



Eerika Lehtisyryjä

Strukturoitu havainnointi ohjelmistokehityksessä

Mallin sovittaminen potilastietojärjestelmän
konfigurointiin

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

17.12.2024

Tiivistelmä

Tekijä:	Eerika Lehtisyryjä
Otsikko:	Strukturoitu havainnointi ohjelmistokehityksessä: Mallin sovittaminen potilastietojärjestelmän konfigurointiin
Sivumäärä:	28 sivua + 1 liite
Aika:	17.12.2024
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	
Ohjaajat:	Käytettävyysasiantuntija Inkeri Tynell Osaamisaluejohtaja Janne Salonen

Tämän opinnäytetyön tavoitteena on kehittää strukturoitu havainnointiopas, joka tukee käyttäjäkeskeistä suunnittelua ja käytettävyyden parantamista Apotti - potilastietojärjestelmän kehityksessä. Työ toteutettiin yhteistyössä Oy Apotti Ab:n kanssa, jossa tunnistettiin tarve systemaattisemmalle tavalle kerätä ja hyödyntää kehityskirien aikana tehtyjä käyttäjähavaintoja.

Työn teoreettinen viitekehys pohjautui käyttäjäkeskeisen suunnittelun (UCD), SAFe-viitekehysten ja design sprint -menetelmän periaatteisiin. Havainnointi valittiin keskeiseksi menetelmäksi, koska se mahdollistaa käyttäjien todellisten haasteiden ja tarpeiden ymmärtämisen. Havainnointiopas suunniteltiin ohjaamaan kehitystiimejä havainnoimaan, dokumentoimaan ja analysoimaan kohdevierailujen aikana kerättyä käyttäjätietoa.

Oppaan pilotointi toteutettiin neljässä erillisessä kehityskirissä, joissa kolme sovelluskehittäjää testasivat oppaan käytettävyyttä ja toimivuutta todellisissa käyttötilanteissa. Tulokset osoittivat, että havainnointiopas selkeytti havaintojen keräämistä, paransi niiden laatua ja mahdollisti tiedon tehokkaamman hyödyntämisen kehitysprosessissa. Saadun palautteen perusteella oppaan sisältöä kehitettiin edelleen, muun muassa taulukoiden ja ohjeistusten osalta.

Havainnointioppaan avulla voidaan parantaa ohjelmistokehityksen käytettävyyttä ja käyttäjäkokemusta sekä varmistaa, että kehitys vastaa loppukäyttäjien tarpeita. Oppaan sovellusmahdollisuudet ulottuvat laajemmalle, esimerkiksi muihin ketteriin ohjelmistokehitysprojekteihin, joissa käyttäjälähtöinen suunnittelu on keskeistä.

Avainsanat:	Havainnointi, ohjelmistokehitys, käyttäjäkeskeinen suunnittelu, potilastietojärjestelmä, SAFe-viitekehys, design sprint, järjestelmän konfigurointi, käytettävyys, havainnointiopas, ketterät menetelmät.
-------------	---

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author: Eerika Lehtisyryjä
Title: Structured Observation in Software Development: Adapting the Model to Patient Record System Configuration
Number of Pages: 28 pages + 1 appendices
Date: 17 December 2024

Degree: Bachelor of Engineering
Degree Programme: Information and Communication Technology
Professional Major:
Supervisors: Inkeri Tynell, Usability Specialist
Janne Salonen, Director of school

The objective of this thesis is to develop a structured observation guide that supports user-centered design and improves usability in the development of the Apotti patient information system. The work was carried out in collaboration with Oy Apotti Ab, where a need was identified for a more systematic approach to collecting and utilizing user observations made during development sprints.

The theoretical framework of the study is based on the principles of User-Centered Design (UCD), the SAFe framework, and the design sprint methodology. Observation was chosen as the key method because it enables a deeper understanding of users' real challenges and needs. The observation guide was designed to instruct development teams on observing, documenting, and analyzing user data collected during site visits.

The guide was piloted in four separate development sprints, during which three software developers tested its usability and functionality in real-world scenarios. The results showed that the observation guide clarified the process of collecting observations, improved their quality, and enabled more efficient utilization of the data in the development process. Based on the feedback received, the content of the guide was further refined, particularly regarding tables and instructions.

The observation guide can enhance the usability and user experience of software development and ensure that development aligns with end-user needs. The guide's applicability extends more broadly to other agile software development projects where user-centered design is a central focus.

Keywords: Observation, software development, user-centered design, patient information system, SAFe framework, design sprint, system configuration, usability, observation guide, agile methods.

The originality of this thesis has been checked using the Turnitin Originality Check software.

Sisällys

Lyhenteet

1	Johdanto	6
2	Teoreettinen tausta	7
2.1	SAFe-viitekehys	7
2.2	Design sprint -menetelmä	8
2.3	Tilaaajaorganisaatio ja kehityskirimalli	10
3	Käytettävyys ja käyttäjäkeskeiset menetelmät	11
3.1	Käyttäjäkeskeinen suunnittelu	11
3.2	Käytettävyys	12
3.3	Käytettävyystestaus	14
4	Havainnointi	15
4.1	Havainnointiprosessi	15
4.2	Havainnoinnin merkitys	16
5	Havainnointioppaan suunnittelu	17
5.1	Havainnointioppaan tavoite ja tarve	17
5.2	Oppaan sisältö ja rakenne	18
5.3	Pilotointi	19
5.4	Palaute ja jatkokehitys	20
6	Pohdinta ja johtopäätökset	22
6.1	Opinnäytetyön tulokset	22
6.2	Suositukset jatkokehitykselle	23
6.3	Opinnäytetyön merkitys ja sovellusmahdollisuudet	24
	Lähteet	25

Liitteet

Liite 1: Havainnointiopas

Lyhenteet

SAFe: Scaled Agile Framework. Suomeksi skaalattu ketterä viitekehys.

ART: Agile Release Train. Suomeksi ketterä kehitysjuona.

UCD: User-Centered Design. Suomeksi käyttäjäkeskeinen suunnittelu.

HCI: Human-Centered Interaction. Suomeksi ihmisen ja tietokoneen vuorovaikutus.

GV: Google Ventures. Design sprint -menetelmän kehittäjäorganisaatio.

ISO: International Organization for Standardization. Suomeksi Kansainvälinen standardisoimisjärjestö.

MoSCoW: Must have, should have, could have, won't have. Suomeksi esimerkiksi pakolliset, tärkeät, hyödylliset ja ei toteuteta.

1 Johdanto

Tämän opinnäytetyön taustana on design sprint -menetelmän hyödyntäminen potilastietojärjestelmän kehityksessä opinnäytetyön tilaajaorganisaatiossa Oy Apotti Ab:ssa. Opinnäytetyö on suunnattu erityisesti design sprintin aikana toteutettavien kohdevierailujen ja niistä nousevien käytettävyyshavaintojen tehokkaaseen integrointiin järjestelmän konfiguraatioprosessiin. Apotissa design sprintit on nimetty kehityskireiksi. Näiden kehityskirien aikana tehtävät kohdevierailut ovat keskeisiä käyttäjäkokemuksen ymmärtämisessä ja järjestelmän käytettävyyden parantamisessa.

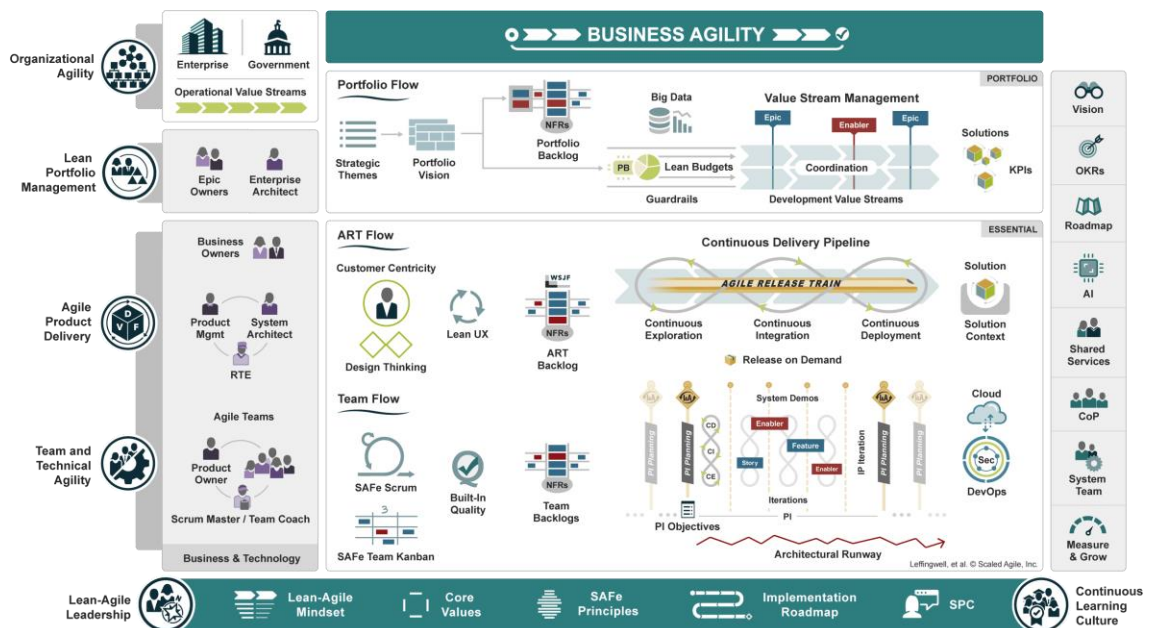
Tämän opinnäytetyön taustalla on tunnistettu selkeä tarve kehittää parempi prosessi kohdevierailuissa nousseiden havaintojen hyödyntämiseksi. Nykyinen tapa, jolla kehityskirien aikaisia havaintoja kerätään ja jäsenellään, johtaa usein siihen, että havaintojen hyödyntäminen kehitysprosessissa jää puutteelliseksi. Tavoitteena on luoda käyttäjäkeskeisen suunnittelun periaatteiden ja SAFe-viitekehityksen avulla prosessi, joka mahdollistaa havaintojen tehokkaamman käsittelyn ja integroinnin kehitystyöhön. Tämä prosessi ei ainoastaan tehostaisi havaintojen keräämistä ja järjestelyä, vaan myös varmistaisi niiden sujuvan soveltamisen kehitystyössä, mikä parantaisi lopputuloksena olevan järjestelmän käytettävyyttä ja vastaavuutta käyttäjien tarpeisiin.

Tämä opinnäytetyö pyrkii täyttämään Apotissa tunnistetun tarpeen kehittämällä ohjeellisen mallin eli havainnointioppaan, joka ohjaa sovelluskehitystiimejä keräämään kohdevierailujen aikana havainnoiteja sekä integroimaan havainnot kehitystyöhön SAFe-mallin mukaisesti. Mallin kehittäminen ja testaaminen tapahtuu käytännössä osana kehityskirejä, jolloin voidaan varmistaa, että tulokset ovat sekä teoreettisesti päteviä että käytännöllisesti sovellettavissa. Tämän opinnäytetyön tavoitteena on tuoda konkreettista hyötyä Apotin kehitystyöhön ja sitä kautta parantaa potilastietojärjestelmän käytettävyyttä.

2 Teoreettinen tausta

2.1 SAFe-viitekehys

Scaled Agile Framework (SAFe) on viitekehys, joka on kehitetty vastaamaan suurten organisaatioiden ja monimutkaisten projektien tai tuotteiden tarpeita yhdistämällä johtamisfilosofia Leanin ja ketterän kehityksen Agilen käsitteitä. SAFe -viitekehityksen keskiössä on ajattelutapa, että vaikka perinteiset ketterät menetelmät, kuten esimerkiksi Scrum, ovat tehokkaita pienissä ryhmissä niin isommissa kokonaisuuksissa tarvitaan monipuolisempaa lähestymistapaa eli skaalausta. SAFe kokoaa eri ketterän kehityksen tiimit yhteen Agile Release Trainin (ART) eli kehitysjunien kautta. Nämä kehitysjunat ylittävät toiminnallisia rajoja, vähentävät turhaa työnsiirtoa ja pyrkivät näin nopeuttamaan arvon toimitusta asiakkaille perustuen ketterän kehityksen periaatteisiin. [1.] Kuva 1 havainnollistaa SAFe-viitekehystä portfoliolaajuudessa.



Kuva 1. SAFe 6.0 [1.]

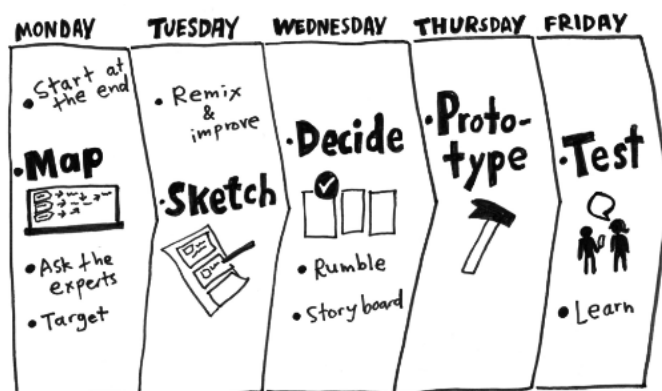
SAFe tarjoaa mallin suurten organisaatioiden skaalautuvalle ketterälle toiminnalle ja se kattaa sekä kehitystiimitason käytännöt että strategiatason suunnittelun. SAFe yhdistää nämä periaatteet neljään viitekehystasoon, jotka

kattavat koko organisaation ja ohjaavat näin toimintaa. Tiimitaso (team level) on ensimmäinen SAFe-viitekehityksen kerros, jossa ketterät kehitystiimit keskittyvät tuotekehitykseen iteratiivisesti tuotekehitysjaksoittain eli sprintsinä ja inkrementaalisesti. Seuraavalla ohjelmatasolla (program level) useat ketterän kehityksen tiimit toimivat yhteistyössä tuottaakseen suuria arvoja ja toimintoja. Suurten ratkaisujen (large solution) tasolla kehitetään suuria ja monimutkaisia ratkaisuja, jotka vaativat useiden ohjelmien yhteistyötä. Viimeisellä tasolla eli portfoliotasolla keskitytään strategiaan tavoitteisiin ja siihen, miten investoinnit ohjelmistokehitykseen linjautuvat yrityksen yleisten tavoitteiden kanssa. [1.]

2.2 Design sprint -menetelmä

Design sprint on menetelmä, joka on kehitetty nopeuttamaan tuote- ja palvelukehitystä keskittymällä suoraan käyttäjien tarpeisiin ja haasteisiin. Jake Knappin suunnittelema design sprint menetelmä on tullut Google Ventures (GV) yhtiön kautta laajempaan käyttöön. Design sprint yhdistää elementtejä suunnitteluprosessista ja tieteellisistä menetelmistä kooten ne yhteen ketterän filosofian avulla. Ydintavoitteena on tuottaa nopeita, tehokkaita ratkaisuja tiivistetyssä ajassa. [2.]

Design sprint -prosessi on tyypillisesti jaettu viiteen vaiheeseen, jossa kullakin vaiheella on omat erityiset tavoitteensa ja toimintansa. Design sprintin eri vaiheet havainnollistuu kuvassa 2.



Kuva 2. Design sprintin vaiheet [3.]

Ensimmäisessä vaiheessa kehitystiimi keskittyy ymmärtämään ongelman perusteellisesti ja valitsemaan kehityskohteen. Tämä vaihe sisältää usein erilaisia käyttäjätutkimuksia kuten asiakashaastatteluita, käyttäjäkuvauksen luomista ja nykyisen tilanteen analysointia. Toisessa design sprintin vaiheessa fokus siirtyy ideoiden luomiseen ja innovaatioiden kehittämiseen. Tällöin tavoitteena on luoda mahdollisimman laaja valikoima mahdollisia ratkaisuja ratkaistavaan ongelmaan. Ratkaisujen innovointiin voidaan käyttää apuna esimerkiksi ajatuskarttoja ja kuvakäsikirjoitusta. Kolmannessa vaiheessa tiimi arvioi ideat ja valitsee lupaavimman konseptin jatkokehitykseen, usein äänestyksen ja keskustelun kautta. [4.]

Neljännessä design sprintin vaiheessa valittu jatkokehittävä idea tai ratkaisu muuttuu toimivaksi prototyypiksi. Prototyyppi on yksinkertaistettu versio lopullisesta tuotteesta tai palvelusta, joka sisältää vain oleelliseen sisältöön. Viidennessä ja viimeisessä vaiheessa kehitetty prototyyppi testataan asiakkaiden ja käyttäjien kanssa. Asiakkaalta saatu palaute antaa arvokasta tietoa siitä, miten tuote tai palvelu vastaa todellisiin tarpeisiin ja odotuksiin, jolloin hyödynnetään jatkuvaa oppimista sekä kehitystä. [4.]

Design sprintin suurin hyöty on sen kyky tarjota tehokas tapa ratkaista ongelmia, testata uusia ideoita ja sitä kautta saavuttaa tuloksia nopeasti. Menetelmä auttaa tiimejä keskittymään olennaiseen, välttämään turhaa työtä ja tekemään perusteltuja päätöksiä nopeasti. Vaikka viiden päivän malli on yleisin, design sprint -menetelmää voidaan mukauttaa ja soveltaa monenlaisiin projekteihin ja organisaatioihin, mikä tekee siitä joustavan ja skaalautuvan työkalun niin pienten startup-yritysten kuin suurtenkin yritysten tuotekehityksessä. [5.]

Design sprint tarjoaa ketterän kehityksen tiimeille strukturoidun ja fokusoituneen tavan lähestyä monimutkaisia haasteita, edistää innovatiivista ajattelua ja johtaa kehitystä nopeasti konkreettisiin ja testattuihin ratkaisuihin. Design sprint -prosessin tavoitteena on nopeuttaa tiimin oppimista ratkaisun kehittämisessä, tuottaen näin jotain, mikä on hyvin lähellä sitä, mitä loppukäyttäjä todella

haluaa, ja siten vähentäen muutoksia, jotka olisivat tarpeen ketterän kehitysprosessin aikana. [6.]

Uusi prosessi, nimeltään käyttäjakeskeinen design sprint (UCD Sprint), kehitettiin vuonna 2021 HCI-tutkijoiden puolesta. Tämä prosessi yhdistää käyttäjakeskeisen suunnittelun menetelmiä perinteisen design sprintin rakenteeseen ja sen erityinen painopiste on käyttäjien varhaisessa ja aktiivisessa osallistamisessa. UCD Sprintin avulla voidaan keskittyä konseptisuunnitteluun siten, että suunnitteluprosessissa huomioidaan moninaiset käyttäjäryhmät. Prosessi sisältää eri vaiheita, kuten perinteisessä design sprintissä. Tämä prosessin vaiheistus varmistaa, että käyttäjien tarpeet integroidaan suunnitteluun heti alusta alkaen. Tämä vuoksi UCD Sprintti on inklusiivinen ja nopea tapa luoda käyttäjakeskeisiä ratkaisuja jo ohjelmistokehityksen alkuvaiheessa. [7.]

2.3 Tilaajaorganisaatio ja kehityskirimalli

Oy Apotti Ab tuottaa asiakkailleen Apotti -sosiaali- ja terveydenhuollon asiakas- ja potilastietojärjestelmää. Apotti-järjestelmä yhdistää perusterveydenhuollon, erikoissairaanhoidon ja sosiaalihuollon palvelut yhteen kattavaan potilastietojärjestelmään. Apotti-järjestelmän myötä pääkaupunkiseudun sadat erilaiset tietojärjestelmät pystyttiin vaihtamaan yhteen ja samalla saatiin sosiaali- ja terveydenhuollon tiedot samaan järjestelmään. Apotti-järjestelmä on käyttöön otettu HUSin alueella sekä Helsingissä, Kauniaisissa, Vantaalla ja Keravan hyvinvointialueilla. Apotti-potilastietojärjestelmää käyttää päivittäin yli 47 000 ammattilaista. [8.]

Vuonna 2021 aloitetun Apotti 2.0 käytettävyyttä parantavan ohjelman aikana kehitettiin kehityskirimalli, jonka pilottivaiheen tulokset olivat onnistuneita, joten kehityskirimalli on otettu pysyvään käyttöön Apotin järjestelmäkehitysprosessissa Apotti 2.0 -ohjelman päätyttyä vuonna 2023. Apotin kehityskirimallin tarkoitus on lisätä Apotin sovelluskehityksen ja loppukäyttäjien yhteistyötä ja yhteisymmärrystä. Kehityskirimallin keskiössä on

käytettävyyden parantaminen ja käyttäjäkeskeisen suunnittelun tehokas implementointi. Kehityskirimalliin kuuluu kohdevierailu, jossa Apotin sovelluskehitys sekä loppukäyttäjät voivat käydä suoraa dialogia järjestelmästä ja esimerkiksi sen käytettävyydestä. Kohdevierailun aikana usein myös havainnoidaan järjestelmän käyttöä. [9.]

Kehityskireissä on vahvasti mukana design sprint -menetelmän ydinidea: tuotetaan nopeita ja tehokkaita ratkaisuja tiivistetyssä ajassa.

Kehityskirimallissa asiakasorganisaatioyksikkö valitsee ennaltamääriteltä tarjottimilta kehityskohteita, joiden jatkokehitykseen keskitytään kehityskirin aikana. Lyhyen kehityskirijakson loppuun mennessä asiakkaan toivomat muutokset ovat toteutettu määrittelyjen mukaisesti Apotin sovelluskehitystiimin toimesta. Kehityskirimallin mahdollistama nopeutettu järjestelmänkehitysprosessi lisää käyttäjätyytyväisyyttä sekä tuottaa suoraa asiakasarvoa. [10.]

3 Käytettävyys ja käyttäjäkeskeiset menetelmät

3.1 Käyttäjäkeskeinen suunnittelu

Käyttäjäkeskeinen suunnittelu (User-Centered Design eli UCD) on prosessi, joka asettaa käyttäjät ja heidän kokemuksensa keskiöön tuotteiden, palveluiden tai järjestelmien sekä suunnittelussa että kehityksessä. UCD:n perusajatus on, että tuotteen kehitysprosessi pohjautuu tietoon ihmisistä, jotka tulevat käyttämään tuotetta. Tavoitteena on luoda ratkaisuja, jotka eivät ainoastaan täytä käyttäjien tarpeita vaan myös tarjoavat positiivisia käyttökokemuksia. [11.]

Käyttäjäkeskeisen suunnittelun keskeisimpiä periaatteita ovat aikainen keskittyminen käyttäjiin ja tehtäviin, empiirinen mittaus ja iteratiivinen kehitys. Varhaisessa vaiheessa suunnitteluprosessia tiimi suorittaa käyttäjätutkimuksia suorassa käyttäjäkontaktissa ja pyrkii sitä kautta saavuttamaan käyttäjäymmärryksen. Empiirisellä mittauksella tarkoitetaan käyttäjien kanssa suoritettavaa käytettävyydestausta, jonka metodeja ovat esimerkiksi kyselyt,

A/B-testaus tai havainnointi. Iteratiivisuus tarkoittaa jatkuvaa kehitysprosessia. Iteratiivisessa kehityksessä käyttäjätutkimusten ja/tai käytettävyystestauksen kautta löytyneitä havaintoja pyritään korjaamaan ja testaamaan uudelleen. Iteratiivinen kehitys mahdollistaa suunnittelun joustavuuden ja muokattavuuden käyttäjäpalautteen perusteella, mikä johtaa lopulta käyttäjäystävällisempiin ja toimivampiin lopputuloksiin. [12.]

Käyttäjakeskeinen suunnittelu ja käytettävyys ovat tiiviisti kytköksissä toisiinsa, sillä molempien tavoitteena on varmistaa, että kehitetyt tuotteet tai järjestelmät vastaavat käyttäjien tarpeisiin ja ovat helppokäyttöisiä. UCD periaatteiden mukaisesti käyttäjät otetaan mukaan suunnitteluprosessin alkuvaiheista lähtien, jolloin heidän tarpeensa ja odotuksensa vaikuttavat suoraan tuotteen suunnitteluun. Tämä vähentää virheitä ja parantaa tuotteen käytettävyyttä, mikä näkyy tehokkaampana ja miellyttävämpänä käyttökokemuksena. Käytettävyys viittaa siihen, kuinka helposti ja tehokkaasti käyttäjät voivat suorittaa tehtäviään tuotteella tai palvelulla. UCD:n iteratiivinen prosessi varmistaa, että käyttäjätestaukset tehdään jokaisessa kehitysvaiheessa, jolloin tuotteen käytettävyyttä voidaan jatkuvasti parantaa. Tämä lähestymistapa vähentää kehityskustannuksia ja varmistaa, että tuote vastaa todellisia käyttäjätarpeita ja täyttää mm. käytettävyysstandardeja. [13.]

3.2 Käytettävyys

Käytettävyys (usability) on keskeinen osa käyttäjakeskeistä suunnittelua ja määrittelee, kuinka hyvin käyttäjät pystyvät käyttämään tuotetta tai palvelua tehokkaasti, tehokkaasti ja tyytyväisesti. ISO 9241-11 -standardin mukaan käytettävyys koostuu kolmesta pääkomponentista: tehokkuus (efficiency), vaikuttavuus (effectiveness) ja tyytyväisyys (satisfaction). [14.]

Tehokkuus tarkoittaa sitä, kuinka nopeasti ja resurssitehokkaasti käyttäjät pystyvät suorittamaan tehtäviä järjestelmää käyttäen. Tehokkuutta voidaan mitata esimerkiksi ajalla, jonka käyttäjät tarvitsevat tehtävän suorittamiseen, tai tehtävän aikana käytettyjen resurssien määrällä. Järjestelmiä, joita käytetään

jatkuvasti tai toistuvasti, on tärkeää optimoida käyttäjän ajan ja resurssien säästämiseksi. Esimerkiksi työkalut tai sovellukset, joita käytetään päivittäin, vaativat tehokasta käyttöliittymää ja suorituskykyä, jotta ne eivät hidasta käyttäjän toimintaa. Tällaisissa tapauksissa tehokkuus korostuu, koska pienetkin hidasteet voivat pitkällä aikavälillä aiheuttaa merkittävää ajanhukkaa tai lisätä työkuormaa. [15.]

Vaikuttavuus viittaa taas siihen, kuinka tarkasti ja virheettömästi käyttäjät saavuttavat tavoitteensa järjestelmää käyttäessään. Vaikuttavuus mittaa sitä, kuinka hyvin käyttäjät pystyvät suorittamaan tehtäviä ilman merkittäviä virheitä ja saavuttamaan halutut tulokset, mikä parantaa järjestelmän käytettävyyttä ja luotettavuutta. Vaikuttavuutta voidaan mitata sen perusteella, kuinka tarkasti ja virheettömästi käyttäjät suorittavat tehtävänsä ilman merkittäviä virheitä. Mittaustapoihin sisältyy esimerkiksi tehtävän suorittamisen onnistumisprosentti ja tehtävien aikana havaittujen virheiden määrä. Vaikuttavuuden mittaaminen on erityisen tärkeää järjestelmissä, joissa tarkkuus ja virheettömyys ovat keskeisiä toiminnan luotettavuuden kannalta, kuten esimerkiksi potilastietojärjestelmissä. [16.]

Tyytyväisyys tarkoittaa käyttäjän omaa subjektiivista kokemusta siitä, kuinka mukavalta ja miellyttävältä järjestelmän käyttö tuntuu. Käyttötyytyväisyys heijastaa sitä, että vastaako järjestelmä käyttäjän odotuksia ja tarpeita. ISO 9241-11 -standardin mukaan tyytyväisyys on tärkeä osa käytettävyyttä, sillä se kannustaa käyttäjiä käyttämään järjestelmää toistuvasti ja tukee sen menestystä pitkällä aikavälillä. Tyytyväisyyttä mitataan yleensä kyselylomakkeilla tai arviointiskaaloilla. Tyypillisiä mittareita ovat Likert-asteikot, joissa käyttäjät arvioivat tyytyväisyyttään esimerkiksi 1–5 asteikolla. Kyselylomakkeet voivat sisältää kysymyksiä, kuten esimerkiksi "Kuinka helppokäyttöiseksi koet järjestelmän?" [17.]

3.3 Käytettävyytestaus

Käytettävyytestaus on keskeinen vaihe käyttäjäkeskeisessä suunnittelussa, ja sen avulla varmistetaan, että järjestelmä toimii käyttäjän näkökulmasta sujuvasti ja vastaa heidän tarpeisiinsa. Testauksen päätavoite on tunnistaa käytettävyyssongelmat jo ennen tuotteen julkaisua, jolloin kehitystiimi voi korjata puutteet ja parantaa käyttäjäkokemusta ennen laajempaa käyttöönottoa. Tämä ennakoiva lähestymistapa auttaa vähentämään jälkikäteen tarvittavia päivityksiä ja niihin liittyviä kustannuksia. [18.]

Laborioritestaus on yksi käytetyimmistä käytettävyytestauksen menetelmistä. Se suoritetaan kontrolloidussa ympäristössä, jossa tutkijat seuraavat käyttäjien toimintaa tarkasti. Tällainen testaus tarjoaa arvokasta tietoa esimerkiksi käyttöliittymän intuitiivisuudesta ja siitä, kuinka hyvin käyttäjät suoriutuvat tehtävistä. Laborioritestauksen etu on sen tarkkuus ja syvälinen analyysi, jonka avulla voidaan tunnistaa pienetkin käytettävyyssongelmat, jotka voivat vaikuttaa käyttäjän kokemukseen. [19.]

Kenttätestauksessa käyttäjät testaavat järjestelmää omassa ympäristössään, mikä tuo esiin todelliset käyttötilanteet ja mahdolliset arjen haasteet. Tämä menetelmä on erityisen hyödyllinen tuotteille, joita käytetään monimuotoisissa ympäristöissä, kuten mobiilisovelluksissa. Kenttätestaus auttaa kehittäjiä ymmärtämään, miten eri käyttöolosuhteet voivat vaikuttaa järjestelmän toimivuuteen ja käytettävyyteen. [20.]

Moderoitu testaus mahdollistaa tarkemman vuorovaikutuksen käyttäjän kanssa. Testaaja voi esittää kysymyksiä ja pyytää selvennyksiä, mikä antaa syvempää tietoa käyttäjän ajattelutavoista ja mahdollisista haasteista, joita käyttäjä kohtaa. Tämä lähestymistapa sopii erityisesti monimutkaisten järjestelmien testaukseen, joissa käyttäjän perusteellinen ymmärrys on tärkeää. [17.]

Moderoimaton testaus mahdollistaa käyttäjien itsenäisen toiminnan ilman tutkijan vaikutusta. Tämä menetelmä tarjoaa arvokasta tietoa käyttäjien luonnollisesta toiminnasta ja siitä, kuinka hyvin järjestelmä toimii intuitiivisesti

ilman ohjausta. Moderoimaton testaus on kustannustehokas tapa kerätä kattavaa palautetta ja soveltuu hyvin laajoihin tutkimuksiin, esimerkiksi verkkosovellusten käytettävyydestä. [21.]

4 Havainnointi

4.1 Havainnointiprosessi

Käyttäjäkeskeisen suunnittelun ja tuotteen käytettävyyden varmentamisen keskeisin menetelmä on käytettävyydestä. Yksi käytettävyydestä menetelmistä on havainnointi ja sen juuret ovat etnografiassa eli kansan ja kulttuurin tutkimuksessa. Havainnointi auttaa kehittäjiä havaitsemaan käyttäjien kohtaamat haasteet ja heidän toimintatapansa. Erityisesti monimutkaisissa tai muuttuvissa ympäristöissä havainnointi on arvokas menetelmä käyttäjävälisyyden parantamiseksi ja aidosti toimivan järjestelmän suunnittelun tukemiseksi. [22.] Havainnointi voidaan toteuttaa monella eri tavalla, ja menetelmän valinta riippuu tutkimuksen tavoitteista sekä käyttöympäristön vaatimuksista.

Suora havainnointi on käytettävyydetutkimuksessa yleinen menetelmä, jossa tutkija seuraa käyttäjän toimintaa luonnollisessa käyttöympäristössä ilman ennalta määriteltyjä havaintokriteerejä. Tässä menetelmässä tutkija tarkkailee käyttäjien spontaania vuorovaikutusta sovelluksen kanssa ja kirjaa ylös huomioita, kuten käyttäjän ilmeet, reaktiot ja haasteet eri toiminnoissa. Suoran havainnoinnin etuna on sen joustavuus ja kyky paljastaa odottamattomia käytettävyyso ongelmia, jotka eivät välttämättä ilmenisi strukturoitujen menetelmien avulla. Lisäksi suora havainnointi mahdollistaa syvemmän ymmärryksen käyttäjän tuntemuksista ja ajattelutavoista, mikä voi olla arvokasta sovelluskehityksessä käyttäjäkokemuksen parantamiseksi. [23.]

Strukturoitu havainnointi on tutkimusmenetelmä, jossa tutkija määrittelee tarkasti havainnoitavat kohteet ja tilanteet etukäteen.

Käytettävyydetutkimuksessa strukturoitu havainnointi auttaa keskittymään

tiettyihin käyttöliittymän osiin tai käyttäjän toimiin, jolloin voidaan vertailla käyttäjien toimintamalleja systemaattisesti ja objektiivisesti. Tämä menetelmä soveltuu erityisesti tilanteisiin, joissa halutaan kerätä määrällistä eli kvantitatiivista dataa. Strukturoitu havainnointi sopii myös tilanteisiin, joissa esimerkiksi halutaan verrata uuden kehityksen vaikutusta ohjelman käyttöön. [24.]

Osallistuvassa havainnoinnissa tutkija osallistuu itse sovelluksen käyttöön tai testaukseen havainnoitavan käyttäjän rinnalla, jotta tutkija ymmärtää käyttökokemuksen syvällisesti. Tässä lähestymistavassa tutkija ei pelkää tarkkailla käyttäjää, vaan voi myös esittää heille kysymyksiä tai ohjeita, mikä auttaa saamaan syvempää tietoa käyttäjän toiminnan taustoista ja mahdollisista esteistä käyttötilanteissa. Sovelluskehityksessä osallistuva havainnointi mahdollistaa myös käyttäjien palauteprosessin parantamisen ja sovelluksen kehittämisen iteratiivisesti käyttäjäkokemusten pohjalta. [25.]

4.2 Havainnoinnin merkitys

Havainnoinnin erityinen merkitys korostuu dynaamisissa ja monimutkaisissa käyttöympäristöissä, joissa käyttäjien toiminta voi vaihdella tilanteen mukaan. Tässä yhteydessä havainnointi ei pelkää dokumentoida käyttäjien toimintaa, vaan myös auttaa analysoimaan heidän päätöksentekoprosessejaan ja tunteitaan. Tämä johtaa parempaan käyttäjäkokemukseen ja mahdollistaa jatkuvan parantamisen prosessin, mikä on erityisen tärkeää iteratiivisessa tuotekehityksessä. [26.]

Havainnointi tukee käyttäjäkeskeistä suunnittelua, koska sen avulla voidaan kerätä sekä laadullista että määrällistä dataa. Laadullinen data, kuten käyttäjien ilmeet, tunteet ja spontaanit reaktiot, paljastaa syvempiä havaintoja, kun taas määrällinen data mahdollistaa objektiivisemmän vertailun eri käyttötapojen välillä. Molemmat lähestymistavat täydentävät toisiaan ja muodostavat kattavan kokonais kuvan käytettävyysongelmista. [27.]

Havainnoinnin avulla voidaan varmistaa, että suunnitteluprosessi perustuu todellisiin käyttäjäkokemuksiin eikä vain teoreettisiin oletuksiin. Tämä johtaa paremmin toimivien ja käytettävyydeltään tehokkaiden järjestelmien kehittämiseen, jotka vastaavat käyttäjien tarpeita ja odotuksia. Havainnoinnin avulla voidaan myös tunnistaa ongelmia, jotka eivät ilmene muilla menetelmillä, kuten haastatteluilla tai kyselyillä. [28.]

Havainnointi on todella arvokas menetelmä käyttäjäkeskeisessä suunnittelussa, koska se tuo esiin sekä näkyviä että piileviä haasteita ja tarjoaa kattavaa käyttäjäymmärrystä. Tämä mahdollistaa käyttäjien tarpeisiin paremmin vastaavien ja käyttöympäristöön sopivien järjestelmien kehittämisen. Havainnointi tarjoaa elintärkeää tietoa sovelluskehitykselle, jonka avulla voidaan parantaa käytettävyyttä ja käyttäjäkokemusta entisestään. [29.]

5 Havainnointioppaan suunnittelu

5.1 Havainnointioppaan tavoite ja tarve

Havainnointioppaan tavoitteena on tukea kehityskirin aikana tapahtuvaa kohdevierailua järjestelmällisellä ja strukturoidulla tavalla, joka mahdollistaa tehokkaan ja laadukkaan käyttäjähavaintojen keräämisen. Oppaan avulla sovelluskehittäjät voivat ymmärtää paremmin käyttäjien jokapäiväistä työympäristöä, työnkulkuja sekä heidän vuorovaikutustaan järjestelmän kanssa. Tämä tieto on keskeistä käyttäjäkeskeisen suunnittelun periaatteiden toteuttamisessa ja järjestelmän jatkuvassa parantamisessa.

Tarve havainnointioppaalle nousee suoraan käyttäjäymmärryksen ja käytettävyydestestauksen keskeisestä roolista järjestelmäkehityksessä. Apotti-järjestelmän kaltaisen monimutkaisen potilastietojärjestelmän käytettävyyden parantaminen vaatii syvällistä tietoa siitä, miten loppukäyttäjät järjestelmää hyödyntävät ja millaisia haasteita he kohtaavat sen käytössä. Nykyisissä kehityskireissä kohdevierailut ovat olleet tärkeä osa käyttäjäymmärryksen luomista, mutta ilman selkeitä ohjeita ja rakenteita havaintojen kerääminen on

ollut laadullisesti vaihtelevaa ja vaikeasti integroitavissa sovelluskehitysprosessiin. [30. & 9.]

Havainnointiopas vastaa strategian mukaiseen tarpeeseen tarjoamalla selkeät ohjeet ja työkalut laadukkaaseen käyttäjähavainnointiin. Oppaan systemaattinen lähestymistapa varmistaa, että havainnot ovat kattavia ja käytettävissä päätöksenteossa sekä järjestelmän jatkokehityksessä SAFe-viitekehyksen mukaisesti. Havainnointiopas suunnitellaan pääasiassa Apotin sovelluskehittäjille, jotka osallistuvat kehityskirin aikana järjestettävään kohdevierailuun. Havainnointiopas soveltuu kuitenkin yleislaatusesti kaikkiin tilanteisiin, joissa on mahdollisuus päästä havainnoimaan loppukäyttäjää ja kaikille käyttäjäryhmille.

5.2 Oppaan sisältö ja rakenne

Havainnointiopas on suunniteltu vastaamaan käyttäjäkeskeisen suunnittelun tarpeisiin sekä teoreettisen tiedon että käytännön tavoitteiden pohjalta. Opas sisältää teoreettisen johdannon, keskeiset havainnointiohjeet, käytännön työskentelytavat sekä strukturoituja lomakkeita havaintojen keräämiseksi ja analysoimiseksi. Havainnointioppaan sisällön suunnittelussa on käytetty tukena Sampsa Hyysalon kirjaa Käyttäjä tuotekehityksessä [31].

Havainnointiprosessi alkaa perusteellisella valmistautumisvaiheella, jonka aikana havainnoija tutustuu tarkasti käyttäjäryhmiin, heidän tyypillisiin työtehtäviinsä ja työnkulkuihinsa. Vierailtavan organisaation toiminnan ja potilasprofiilien ymmärtäminen on välttämätöntä, jotta havainnoija voi optimoida havaintostrategiansa ja saavuttaa kattavan ymmärryksen siitä, kuinka loppukäyttäjät hyödyntävät Apotti-järjestelmää päivittäisessä työssään.

Oppaassa esitellään kaksi havainnointimenetelmää: aktiivinen havainnointihaastattelu ja passiivinen havainnointi. Aktiivisessa havainnointihaastattelussa havainnoija osallistuu suoraan työprosessiin, tarkkailee toimintaa ja esittää selventäviä kysymyksiä toiminnan

ymmärtämiseksi syvällisemmin. Passiivinen havainnointi keskittyy sivusta seuraamiseen ja muistiinpanojen tekemiseen, jolloin tarkentavat kysymykset voidaan kysyä havainnointijakson jälkeen.

Muistiinpanojen kirjaamisessa korostuu objektiivisuuden ja tarkkuuden merkitys. Havainnoijan tulee dokumentoida havainnot yksityiskohtaisesti, käyttäen hyväkseen lyhenteitä ja symboleja tehokkuuden lisäämiseksi. Systemaattinen muistiinpanojen tekeminen tukee myöhemmin tapahtuvaa analysointivaihetta, jossa kaikki kerätyt tiedot siirretään digitaaliseen muotoon analyysia ja jatkokehitystä varten.

Analyysivaiheessa havainnot järjestellään ja teemoitellaan järjestelmän eri osa-alueiden mukaan. Tässä vaiheessa käytetään MoSCoW-menetelmää havaintojen priorisointiin, joka auttaa arvioimaan niiden merkitystä käyttäjäkokemuksen, laadun ja potilasturvallisuuden kannalta. Tämän jälkeen havainnot siirretään SAFe-viitekehyksen mukaiseen kehityspotkeen, mikä varmistaa, että havainnoista johdetut toimenpiteet saadaan tehokkaasti ja oikein vietyä käytäntöön.

5.3 Pilotointi

Havainnointioppaan testauksen toteutus ja palautteiden keräys ovat kriittisiä vaiheita oppaan kehitysprosessissa. Testaustavaksi valikoitui pilotointi, jossa opasta kokeillaan todellisissa havainnointitilanteissa. Pilotointi on keskeinen osa havainnointioppaan testausprosessia, joka mahdollistaa teoreettisten olettamusten ja käytännön toteutuksen kohtaamisen. Käyttäjätestauksessa havainnointioppaan kohderyhmä, eli sovelluskehittäjät, käyttävät opasta ohjatusti ja antavat palautetta sen käytettävyydestä, ymmärrettävyydestä ja hyödyllisyydestä. Tämä menetelmä auttaa tunnistamaan mahdollisia puutteita oppaassa ja tarjoaa konkreettista tietoa siitä, miten sitä voidaan parantaa.

Pilotointiin osallistui kolme sovelluskehittäjää, jotka käyttivät havainnointioppasta neljän eri kehityskirin aikana järjestetyillä kohdevierailuilla. Havainnointioppaat

tulostettiin ja sovelluskehittäjät tutustuivat ensin oppaan sisältöön ja teoriaosuuteen omatoimisesti. Tämä varmisti, että heillä oli perusteellinen ymmärrys havainnointimenetelmistä ja -ohjeista sekä mahdollisuus kysyä tarkentavia kysymyksiä ennen kenttätöitä. Pilotoinnin aikana sovelluskehittäjät dokumentoivat itsenäisesti havaintonsa oppaan ohjeistuksen mukaisesti.

Neljän kehityskirin jälkeen kerätty palaute sovelluskehittäjiltä oli keskeinen osa pilotointia. Palaute kerättiin vapaamuotoisina haastatteluina, joissa kehittäjät arvioivat oppaan käytettävyyttä, ymmärrettävyyttä ja hyödyllisyyttä suullisesti. Kukin haastattelu toteutettiin keskustelunomaisesti, jossa kehittäjät saivat tuoda esiin kokemuksiaan ja huomioitaan reaaliajassa käytön aikana kohdanneista haasteista ja eduista. He keskittyivät erityisesti siihen, kuinka oppaan ohjeet tukivat heidän havainnointityötään ja mitä jatkokehitysehdotuksia oli pilotoinnin aikana nousnut. Tämä interaktiivinen palaute auttoi hahmottamaan, mitkä osat oppaasta toimivat hyvin ja mitkä kaipasivat parannuksia tai tarkennuksia.

5.4 Palaute ja jatkokehitys

Pilotoinnin aikana nousseiden palautteiden analysointi ja huomioonottaminen ovat olennainen osa havainnointioppaan kehitystä. Ne tarjoavat arvokasta tietoa oppaan vahvuuksista ja jatkokehityskohteista. Pilotoinnin tuloksena saadut palautteet osoittivat, että vaikka opas toimi pääosin tarkoitetulla tavalla, siinä oli tilaa pienille säädöille, jotka parantaisivat sen soveltuvuutta käytännön työhön.

Osa sovelluskehittäjistä kokivat haasteita erityisesti käsinkirjoittamisen kanssa, mikä hidasti havainnointi ja vaikutti heidän kykyynsä keskittyä ja kommunikoida tehokkaasti. Tämän seurauksena kehitystiimi otti käyttöön oman mieltymyksen mukaan joko digitaalisen tai paperisen dokumentointipohjan. Tämä mahdollistaa nopeamman ja helpomman tavan kirjata havaintoja ilman keskeytyksiä.

Havainnointioppaan taulukoiden pieni koko ja epäselvät ohjeistukset myös nousivat esille palautteissa. Kehittäjät eivät aina ymmärtäneet, mitä tietoja taulukoiden eri osiin tuli kirjata. Vastauksena tähän, taulukot suunniteltiin uudelleen suuremmiksi ja selkeämmin jaotelluiksi, ja niihin lisättiin tarkemmat ohjeistukset jokaiselle osiolle. Tämän muutoksen myötä kehittäjät pystyivät kirjaamaan havaintojaan yksityiskohtaisemmin ja jäsenytyneemmin, mikä paransi havainnointidatan analysointia ja hyödyntämistä. Tämä helpottaa myös havainnointioppaan laajempaa käyttöä sekä käyttöönottoa.

Sovelluskehittäjät suosittelivat myös keskittymistä yhteen ammattihenkilöön kerrallaan havainnoinnin aikana, mikä mahdollisti syvällisemmän ymmärryksen yksittäisen käyttäjän toiminnasta ja tarpeista. Tämä ehdotus johti siihen, että oppaaseen lisättiin suositus havainnoinnin kohdentamisesta, mikä auttaa havainnoijia fokuusoimaan huomionsa tehokkaammin ja vähentämään informaatioähkyä.

Lisäksi kehittäjät nostivat esille tarpeen tiedottaa kohdevierailukohteisiin etukäteen havainnointien tarkoituksesta, jotta havainnointitilanteet olisivat sujuvampia ja osallistujat tietoisia prosessista. Tähän vastattiin luomalla selkeät kehityskiritiimin sisäiset kommunikaatioprosessit, jotka selittävät havainnointien tavoitteet ja menetelmät etukäteen vierailukohteille.

Vaikka haasteita ilmeni, kehittäjät kokivat oppaan yleisesti hyödylliseksi. He löysivät pilottikehityskirien aikana hyvän tavan kirjata havaintoja ja pitivät oppaan jaottelua hyvin tehtynä. Teoriaosuus oli selkeä ja auttoi ymmärtämään kontekstia paremmin. Opas helpotti muistiinpanojen tekemistä, mikä paransi havainnoinnin laatua. Opasta pidettiin hyödyllisenä työkaluna jatkossakin ja sen käyttöä ehdotettiin laajennettavaksi.

Näiden parannusten myötä sekä havainnointioppaan käytettävyys että tehokkuus parantuivat merkittävästi. Pilotoinnista saatu palaute oli korvaamatonta oppaan kehittämisen kannalta ja se mahdollisti

käyttäjystävällisemmän, toimivamman oppaan luomisen tulevia kohdevierailuja varten.

6 Pohdinta ja johtopäätökset

6.1 Opinnäytetyön tulokset

Opinnäytetyön keskeinen tavoite oli luoda strukturoitu havainnointiopas, joka tukee käyttäjäkeskeistä suunnittelua Apotti -potilastietojärjestelmän kehityksessä. Opinnäytetyö onnistui vastaamaan tunnistettuun tarpeeseen kehittää opas, jolla kehityskirien aikana kerätyt havainnot voidaan systemaattisesti dokumentoida ja integroida ohjelmistokehitysprosessiin SAFe-viitekehityksen mukaisesti. Pilotoinnin tulokset osoittivat, että havainnointioppaan avulla havaintojen kerääminen oli selkeämpää, tarkempaa ja yhtenäisempää.

Pilotoinnissa ilmeni, että sovelluskehittäjät kokivat oppaan ohjeet hyödyllisiksi, erityisesti havainnointimenetelmien selkeä erittely auttoi heitä toimimaan tehokkaammin kenttäolosuhteissa. Samalla havaittiin kuitenkin pieniä kehityskohteita, kuten dokumentointilomakkeiden käytännöllisyys ja ohjeistuksen tarkentaminen, jotka korjattiin saadun palautteen pohjalta. Kehittäjien mukaan opas auttoi parantamaan käyttäjäymmärrystä ja tukemaan käytettävyyden parantamista Apotti-järjestelmän kehityksessä.

Tulokset osoittavat, että havainnointi on tehokas menetelmä monimutkaisten järjestelmien käytettävyyden parantamisessa. Havainnointioppaan avulla saatu strukturoitu tieto auttaa tunnistamaan loppukäyttäjien todellisia tarpeita ja haasteita, minkä avulla kehitysprosessia voidaan ohjata tarkemmin ja tuottaa käyttäjälähtöisempiä ratkaisuja. Opinnäytetyön tulokset vahvistavat myös sen, että käyttäjäkeskeinen suunnittelu ja SAFe-viitekehitys voivat toimia hyvin yhdessä, kun käytettävyyttä ja käyttäjäkokemusta parannetaan iteratiivisesti.

6.2 Suositukset jatkokehitykselle

Havainnointioppaan pilotointi toi esiin kehitysalueita, joiden parantaminen voi tehostaa sen käyttöä entisestään. Ensimmäisenä kehityssuuntana on oppaan täysi digitalisointi. Paperipohjaisen dokumentoinnin haasteet, kuten kirjoitusnopeuden hitaus ja tietojen siirtäminen analysointiin, voitaisiin ratkaista kehittämällä digitaalinen ja helppokäyttöinen versio oppaasta. Tällainen työkalu voisi sisältää valmiita lomakkeita, jotka helpottavat havaintojen kirjaamista ja automaattista analysointia.

Toinen suositus liittyy oppaan laajempaan käyttöönottoon. Oppaan sovellettavuutta voidaan testata muissa projekteissa ja tiimeissä, jotka hyödyntävät käyttäjäkeskeistä suunnittelua ja ketteriä menetelmiä. Tämä mahdollistaisi oppaan toimivuuden arvioinnin erilaisissa konteksteissa ja käyttäjäryhmissä.

Lisäksi havainnointioppaan koulutusprosessi on tärkeä kehityskohde. Havainnoinnin toteuttaminen vaatii osaamista, ja siksi oppaaseen voidaan liittää erillinen koulutusmateriaali tai ohjevideo, joka tukee käyttäjiä ymmärtämään havainnoinnin ja oppaan käytön perusteet. Tämän avulla voidaan varmistaa, että havainnoijat käyttävät opasta johdonmukaisesti ja sen tuottama data on luotettavaa ja kattavaa.

Viimeisenä suosituksena on kehittää havainnointidatan visualisointia. Kerätyn tiedon esittäminen selkeästi ja visuaalisesti voi helpottaa päätöksentekoa ja kehitystoimien priorisointia. Esimerkiksi dashboard-tyyppinen ratkaisu voisi tarjota kehitystiimille nopeasti hahmotettavan kokonaiskuvan tärkeimmistä havainnoista.

6.3 Opinnäytetyön merkitys ja sovellusmahdollisuudet

Tämä opinnäytetyö on merkittävä lisäys Oy Apotti Ab:n kehityskirimalliin, jossa tavoitteena on käyttäjälähtöinen ja käyttäjätyytyväisyyttä parantava ohjelmistokehitys design sprint -mallilla. Strukturoidun havainnointioppaan kehittäminen vastaa suoraan organisaation tunnistamaan tarpeeseen parantaa havainnointien systemaattisuutta ja hyödyntämistä. Oppaan avulla käyttäjien tarpeet voidaan tunnistaa aikaisessa vaiheessa, mikä tukee ohjelmistokehityksen laatua ja vähentää korjaustarpeita myöhemmissä vaiheissa.

Havainnointioppaan sovellusmahdollisuudet ulottuvat laajemmin myös muille toimialoille, joissa käyttäjälähtöinen suunnittelu on keskeistä. Esimerkiksi terveydenhuollon ohjelmistojen lisäksi opas voidaan mukauttaa muiden monimutkaisten järjestelmien kehitykseen, kuten logistiikkaan, teollisuuden automaatioon tai julkishallinnon järjestelmiin. SAFe-viitekehystä hyödyntävät organisaatiot voivat omaksua oppaan osaksi kehitysprosessejaan ja parantaa käyttäjäymmärryksen integroimista tuotekehitykseen.

Opinnäytetyö osoittaa, kuinka havainnoinnin systematisointi voi tuoda konkreettisia tuloksia ohjelmistokehitykseen. Se korostaa käyttäjäkeskeisen suunnittelun merkitystä ja tarjoaa työkalun, joka auttaa organisaatioita saavuttamaan parempaa käytettävyyttä ja käyttäjäkokemusta. Tulevaisuudessa havainnointioppaan kehittäminen ja sen integrointi osaksi muita ketteriä menetelmiä voi avata uusia mahdollisuuksia tehokkaaseen ja laadukkaaseen ohjelmistokehitykseen.

Lähteet

- 1 Scaled Agile Framework. 2024. Verkkoaineisto. <<https://scaledagileframework.com/>>. Viitattu 20.2.2024.
- 2 Google Ventures The Design Sprint. 2024. Verkkoaineisto. <<https://www.gv.com/sprint/>>. Viitattu 22.2.2024.
- 3 Knapp Jake, Zeratsky John, Kowitz Braden. 2016. Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days. 1. painos. Simon & Schuster 8.3.2016, s. 3.
- 4 Rikke Friis Dam. 2024. The 5 Stages in the Design Thinking Process. Verkkoaineisto. <<https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process>>. Viitattu 10.5.2024.
- 5 Banfield Richard, Lombardo C. Todd, Wax Trace. 2015. Design Sprint. O'Reilly Media, Inc. syyskuu 2015, s. 18-19.
- 6 Ferreira Vinícius Gomes, Canedo Edna Dias. 2019. Using design sprint as a facilitator in active learning for students in the requirements engineering course: an experience report. SAC '19: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, s. 1852–1859. <<https://doi.org/10.1145/3297280.3297463>>.
- 7 Larusdottir Marta Kristin, Roto Virpi, Lanzilotti Rosa, Visescu Ioana Duta. 2022 Tutorial on UCD Sprint: Inclusive Process for Concept Design. NordiCHI '22 Adjunct: Adjunct Proceedings of the 2022 Nordic Human-Computer Interaction Conference 19, s. 1-3. <<https://doi.org/10.1145/3547522.3558901>>.
- 8 Apotti yrityksenä. 2024. Verkkoaineisto. <<https://www.apotti.fi/apotti/apotti-yrityksena/>>. Viitattu 1.3.2024.
- 9 Naistentautien poliklinikan kehityskiri. 2024. <<https://www.apotti.fi/naistentautien-poliklinikan-kehityskiri/>>. Viitattu 1.3.2024.
- 10 Lisää käyttäjätyytyväisyyttä ja yhteistyötä – Peijaksen sairaalan ortopedian vuodeosaston kehityskirissä pureuduttiin työn sujuvuuteen. 2024. Verkkoaineisto. <<https://www.apotti.fi/kehityskiri-lisaa-kayttajatyytyvaisyytta-ja-yhteistyota-peijaksessa/>>. Viitattu 1.3.2024.
- 11 Lyon Aaron R., Koerner Kelly. 2016. User-Centered Design for Psychosocial Intervention Development and Implementation. Clinical Psychology:

Science and Practice, 23(2), s. 180–200.
<<https://doi.org/10.1111/cpsp.12154>>.

- 12 Gould John D., Lewis Clayton. 1985. Designing for usability: key principles and what designers think. *Communications of the ACM*, 28 (3), s. 300-311. <<https://dl.acm.org/doi/10.1145/3166.3170>>.
- 13 Nemeth Adam, Bekmukhambetova Anara. 2022. Achieving Usability: Looking for Connections between User-Centred Design Practices and Resultant Usability Metrics in Agile Software Development. *Periodica Polytechnica Social and Management Sciences*, 31(2), s. 135–143. <<https://doi.org/10.3311/PPso.20512>>.
- 14 ISO 9241-11. 2018. Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts. International Organization for Standardization.
- 15 Bevan Nigel, Carter Jim, Earthy Jonathan, Geis Thomas, Harker Susan. 2016. New ISO Standards for Usability, Usability Reports and Usability Measures. *Human-Computer Interaction: Theory, Design, Development and Practice*, s. 268–278. <https://doi.org/10.1007/978-3-319-39510-4_25>.
- 16 Georgsson Mattias, Staggers Nancy. 2015. Quantifying usability: an evaluation of a diabetes mHealth system on effectiveness, efficiency, and satisfaction metrics with associated user characteristics. *Journal of the American Medical Informatics Association* 23(1), s. 5–11. <<https://doi.org/10.1093/jamia/ocv099>>.
- 17 Weichbroth Pawel. 2020. Usability of Mobile Applications: A Systematic Literature Study. *IEEE Journals & Magazine* 8. <<https://ieeexplore.ieee.org/document/9042272>>.
- 18 Fox Jean E. 2015. The Science of Usability Testing. Federal Committee on Statistical Methodology (FCSM) Research. <https://www.fcsm.gov/assets/files/docs/C2_Fox_2015FCSM.pdf>.
- 19 Weichbroth Pawel. 2024. Usability Testing of Mobile Applications: A Methodological Framework. *Applied Sciences*, 14(5), s. 1792. <<https://doi.org/10.3390/app14051792>>.
- 20 Pins Dominik, Paul Dennis. 2022. Towards a Framework for Supporting User Satisfaction of Conversational Agents according to the Usability Norm DIN EN ISO 9241-11. *Wirtschaftsinformatik* 2022. <<https://aisel.aisnet.org/wi2022/workshops/workshops/12>>.

- 21 Relawati Ambar, Zamroni Guntur Maulana, Primanda Yanuar. 2022. Unmoderated Remote Usability Testing: An Approach during Covid-19 Pandemic. *International Journal of Advanced Computer Science and Applications* 13(1). <<https://doi.org/10.14569/IJACSA.2022.0130135>>.
- 22 Weston Lauren E., Krein Sarah, Harrod Molly. 2022. Using observation to better understand the healthcare context. *Qualitative Research in Medicine & Healthcare* 5(3), s. 9821. <<https://doi.org/10.4081/qrmh.2021.9821>>.
- 23 Thompson Susan. 2003. Remote observation strategies for usability testing. *Information Technology and Libraries* 22(1), s. 22-31. <<https://www.proquest.com/scholarly-journals/remote-observation-strategies-usability-testing/docview/215828867/se-2>>.
- 24 Carthey Jane. 2003. The role of structured observational research in health care. *BMJ Quality & Safety* 12(2), s.13–16. <https://doi.org/10.1136/qhc.12.suppl_2.ii13>.
- 25 Amy J. Ko. 2017. A three-year participant observation of software startup software evolution. *IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track*, s. 3–12. <<https://doi.org/10.1109/ICSE-SEIP.2017.29>>.
- 26 Blocker Addy, Datay Mohammed Ishaq, Mwangama Joyce, Malila Bessie. 2024. Development of a telemedicine virtual clinic system for remote, rural, and underserved areas using user-centered design methods. *Digital health* 10. <<https://doi.org/10.1177/20552076241256752>>.
- 27 Aripiyanto Saepul, Muzayyana Agustin Fenty Eka, Saputrie Faraditya Praba, Durrachman Yusuf, Khairani Dewi, Sukmana Husni Teja. 2024. Evaluation of User Experience in the Lecture Activity Menu of AIS Mobile for Student Using Heuristic Evaluation, System Usability Scale and User Experience Questionnaire. *12th International Conference on Cyber and IT Service Management (CITSM)*, s. 1–7. <<https://doi.org/10.1109/CITSM64103.2024.10775924>>.
- 28 Amani Sara, Obeidat Noor, Ferris Thomas, Shryock Kristi. 2024. Enhancing the Onboarding Experience with Wearable Technology for Research Applications. *AHFE International Conference*. <<https://doi.org/10.54941/ahfe1005662>>.
- 29 Märtin Christian, Herdin Christian. 2024. Enabling real-time adaptations for individualized customer experience in user-centered e-business applications. *Procedia Computer Science* 239, s. 1425–1432. <<https://doi.org/10.1016/j.procs.2024.06.315>>.

- 30 Apotin strategia uudistunut. 2024. Verkkoaineisto.
<<https://www.apotti.fi/apotin-strategia-uudistunut/>>. Viitattu 12.11.2024.
- 31 Hyysalo Sampsa. 2009. Käyttäjä tuotekehityksessä. Tieto, tutkimus, menetelmät. Taideteollisen korkeakoulun julkaisu. B, 97/2009. Aalto-yliopiston taiteiden ja suunnittelun korkeakoulu, Muotoilun laitos.

Havainnointiopas

Kehityskirin kohdevierailulle

10.10.2024

Havainnoinnin tavoite

Kehityskirin kohdevierailun aikana tehtävän havainnoinnin tavoitteena on saada katsaus loppukäyttäjien työympäristöön, potilaskontekstiin ja jokapäiväiseen työhön. Tavoitteena on myös saada yleisymmärrys loppukäyttäjien tämän hetkisestä tavasta käyttää Apottia ja löytää mahdollisia kehityskohteita. Havainnointi voi kohdistua tiettyyn työnkulkuun tai laajemmin tietyn käyttäjäryhmän tehtäviin.

Havainnointiohjeet

- 1. Valmistautuminen:** Tutustu etukäteen käyttäjäryhmiin, joita aiot havainnoida ja heidän tyypillisiin työtehtäviinsä ja työnkulkuihinsa. Tutustu myös vierailtavan kohteen toimintaan ja tyypillisiin potilaisiin. Käyttäjäymmärryksessä auttavat mahdolliset käyttäjäkuvaukset ja kehityskirissä mukana olevan apottiasiantuntijan haastattelu. Pyri oppimaan ”oikeat” työnkulut, jotta huomaat mahdolliset havainnot kohdevierailussa. Informoi kohdevierailuyksikköä tehtävästä havainnoinnista.
- 2. Valitse havainnointimenetelmä:**
 - a) Havainnointihaastattelu**

Havainnointihaastattelussa havainnoija seuraa aktiivisesti työnkulkua ja kysyy aina, kun hänelle on epäselvää mitä, miten tai miksi havainnoitava toimii kuten toimii. Havainnoija voi myös pyytää havainnoitavaa selostamaan tekemistään. Havainnointihaastattelu sopii tilanteeseen, jossa työnkulut tai työtehtävät eivät ole havainnoijalle tuttuja.
 - b) Passiivinen havainnointi**

Passiivisessa havainnoinnissa havainnoija seuraa sivusta työnkulkua ja keskittyy muistiinpanojen tekemiseen. Mahdolliset tarkentavat avoimet kysymykset voidaan kysyä havainnoinnin jälkeen. Passiivinen havainnointi soveltuu tehokkaaseen kokonaiskuvan saamiseen.
- 3. Havainnoi aktiivisesti:** Keskity yhteen loppukäyttäjään kerrallaan. Keskitä huomiosi loppukäyttäjän vuorovaikutukseen järjestelmän kanssa. Kiinnitä huomiota sekä suoriin että epäsuoriin vihjeisiin, kuten käyttäjien eleisiin, ilmeisiin ja kommentteihin. Kirjaa ne sellaisenaan muistiinpanoihin.
- 4. Muistiinpanojen tekeminen:** Kirjoita muistiinpanot käsin paperille. Kirjaa ylös havaintosi mahdollisimman tarkasti ja objektiivisesti. Älä lisää havaintoihin omia päätelmiä. Voit käyttää lyhenteitä ja symboleja nopeuttaaksesi muistiinpanojen tekemistä. Jokaiseen muistiinpanopohjan kysymykseen ei tarvitse olla vastausta.

Havaintojen käsittely ja analysointi

1. **Havaintojen järjestely:** Kerää kaikki kohdevierailulla tehdyt havainnot yhteen paikkaan. Käytä digitaalista alustaa, kuten esimerkiksi Miroa tai Exceliä, joka mahdollistaa helpon havaintojen jakamisen ja käsittelyn.
2. **Teemoittelu:** Ryhmittele havainnot järjestelmän osa-alueitten tai toimintojen mukaan. Etsi mahdollisia yhdistäviä tekijöitä, kuten yhteisiä ongelmia, käyttäytymismalleja ja käyttäjien tarpeita.
3. **Priorisointi:** Arvioi havaintojen kriittisyys ja vaikutus käyttäjäkokemukseen, laatuun ja potilasturvallisuuteen. Käytä esimerkiksi MoSCoW-menetelmää (Must have, Should have, Could have, Won't have this time).
4. **Muunnos SAFe-viitekehykseen:** Arvioi havainnoille oikea kehityspotki, jolla korjaus asiakkaalle voidaan toimittaa SAFe-viitekehyksen mukaisesti.
 - a) **Kehitysehdotus:** Havainnon korjaus vaatii järjestelmän uutta kehitystä.
 - b) **Häiriö:** Havainto johtuu nykyisen työnkulun häiriöstä.
 - c) **Palvelupyyntö:** Havainto voidaan korjata pienkehityksellä ja korjaustoimenpiteelle löytyy palvelupyyntökatalogista sopiva lomake.

