktop apps: according to Budiu (2015), the average session on mobile is 72 seconds, while on desktop it is 150 seconds. Additionally, mobile users are often multitasking or using the devices on the move and their attention is more fragmented. This is not always the case, but these aspects must be taken into consideration.

Budiu (ibid.) points out that although the screen sizes of smart phones have been growing, they are still relatively small. One screenful on a smart phone fits much less content than, for example, on a laptop. This increases the cognitive load of the user as they must use their short-term memory to remember the information that is not currently on the screen. The small size of the screen also forces application developers to consider carefully what they want to display.

Because the use of mobile applications is easily interruptible, the application should make it easy to resume what the user was doing. Ways to achieve this include making the tasks as simple as possible and prioritising the most important content and features so that they are easy to access. Mobile users often switch between multiple apps, and the application should save the state the user is in so that the user can resume where they left off.

The touch screen also has its strengths and weaknesses. It allows the use of gestures. A well-known example of this is in the dating application Tinder (2018): the application displays profiles of nearby users and the user can swipe right if they are interested in the person, and left if they are not. Application-specific gestures must be introduced to the user, because otherwise there is no way for them to know how to use them. If properly documented, gestures add a versatile layer to the user interface. One of the weaknesses of the touch screen are accidental touches. According to Budiu (2015), the small size of the touch screen causes difficulties in

input: the keyboard on the screen is small and does not offer same kind of physical feedback as a physical keyboard would. Actions should be reversible so that accidental touches can be undone.

Although there are many challenges in designing user experience for mobile applications, Budiu (*ibid.*) notes that there are also opportunities that are unique to mobile devices. For example, GPS, camera, and microphone can be utilised to minimise the effort required from the user: camera can be used to read bar codes and credit cards, and finger print identification can be used instead of passwords. Movement can also be used to control the application: this is mostly utilised in mobile games (see e.g. Ellis 2017).

User documentation can be used to address the challenges of user experience design. The cognitive load can be decreased by making the simultaneous use of the application and the documentation as easy as possible, ideally with context-sensitive documentation that is either located in the same user interface view with the application's functions or directly accessible from the view it concerns. The documentation should also be concise and to-the-point to reduce the impact of possible interruptions.

Hartmut Hoehle and Viswanath Venkatesh (2015, 436–437) state that many of the usability problems in mobile applications are caused by the lack of consideration for the characteristics and contexts that are mobile-specific, for example, the small screens. According to Hoehle and Venkatesh (*ibid.*, 440), the same is true for mobile usability research. In addition, they state that the research does not sufficiently build upon previous work and different researchers use different conceptualisations and scales for measuring usability. For this reason, Hoehle and Venkatesh (*ibid.*) aim to conceptualise mobile usability by creating a set of usability attributes or constructs. They present these attributes in the context of previous research, thus bridging the gap between the existing studies.

Hoehle and Venkatesh (*ibid.*) base their list of usability attributes on Apple's user experience guidelines. They (*ibid.*, 441–450), present 19 attributes which they divide into six categories. Some of these attributes can be considered somewhat common to all interactive systems, but many are very mobile-specific: the application's ability to handle different screen orientations, fingertip-size controls, and de-emphasis of user settings are almost exclusively related to the mobile context. They underline the characteristics and contexts central to mobile application use. Some of the attributes are also clearly connected to user documentation, such as the

conciseness of language and user-centred terminology: these characteristics also bear resemblance to some prominent principles of minimalism, which I will address in the next chapter.

## **Minimalism in Technical Communication**

In technical communication, minimalism is an approach that was first comprehensively introduced by Carroll (1990). It was born as a countermove for the *systems approach*, which was an approach to technical documentation characterised by its tendency to decompose the instructional tasks into sequences of tiny components. Carroll (ibid., 5) states that the approach was not task-oriented and did not answer the needs of the users. The users were not given any freedom to pursue their own goals or the possibility to utilise their prior knowledge. Minimalism, on the other hand, is characterised by its task orientation and user-centeredness.

According to Carroll (ibid., 7), the goal of minimalism is to remove the barriers that prevent the user from learning. Stephen W. Draper (1998, 360) states that all characteristics of minimalism are centred around the user. Because the user interfaces today are easy to use, the documentation should focus on the tasks of the user instead of the product's features and functionalities (ibid., 350, 368). The documentation may also remedy usability issues: the complementary relationship between the documentation and the product is central to minimalism (ibid., 357).

Van der Meij and Carroll (1995, 245) present the principles of minimalism, along with a set of heuristics for designing minimalist documentation. The principles are the following:

1. Choose an action-oriented approach
2. Anchor the tool in the task domain
3. Support error recognition and recovery
4. Support reading to do, study, and locate

According to the first principle (van der Meij & Carroll 1995, 244–247), the user should be given an immediate opportunity to act, to do something that is meaningful to them. Long introductions take time away from learning: minimalism balances between action and knowledge. The users should also be encouraged to try things out for themselves. They should also always have the feeling of control and security, a feeling that they can return to guided learning whenever they need. Additionally, the documentation should not disturb the user.

The second principle (van der Meij & Carroll 1995, 248–250) is concerned with real tasks. For the user, the product is not an end in itself but a tool to help them reach their goals. The documentation should be based on actual tasks and the structure of the documentation should reflect the structure of these tasks. This makes the instructions easier to follow. When instructions are centred around real tasks with a connection to the user's prior knowledge and skills, the user is more motivated to carry out the tasks.

The third principle concerns errors. The documentation should help the user to recognise errors and to recover from them. According to van der Meij and Carroll (ibid., 252–253), the best way to remedy errors is simply to help users avoid them, which can be achieved using hints. User testing is also elemental for effective error prevention. Error information should be provided near actions that are error-prone or when recovering from the error would be difficult. The instructions for correcting errors should be located near the actions with which the error may occur so that the user can recognise the error immediately.

The fourth principle (van der Meij & Carroll 1995, 256–157) pertains to the structure of the documentation. The documentation should be concise: not everything needs to be explained. The tasks should be quick to complete, and if explanations are needed, they should be short. When everything is not explained, the user can utilise their own prior knowledge, which facilitates learning. The chapters of the documentation should be independent so that reading only the chapter that is immediately concerned with the task at hand would be enough, and the user would not have to search for additional information elsewhere.

The challenges of minimalism include the question of how much information can be omitted (Rosenbaum 1998, 122). David Farkas (1998, 249) points out that if the information is insufficient, the user may not be able to complete the task or it may take more time. Another problem is the tendency to base minimalistic documentation on the idea of users that are active, learning-oriented, and that learn best by doing (Redish 1998, 220). Draper (1998, 364–365) notes that users act differently depending on their personal goals – whether they are reading documentation to learn or to get their work done. The same documentation may not work for all users or even the same user in all situations. One tool for considering different users' needs is *layering*: for example, marking the shortcuts for more advanced users so that the novice users can disregard them (Farkas 1998, 254).



Minimalism is a very software-centred approach to documentation and its principles may not work for hardware documentation. For example, encouraging users to try things out for themselves may be dangerous when operating heavy machinery. As my research concerns software for smartphones, minimalism should be a suitable theoretical background for it.

As I mentioned in the Introduction, the user documentation of mobile applications has been very scarcely studied. Although many usability heuristics for mobile applications exist, I have found only one set of heuristics that explicitly mentions documentation. These heuristics by Yáñez Gómez et al. (2014, 5) include documentation as one of their thirteen primary heuristics. In the first version of their heuristics, there are 158 secondary heuristics under the primary heuristics, 16 of which belong under the primary heuristic of *Help and documentation*. In the final version, there is only one secondary heuristic under *Help and documentation*, but I will compare the 16 heuristics from the first version to minimalism, as these heuristics include several aspects that are worth considering in the user documentation of mobile applications.

The heuristics of Yáñez Gómez et al. (2014) concern, for example, the task structure, the findability of information, and the relevance of the information. The task structure should follow the flow of the actual task, and findability of information decreases the amount of reading required to accomplish tasks. Both are clearly connected to the second principle of minimalism (van der Meij & Carroll 1995, 250), which is concerned with the use of actual tasks and the task structure. The relevance of information is also a central idea in minimalism: the user should be presented with only the information relevant in the particular context.

The ease of simultaneous use of the application and documentation is also a central question in the user documentation of mobile applications. Yáñez Gómez et al (2014, 10) raise these issues in their heuristics. The documentation must be easy to access and the required information must be easy to find in order for the simultaneous use to be effective. The means mentioned by Draper (1998, 366–366), such as clear headings and visual distinction of different kinds of information, can also be utilised in the user documentation of mobile applications. Pictures of the user interface can be used in the documentation to help the user remember the instructions when they return to the task at hand.

The mobile usability attributes gathered by Hoehle and Venkatesh (2015, 442–444) also include some elements with clear connections to minimalism: concise language and user-centred terminology are central to minimalism. Minimalism appears to be a rather good background

against which to analyse documentation of mobile applications: I will re-examine this idea based on my analysis in the Conclusions.

## **Method**

According to Heta Korvenranta (2005, 111–113), heuristic evaluation is a usability evaluation method that consists of a small group of usability experts assessing the usability of a product using a list of heuristics. Heuristics are usability principles, rules, and guidelines: they may be general, or specific to a particular product or category of products, and they can be used both for designing new products and for evaluating existing products. The usability experts review the user interface, noting any violations of the heuristics. Although heuristic evaluation is most effective when the group of experts consists of 3–5 people (ibid., 114), I believe I can form an adequate picture of the usability of the documentation I have selected for evaluation.

One of the most renowned sets of heuristics is a list by Jakob Nielsen and Rolf Molich (1990). Many later sets of heuristics have been strongly influenced by these heuristics (see e.g. Yáñez Gómez et al. 2014). The impact can also be seen in Purho's *10 Recommended Documentation Heuristics* (2000), which I will use in creating my own heuristics.

Purho's (ibid.) heuristics present general ideas about good documentation. There are some similarities between these heuristics and the minimalist heuristics by van der Meij and Carroll (1995): for example, both emphasise recovery from errors. Purho's (ibid.) heuristics include language-related guidelines as well as recommendations on information design, among others. The user should be addressed directly and the language should be familiar to the user; jargon should be avoided. The purposes of different parts of the documentation should be clear and visual elements such as pictures and tables should be used appropriately. Many of Purho's (ibid.) heuristics include elements that are applicable to the user documentation of mobile applications.

The mobile operating systems Android and iOS have some guidelines of their own concerning, for example, the style of language used in applications. Google (Android Developers 2017) has specific guidelines for user documentation. While Apple does not have documentation guidelines, its user interface guidelines (Apple Developers 2017) include some ideas that are relevant to documentation. Google's and Apple's guidelines have some common ground, mainly concerning language. Both emphasise the use of informal language and discourage the use of jargon. There are clear similarities to Purho's (2000) documentation heuristics: the user

should be addressed directly and the use of user interface terminology should be consistent and appropriate to the specific context. Additionally, Google (Android Developers 2017) has some recommendations about the location of the user documentation within the application.

I have created my own set of usability heuristics for user documentation of mobile applications based on the theory, heuristics, and guidelines presented above. The heuristics can be viewed in table 1 below:

*Table 1. Usability Heuristics for User Documentation of Mobile Applications*

<b>Number</b>	<b>Item</b>
<b>1.</b>	<b>Findability of the documentation</b>
<b>2.</b>	<b>Information structure of the documentation</b>
2.1.	Findability of information
2.2.	Modularity of tasks
2.3.	Appropriate use of visual elements
<b>3.</b>	<b>Ease of simultaneous use of documentation and application</b>
<b>4.</b>	<b>Task orientation</b>
<b>5.</b>	<b>Match between documentation and the product</b>
5.1.	Match and consistency of terminology between the documentation and the product
5.2.	Consistency between the user interfaces of the documentation and the application
<b>6.</b>	<b>Error prevention</b>
6.1.	Troubleshooting section for common problems
6.2.	Asking the user for confirmation before executing irreversible or possibly harmful actions
<b>7.</b>	<b>Appropriate use of language in documentation</b>
7.1.	Standard language
7.2.	Addressing the user
7.3.	Conciseness
<b>8.</b>	<b>Meaningfulness of context-sensitive documentation</b>
<b>9.</b>	<b>Following the conventions of the operating system</b>

The first heuristic, *Findability of documentation*, is an elementary one: if the user cannot find the documentation, other qualities of the documentation have no value. According to the

Android guidelines (Android Developers 2017), the documentation should be located within the application – the best practice would be to use contextual help.

The second heuristic is concerned with the information structure of the documentation. When information within the documentation is easy to find, the tasks are independent units, and visual elements such as pictures are used appropriately, the user's cognitive load is decreased and the ease of simultaneous use of the documentation and application – heuristic 3 – is increased. Other factors making the simultaneous use easier is to include the ability to access the documentation directly from anywhere in the application and the ability to return to the same place in the application afterwards.

The fourth heuristic pertains to one of the central principles of minimalism – task orientation. The documentation should focus on real tasks that the user wants to accomplish. The fifth heuristic, *Match between documentation and product*, calls for consistent use of user interface terminology and consistency between the user interface of the application and that of the documentation.

Error recognition and recovery is also one of the key principles of minimalism, and my sixth heuristic concerns this. As I believe the users are likely to only search for help when an error has already happened, the documentation should include a troubleshooting section with solutions to common problems. In order to prevent errors, the application should also ask for confirmation before performing irreversible actions.

The seventh heuristic is concerned with the use of language in the documentation. The language should be standard language familiar to the user, concise, and the user should be addressed directly. Similar elements can be found in heuristic by Purho (2000) and Yáñez Gómez et al. (2015) and the guidelines for Android (Android Developers 2017) and iOS (Apple Developers 2017).

Context-sensitive help that is located in the application's user interface should be meaningful to the user (heuristic 8). That is, the documentation should not give instructions on controls that immediately obvious or system-wide (such as scrolling content), but should focus on the most central features and especially ones that are used with gestures. Finally, the documentation should follow the conventions of the operating system, where those conventions do not violate some other heuristic.

## **Research Data**

The data I will analyse consists of the user documentation of three mobile applications: Pocket (Read It Later 2018), Evernote (Evernote Corporation 2018), and Trello (Trello, Inc 2018). These applications have been downloaded from the Google Play store 10 million, 100 million, and 5 million times, respectively: this makes them relatively popular. The most popular Android applications, many of which come pre-installed in mobile phones, have over a billion downloads: these include, for example, Facebook and Gmail (Google Play 2018).

The applications I have selected for analysis allow the user to save content from the web, create notes, and manage projects, among other things. Pocket's main purpose is saving articles, videos, and other content in one place to be viewed later, even without network connection. Evernote allows the user to create notes and scan documents and share them with others. Trello allows the creation of notes as well, but in a more project-oriented manner, and is well suited for managing projects. For my analysis, I have picked the free versions of these applications: premium versions are also available, and as they include more features, there may also be differences in the documentation compared to the free versions.

I have excluded mobile games, because the user experience of games – player experience – differs from so called utility software (see e.g. Lucero et al. 2013). I also excluded applications I had previously used and those where the documentation was located outside the application, for example on a website. The applications I have selected are relatively close to one another in their intended purposes, and it is interesting to see whether the documentation of these applications is also similar. Next, I will present my most important findings.

## **Analysis**

Although the applications I analysed are rather similar in their purpose, there was wide variation in the design and implementation of documentation. Pocket had both contextual tips and instructions and a separate help view with a rather traditional structure, Evernote's documentation was largely located in the same views as the actual features of the application and included inspirational and motivational content alongside instructions, and Trello's documentation was divided in two: a space where the features of the application are presented and the user can try them out, and a separate space with categorised help topics.

The suitability of my heuristics differed slightly between the applications. This is in large part due to some assumptions I had, such as the lack of surplus space in the user interface of mobile applications and conservative expectations for the information structure of documentation. However, I still found the heuristics rather appropriate for their intended purpose. For example, the heuristics *1. Findability of the documentation*, *3. Ease of simultaneous use of documentation and application*, and *8. Meaningfulness of context-sensitive documentation* proved to be important in practice.

I also made some small changes based on my findings: I removed the heuristic *2.2 Modularity of tasks*, because it does not always seem to apply to units of documentation in mobile applications. I also deemed the heuristic *5. Match between documentation and the product* unnecessary: there is no reason to assume that the user interfaces of the application itself and the documentation would not match, and the consistency of terminology seemed to fit better under heuristic *7. Appropriate use of language in documentation*; therefore, I moved it there. I also gave up the idea of a separate troubleshooting section, as there seems to be few possibilities for errors and it would be ideal to present the error information in the context where the errors occur; this heuristic is now more general, *Providing error information*. The last change I made was removing the heuristic *7.3 Conciseness*. Conciseness is difficult to define as it is very context-specific, and the limited space in user interfaces forces the documentation to be at least relatively concise.

My first research question concerned the characteristics of user documentation of mobile applications. Although the documentation of the three applications was very varied, they had some features in common. Based on my analysis, I have identified the following characteristics:

- **Stylistic consistency with the application's user interface:** the language and visuals of the documentation are consistent with the user interface.
- **Utilising the possibilities provided by the user interface:** making use of, for example, empty space or the application's own tools and features.
- **Standard language:** the language is familiar to the user; jargon is avoided. The language can be relaxed and have humoristic features.
- **Intermingling of different styles:** in addition to instructive text, the documentation can include humoristic, inspiring, or marketing features.
- **Asking for confirmation before executing irreversible actions.**

All these features were present in at least two of the three applications. Although three applications is a rather small set of data, I believe these features are common. Some of them, such as consistency and familiar language, are commonly accepted usability traits of mobile applications (see e.g. Hoehle & Venkatesh 2015).

My second research question pertained to the role of user documentation in creating user experience. It seems that documentation can indeed have an impact on user experience, particularly by decreasing the cognitive load. This can be achieved by introducing the most important features of the application when the user first starts using the application and by making the documentation contextual so that it is always easy to access when needed. Good use of visualisation in the documentation can also help the user to learn and remember what they need to do – this is particularly important when the documentation is not contextual. Other ways to improve user experience include taking the available screen space into consideration: the documentation should not disturb the user (Lakatos 2016), and it can even be used to fill surplus empty space, making the application look better and providing the user with information on what the empty view is for and how to start filling it up (Babich 2017): both Pocket and Evernote took advantage of empty states in this way.

Despite the somewhat small data set and some problems with the suitability of the heuristics, I believe my analysis is rather successful. Analysing a broader set of data in such depth would not have been possible within the constraints of a Master's Thesis. Although the heuristics may have steered my thought processes and some aspects outside the heuristics may have gone unnoticed, the analysis is well balanced and sets a solid foundation for future research.

## **Conclusions**

My research questions concerned the characteristics of user documentation of mobile applications and the role user documentation plays in creating user experience. The documentation I analysed was very varied, but there were some common characteristics, which were mainly centred around the connection between the application's user interface and documentation, and the style of the documentation. Despite the rather small data set, I believe these characteristics are common, as many of them are based on widely accepted usability traits such as consistency. The common characteristics include stylistic consistency between documentation and the user interface, utilising the features and functions of the user interface

in the documentation, using common language, intermingling of different textual styles in the documentation, and asking for confirmation when executing an irreversible action.

Based on my analysis, user documentation of mobile applications does seem to have an impact on the overall user experience of the application. The most prominent way of improving user experience with documentation is decreasing the cognitive load of the user by making the simultaneous use of the application and documentation as effortless as possible and the documentation itself as easily absorbable as possible. Effortlessness and ease of learning address the unique contexts and characteristics of mobile application use such as the susceptibility to interruptions and the shortness of use sessions. User experience can also be improved by making the documentation more agreeable: for example, by using well thought-out language and visuals.

As I expected, minimalism seems to be a good starting point for user documentation of mobile applications. It has clear connections to user experience: findability of information and prioritising the most important information help decrease the strain on the user. Task orientation, conciseness, and layering of information are also aspects of minimalism that are emphasised in documentation of mobile applications.

My data set was rather small, but analysing more than three applications would not have been possible in such depth within the scope of this thesis. However, I believe the data was sufficient. It was rather varied and yielded interesting results. I also believe my choice of method was successful: without existing research or heuristics on the topic, creating a list of heuristics based on a broad theoretical background was a good approach. The main limitation was the lack of other evaluators, as Nielsen (1995b) recommends 3–5 heuristic evaluators, although I do not think this hindered my analysis notably. Some of my heuristics turned out to be difficult to apply and I adjusted the heuristics accordingly so that they would be applicable to broader sets of data.

With a solid theoretical background, I believe my heuristics are well suited for analysis of mobile application documentation and can be used for future research and practical purposes with or without modification. With this thesis, I have set a foundation for this area of research in the field of technical communication. The subject is very topical and I hope to see this research taken further as well as put into practice to produce better user documentation for mobile applications that improves user experience and increases their value for the users.