



Sakeus Oy:n muotoiluprosessit käytännössä

Simo Virtanen | Opinnäytetyö 2024
Metropolia Ammattikorkeakoulu
Muotoilun tutkinto-ohjelma YAMK

Tiivistelmä

Otsikko: Sakeus Oy:n muotoiluprosessit käytännössä

2024

Tämä opinnäytetyö tarkastelee kolmihenkisen design- ja ohjelmistokehityskonsulttiyritys Sakeus Oy:n muotoiluprosessia. Työssä käydään läpi kaksi HeatSpring-asiakkaalle toteutettua toimeksiantoa: Team Dashboard -ominaisuuden uudelleensuunnittelu ja kouluttajien työkalujen kehittämistä. Toimeksiantojen kautta käydään läpi sitä, miten muotoiluajattelun, palvelumuotoilun ja ketterän ohjelmistokehityksen periaatteet toteutuvat Sakeuden työskentelyssä.

Työssä kuvataan Sakeuden käyttämiä menetelmiä, kuten työpajatyöskentelyä asiakastarpeiden kartoituksessa, prototyyppien kehittämistä ja työvaiheiden porrastamista. Lisäksi tarkastellaan miten tiimin monipuolinen osaaminen vaikuttaa muotoiluprosessiin. Työ osoittaa, että Sakeus toteuttaa muotoilua käytäntölähtöisesti, pyrkien validoimaan ideoita nopeasti prototyypeillä ja MVP-toteutuksilla.

Opinnäytetyössä tunnistetaan myös haasteita, kuten systemaattisen loppukäyttäjäpalautteen keräämisen puute ja pienien budjettien vaikutus kokeelliseen kehittämiseen. Työ tarjoaa lopputuloksenaan Sakeudelle näkemyksiä omien prosessiensa toimivuudesta ja herättää keskustelua yrityksen asemoinnista alan liiketoimintakentässä. Työssä käydään läpi se, miten Sakeuden muotoiluprosessi on kehittynyt kypsälle tasolle, mutta kuinka sen kehittyminen seuraavalle tasolle edellyttäisi tiiviimpää asiakasyhteistyötä ja kattavampaa käyttäjätutkimusta asiakkaiden liiketoimintakentässä.

Avainsanat: muotoiluajattelu, palvelumuotoilu, muotoiluprosessi, käyttäjälähtöinen suunnittelu, ketterä ohjelmistokehitys

Tekijä	Simo Virtanen
Ammattikorkeakoulu	Metropolia Ammattikorkeakoulu
Tutkinto	Muotoilija (ylempi AMK)
Tutkinto-ohjelma	Muotoilun tutkinto-ohjelma
Ohjaaja	Sauli Laitinen
Sivumäärä	50

Abstract

Title: The Practical Design Processes of Sakeus Oy

2024

This thesis examines the design process of Sakeus Oy, a three-person design and software development consultancy. The work reviews two project assignments completed for the client HeatSpring: the redesign of the Team Dashboard feature and the development of tools for course instructors. Through these assignments, the thesis explores how the principles of design thinking, service design, and agile software development are implemented in Sakeus's work.

The thesis describes the methods used by Sakeus, such as workshop sessions for mapping customer needs, prototype development, and phasing of work stages. Additionally, it examines how the team's diverse expertise influences the design process. The work demonstrates that Sakeus implements design in a practice-oriented manner, aiming to validate ideas quickly with prototypes and MVP implementations.

The thesis also identifies challenges, such as the lack of systematic end-user feedback collection and the impact of small budgets on experimental development. As a result, the work provides Sakeus with insights into the effectiveness of their own processes and helps discussion about the company's positioning in the industry's business landscape. The thesis reviews how Sakeus's design process has matured to a sophisticated level, but how its evolution to the next level would require closer customer collaboration and more comprehensive user research in the clients' business fields.

Keywords: design thinking, service design, design process, user-centered design, agile software development

Author	Simo Virtanen
UAS	Metropolia University of Applied Sciences
Degree	Master of Culture and Arts
Degree Programme	Master's Degree Program in Design
Instructor	Sauli Laitinen
Number of Pages	50

Sisällysluettelo

1 Johdanto	1
1.1 Työn tilaaja	2
1.2 Työn tavoitteet ja rajaukset	2
2 Teoria ja käsitteet	3
2.1 Muotoiluajattelu ja palvelumuotoilu	3
2.2 Käyttäjälähtöinen suunnittelu ja käyttäjäkokemus	8
2.3 Ketterä ohjelmistokehitys	9
2.4 Lean Startup	10
2.5 Lean UX	12
2.6 Shape Up -malli	13
2.7 Keskeiset käytettävät työkalut ja teknologiat	15
2.7.1 Figma	15
2.7.2 Ruby on Rails	16
2.7.3 Tailwind CSS	16
3 Toimeksiannot	17
3.1 HeatSpring asiakkaana	17
3.2 HeatSpring – Team Dashboard	18
3.2.1 Suunnittelutyö	21
3.2.2 Rakentaminen	24
3.2.3 Tulokset ja retrospektiivi	26
3.3 HeatSpring – Kouluttajien työkalut	28
3.3.1 Suunnittelutyöpaja	28
3.3.2 Suunnitteluvaihe	31
3.3.3 Rakentaminen	34
3.3.4 Tulokset ja tavoitteissa onnistuminen	35
4 Sakeuden muotoiluprosessit ja toimeksiannoissa tehdyt havainnot	36
4.1 Asiakastarpeiden kartoittaminen	37

4.2 Prototyyppaus	37
4.3 Työvaiheiden porrastaminen	39
4.4 Toiminnallisuuksien testaus ja julkaisu	40
4.5 Työn jakautuminen ja toteutusosaaminen muotoiluprosessissa	41
4.6 Havaitut haasteet muotoiluprosessin täytäntöönpanolle	44
5 Yhteenveto	45
Lähteet	48

1 Johdanto

Suomessa on lukuisia erikokoisia ohjelmistokonsultointia tarjoavia yrityksiä, joilla on erilaiset henkilöstöresurssit, toimintamallit ja tarjooma. Jotkin yritykset painottavat enemmän muotoilua, toiset tarjoavat lisää ohjelmistokehittäjiä yritysten sisäisiin kehitystiimeihin, ja joidenkin liiketoiminta keskittyy tarjoamaan mukautuksia valmiisiin ohjelmistoihin. Liiketoimintaympäristö on haastava sekä siinä toimiville pienemmille yrityksille, jotka pyrkivät erottumaan varteenotettavina toimittajina suuremmilla resursseilla varustettujen yritysten rinnalla, että asiakkaille, jotka eivät välttämättä ole tietoisia siitä, minkä yrityksen toimintamallit soveltuvat juuri heidän toimintansa, tuotteidensa ja palveluidensa kehittämiseen.

Tässä opinnäytetyössä tarkastellaan konsulttiyritys Sakeus Oy:n (myöhemmin tekstissä Sakeus) työskentelyä. Sakeus toimii kolmihenkisen tiimin voimin moniprojektiympäristössä, jossa tiimin jäsenet työskentelevät samanaikaisesti useiden asiakkaiden ja toimeksiantojen parissa. Yritys pyrkii hyödyntämään projekteissa koko tiiminsä laajaa osaamista palvelumuotoilussa, käyttöliittymäsuunnittelussa ja -kehityksessä sekä ohjelmistokehityksessä. Tämän kokonaisvaltaisen lähestymistavan tuottaman arvon selittäminen asiakkaalle on kuitenkin osoittautunut toisinaan haastavaksi. Haasteet korostuvat erityisesti silloin, kun asiakkaalla on vähäinen kokemus ohjelmistokehityksestä tai kun asiakas näkee ohjelmiston pelkkänä hyödykkeenä, jonka kehitystiimi voi tuottaa "liukuhihnalta" ymmärtämättä asiakkaan liiketoimintaa tai tarpeita. Arvo on kuitenkin mahdollista tuoda näkyväksi tekemisen kautta.

Opinnäytetyössä kuvataan sitä, miten muotoilun menetelmät toteutuvat Sakeuden käytännön työskentelyssä. Koska lähestymistapa on käytännönläheinen, opinnäytetyössä kuvataan toimeksiantojen prosessia tiimin toiminnan kautta. Töiden etenemistä havainnollistetaan useilla kuvakaappauksilla muotoiluprosessin eri vaiheista.

Johdannossa esitellään työn tilaaja Sakeus, opinnäytetyön merkitys

Sakeuden liiketoiminnalle sekä tutkimuskysymys. Tämän jälkeen käsitellään työn teoreettinen viitekehys, joka luo perustan seuraavassa osiossa yksityiskohtaisesti kuvattavien kahden asiakastoimeksiannon analysoinnille. Lopuksi tarkastellaan Sakeuden hyödyntämiä muotoilumenetelmiä sekä yrityksen toteuttaman muotoiluprosessin kehityskohteita.

1.1 Työn tilaaja

Sakeus on vuonna 2022 perustettu design-konsulttiyritys, joka tarjoaa ohjelmistokehityspalveluja. Yritys suunnittelee ja toteuttaa räätälöityjä ohjelmistoja yritysten yksilöllisiin tarpeisiin. Sakeuden asiantuntijat ovat moniosaajia, jotka eivät ole erikoistuneet vain yhteen osa-alueeseen, kuten käyttöliittymäsuunnitteluun tai koodaukseen. Sen sijaan he mukauttavat rooliaan ohjelmistoprojektin elinkaaren eri vaiheissa. Asiantuntijoilla on kuitenkin tietyt osaamispainotukset, joiden perusteella heidät voidaan jakaa suunnittelijoihin (designer) ja kehittäjiin (software developer).

Opinnäytetyössä kuvatut projektit toteuttanut tiimi koostuu seuraavista henkilöistä:

- Simo Virtanen, suunnittelija ja opinnäytetyön kirjoittaja
- Henri Nikka, suunnittelija
- Virgil Mocanu, kehittäjä

Opinnäytetyön sisällöstä ja siinä esitetyistä näkemyksistä vastaa Simo Virtanen. Esiteltyjen projektien lopputuloksiin on päästy tiimin tasavertaisella yhteistyöllä ja työpanoksella. Projektien aikana päätöksenteossa on tehty läheistä yhteistyötä Sakeuden asiakkaiden kanssa, joille projektit on toteutettu toimeksiantoina.

1.2 Työn tavoitteet ja rajaukset

Opinnäytetyön päätavoitteena on kuvata ja analysoida Sakeus Oy:n muotoiluprosessia käytännön projekteissa. Työssä pyritään

havainnollistamaan, miten teoreettiset muotoilun ja ohjelmistokehityksen mallit toteutuvat ja mukautuvat todellisissa asiakasprojekteissa. Lisäksi tavoitteena on tunnistaa ja esitellä ne käytännöt ja menetelmät, jotka ovat osoittautuneet tehokkaiksi Sakeuden toimintaympäristössä.

Opinnäytetyön tavoitteena on myös kehittää Sakeuden prosesseja käyttäjä- ja asiakaslähtöisten palveluiden tuottajana sekä sanallistaa yrityksen työskentelytapoja ja niiden tuottamaa asiakasarvoa. Tätä tietoa yritys voi hyödyntää viestinnässään mainonta- ja markkinointi-kontekstissa löytääkseen asiakkaita, joille Sakeuden käytännönläheinen ja ideoiden koestamiseen tähtäävä työskentelymalli soveltuu.

Vaikka tämä työ ja siinä kuvatut toimeksiannot sijoittuvat ohjelmistokehityksen kontekstiin, opinnäytetyössä ei pyritä kuvaamaan toteutuksissa käytettyjen teknisten ratkaisujen tai työkalujen vaikutuksia. Työssä ei myöskään tehdä vertailua muihin alan toimijoihin ja näiden prosesseihin, vaan työ keskittyy yksinomaan Sakeuden omaan toimintaan. Nämä rajaukset huomioiden työn tutkimuskysymykseksi muodostuu: *Miten Sakeus Oy:n muotoiluprosessi toteutuu käytännön projekteissa?*

2 Teoria ja käsitteet

Tässä osiossa käsitellään opinnäytetyön kannalta keskeisiä teoreettisia käsitteitä ja viitekehyksiä. Nämä teoreettiset lähtökohdat muodostavat perustan, jonka pohjalta työssä esiteltyjä toimeksiantoja ja prosesseja tarkastellaan. Erityisesti perehdytään muotoiluajattelun, palvelumuotoilun, käyttäjälähtöisen suunnittelun ja ketterän ohjelmistokehityksen periaatteisiin.

2.1 Muotoiluajattelu ja palvelumuotoilu

Muotoiluajattelu on työssä kuvattujen toimeksiantojen kannalta tärkeä käsite. Sille ei ole määritetty yhtä selkeää määritelmää, vaan muotoilijoilla on tapana määrittää termiä hieman eri tavoin. Koska

muotoiluajattelun määritelmä vaihtelee, esitetään seuraavaksi muutamia esimerkkejä termin määritelmälle. Tämän jälkeen kuvaan, miten muotoiluajattelu määritetään tämän työn viitekehyksessä.

“Muotoiluajattelu voidaan määritellä oppialaksi, joka hyödyntää muotoilijan herkkyyttä ja menetelmiä yhdistääkseen ihmisten tarpeet teknologisesti toteutettavissa oleviin ratkaisuihin ja siihen, miten toimiva liiketoimintastrategia voi muuntaa ne asiakasarvoksi ja liiketoimintamahdollisuuksiksi (Brown 2008).”

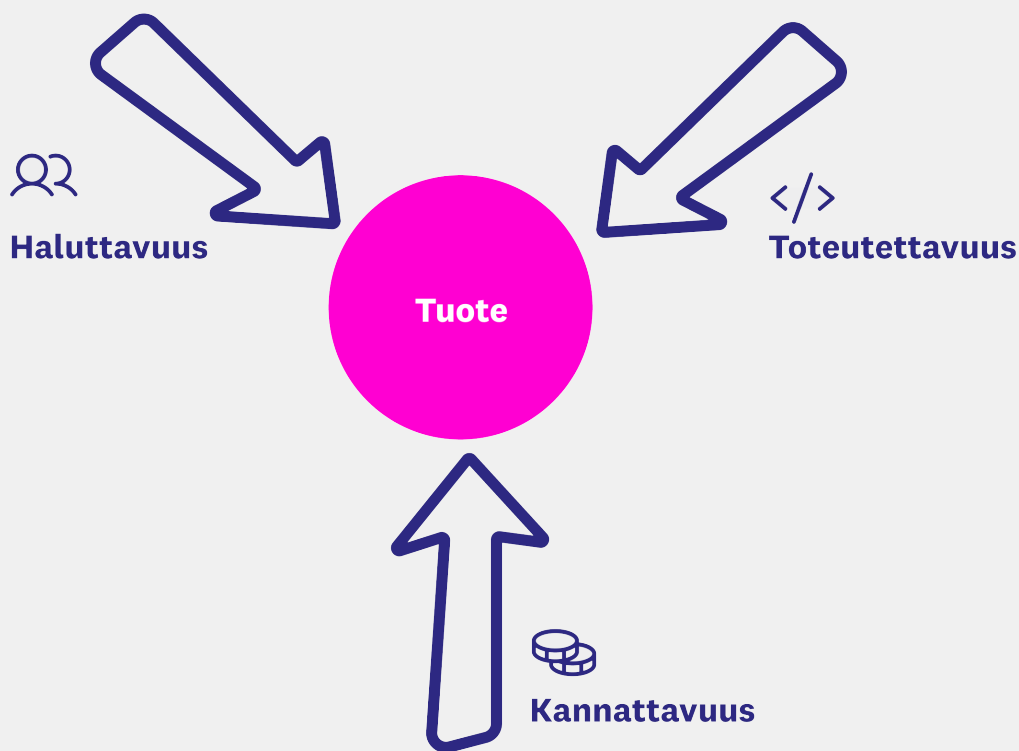
“Se tarkoittaa astumista askeleen taaksepäin välittömästä ongelmasta ja laajemman näkökulman ottamista. Se vaatii systeemiajattelua: ymmärrystä siitä, että jokainen ongelma on osa suurempaa kokonaisuutta, ja että ratkaisu todennäköisesti edellyttää koko järjestelmän ymmärtämistä. Muotoiluajattelu vaatii syvää perehtymistä aiheeseen, usein havainnoinnin ja analyysin kautta. Testit ja toistuvat tarkistukset voivat olla osa prosessia. Joskus tätä tehdään ryhmissä: moniammatilliset tiimit tuovat erilaista asiantuntemusta ongelman ratkaisuun.” (Norman 2010.)

“Muotoiluajattelu on iteratiivinen prosessi, jossa pyritään ymmärtämään käyttäjiä, haastamaan oletuksia, määrittelemään ongelmia uudelleen ja luomaan innovatiivisia ratkaisuja, joita voidaan prototypoida ja testata. Yleisenä tavoitteena on tunnistaa vaihtoehtoisia strategioita ja ratkaisuja, jotka eivät ole välittömästi ilmeisiä alkuperäisellä ymmärryksen tasolla.” (Dam & Siang 2024.)

Esitettyjen esimerkkien pohjalta tämän työn määritelmä muotoiluajattelulle on seuraava: Muotoiluajattelu on yhdistelmä tapoja lähestyä, määritellä ja selvittää ongelmia, joiden tavoitteena on toteuttaa uusia ihmis- ja liiketoimintalähtöisiä ratkaisuja. Näillä ratkaisuilla pyritään holistisesti huomioimaan myös niiden merkitys ja vaikutukset

sidoksissa oleviin ihmisiin ja toimintoihin, eli ratkaisuja tarkastellaan osana laajempaa järjestelmää, sekä niiden toteutettavuus.

Brownin (2019) mukaan muotoiluajattelu koostuu kolmesta osatekijästä: haluttavuus, toteutettavuus ja kannattavuus. Haluttavuus viittaa tuotteen tai palvelun kykyyn vastata käyttäjiensä tarpeisiin ja tavoitteisiin. Toteutettavuus tarkoittaa ratkaisujen suunnittelua, jotka ovat toiminnallisesti mahdollisia toteuttaa tarvittavassa ajassa. Kannattavuus puolestaan liittyy suunnitelmien tavoitteeseen tuottaa ratkaisu, joka muodostuu osaksi kestäväää liiketoimintamallia. Kuva 1 havainnollistaa näitä osatekijöitä.



Kuva 1. Muotoiluajattelun osa-alueet (Mukaelma: Brown 2019, 26).

Kuten muotoiluajattelu, myös palvelumuotoilun määritelmä vaihtelee eri lähteiden välillä. Esimerkiksi palvelumuotoilua on kuvattu seuraavasti:

“Palvelumuotoilu on palveluiden suunnittelun käytäntö. Se käyttää kokonaisvaltaista ja erittäin yhteistyöhön perustuvaa lähestymistapaa tuottaakseen arvoa sekä palvelun käyttäjälle että palveluntarjoajalle koko palvelun

elinkaaren ajan. Käytännössä palvelumuotoilu auttaa koreografioimaan prosesseja, teknologioita ja vuorovaikutuksia, jotka ohjaavat palveluiden toimitusta, käyttäen ihmiskeskeistä näkökulmaa. Palvelumuotoilu on nykyään sovellettavissa useilla eri sektoreilla, auttaen toteuttamaan strategisia ja taktisia tavoitteita sekä yksityisellä että julkisella sektorilla.” (Service Design Network 2019.)

”Palvelumuotoilu auttaa organisaatioita näkemään palvelunsa asiakkaan näkökulmasta. Se on lähestymistapa palveluiden suunnitteluun, joka tasapainottaa asiakkaan tarpeet liiketoiminnan tarpeiden kanssa, pyrkien luomaan saumattomia ja laadukkaita palvelukokemuksia. Palvelumuotoilu pohjautuu muotoiluajatteluun ja tuo luovan, ihmiskeskeisen prosessin palveluiden parantamiseen ja uusien palveluiden suunnitteluun. Yhteistyöhön perustuvien menetelmien avulla, jotka osallistavat sekä asiakkaita että palvelun toimittamisesta vastaavia tiimejä, palvelumuotoilu auttaa organisaatioita saamaan todellisen, päästä päähän - ymmärryksen palveluistaan, mahdollistaen kokonaisvaltaiset ja merkitykselliset parannukset.” (Lawrence ym. 2018, 20.)

”Palvelumuotoilu toimii ihanteellisesti strategisella liiketoimintatasolla, yhdistäen liiketoimintaehdotukset siihen, miten ne käytännössä toteutetaan. Se myös edistää ajatusta ihmisten kanssa suunnittelusta, ei vain heille suunnittelusta. Tämä voi tarkoittaa esimerkiksi ”yhteistuotannon” käsitteen käyttöä tai menetelmiä, jotka sisällyttävät useita sidosryhmiä organisaation sisällä, kuten johdon ja etulinjan henkilöstön. Näemme palvelumuotoilun erilaisena kuin muotoiluajattelun siinä, että se keskittyy myös suunnittelun toteuttamiseen ja implementointiin. Se hyödyntää myös suunnittelijoiden kykyä visualisoida ja tehdä

abstrakteista ideoista konkreettisia.” (Løvlie, Polaine & Reason 2013, 10.)

Esimerkkien kautta on havaittavissa, että muotoiluajattelu ja palvelumuotoilu sisältävät paljon yhteisiä osa-alueita. Tällaisia ovat muun muassa käyttäjä- ja liiketoimintalähtöisyys ja kokonaisuuden sidokseisuuksien tarkastelu ja huomioiminen. Eroavaisuutena nostetaan se, että palvelumuotoilu keskittyy lisäksi käytännön toteutukseen ja implementaatioon. Lisäksi palvelumuotoilua ei nähdä muotoiluajattelun tapaan vain innovaation välineenä, vaan myös uudistamisen ja palveluiden parantamisen keinona. Ero muotoiluajattelun ja palvelumuotoilun käytännönläheisyydessä ei kuitenkaan ole yksiselitteinen, vaan tekeminen voidaan nähdä myös ajattelun jatkeena:

”Muotoiluajattelun keskeinen vetovoima, joka usein ymmärretään väärin, on se, että ajattelun määritelmää venytetään paljon kartesiolaista ajattelevan subjektin mallia pidemmälle. Muotoilija ei ole henkilö, joka teoretisoi maailmasta ja toimii spekulaaation pohjalta, sillä tämä ei edusta ihmisen todellista olemusta. Ajattelu sisältää luonnostaan tekemisen. Ajatusten ulkoistaminen pään ulkopuolelle ei ole vain hyödyllistä, vaan välttämätöntä. Vain tekemisen kautta pystymme todella ajattelemaan. Prototyyppien luominen ei esimerkiksi ole vain esineiden valmistamista, vaan pikemminkin ideoiden tutkimista fyysisessä tilassa.” (Wendt 2015, 49.)

Kun sisällytämme Wendtin näkemyksen tekemisen kautta ajattelusta osaksi muotoiluajattelua, voidaan sanoa, ettei muotoiluajattelun ja palvelumuotoilun välillä ole eroa. Tässä työssä palvelumuotoilu kuitenkin määritetään siten, että muotoiluajattelua toteuttavia prosessit ja työskentelymenetelmät ovat palvelumuotoilua, kun taas muotoiluajattelu on tapa ajatella ja lähestyä ongelmia. Muotoiluajattelu toimii siis ikään kuin filosofisena perustana, jonka pohjalta palvelumuotoilun käytännön menetelmät ja työkalut on kehitetty. Tämä erottelu auttaa ymmärtämään

muotoiluajattelun laajemman sovellettavuuden, kun taas palvelumuotoilu keskittyy erityisesti palveluiden kehittämiseen ja parantamiseen.

2.2 Käyttäjälähtöinen suunnittelu ja käyttäjäkokemus

Käyttäjälähtöinen suunnittelu, josta käytetään myös termiä ihmislähtöinen suunnittelu, on lähestymistapa, jossa asetetaan ihmisten tarpeet, kyvyt ja käyttäytymisen etusijalle, ja suunnitellaan sitten näihin tarpeisiin, kykyihin ja käyttäytymistapoihin soveltuvia tuotteita (Norman 2013, 8). Käyttäjälähtöinen suunnittelun ydinajatus on auttaa luomaan ohjelmistoja, jotka vastaavat käyttäjien todellisiin tarpeisiin. Menetelmässä pyritään minimoimaan suunnittelijan subjektiivisten oletusten vaikutus lopputulokseen. Sen sijaan suunnittelupäätökset perustuvat tutkittuun tietoon, jota saadaan tarkkailemalla ja kuuntelemalla käyttäjiä. (Lowdermilk 2013, luku 2.) Käyttäjälähtöisen suunnittelun lopputuotteissa pyritään siihen, että niillä on hyvä käytettävyys. Krug (2013, 9) määrittelee käytettävyyden siten, että keskiverto tai jopa keskivertoa heikommin pärjäävä henkilö pystyy selvittämään, miten asiaa käytetään jonkin tehtävän suorittamiseen ilman, että siitä koituu enemmän vaivaa kuin hyötyä.

Käyttäjälähtöisen suunnittelun rinnalla puhutaan usein käyttäjäkokemuksesta. Nielsen & Norman (1998) määrittelevät käyttäjäkokemuksen kokonaisvaltaiseksi kokemukseksi, joka syntyy loppukäyttäjän ja yrityksen sekä sen tuotteiden ja palveluiden välisestä vuorovaikutuksesta. Laadukkaan käyttäjäkokemuksen edellytyksenä on useiden eri alojen – kuten insinööriyön, markkinoinnin, graafisen ja teollisen muotoilun sekä käyttöliittymäsuunnittelun – saumatonta yhteensovittamista. (Nielsen & Norman 1998.) Lowdermilkin (2013, luku 2) mukaan käyttäjäkokemuksella tarkoitetaan usein ensisijaisesti ohjelmistotuotteen kokonaisvaltaista kokemusta. Sillä ei pyritä kuvaamaan vain toiminnallisuuksia, vaan myös sitä, kuinka miellyttävää tuotteen käyttö on. (Lowdermilk 2013, luku 2.) Ohjelmistokehityksessä käyttäjäkokemuksusuunnittelulla pyritään varmistamaan mahdollisimman laadukkaan käyttäjäkokemuksen toteutuminen.

Tässä työssä käyttökokemussuunnittelua käsitellään yleisesti osana käyttäjälähtöistä suunnittelua. On kuitenkin huomioitavaa, että käyttäjäkokemussuunnittelu sisältää useita eri osaamisaloja, kuten informaatioarkkitehtuuri, interaktiosuunnittelu sekä sisältö- ja ulkoasusuunnittelu (Knight 2019, luku 1), joita hyödynnetään toimeksiantojen toteuttamisessa. Näiden osaamisalojen avaaminen ei ole kuitenkaan olennaista työssä kuvattavien prosessien osalta. Keskeistä on, että käyttäjälähtöinen suunnittelu ja käyttäjäkokemussuunnittelu ovat olennainen osa nykyaikaista ohjelmistokehitystä, sillä ne auttavat varmistamaan, että lopputuote vastaa käyttäjien todellisia tarpeita ja mieltymyksiä. Tämä lähestymistapa edistää käyttäjätyytyväisyyttä, tehokkuutta ja tuotteen menestystä markkinoilla. Käyttäjäkokemussuunnittelun periaatteet ja menetelmät ovat myös tiiviisti sidoksissa muihin ohjelmistokehityksen lähestymistapoihin, kuten ketterään kehitykseen ja palvelumuotoiluun.

2.3 Ketterä ohjelmistokehitys

Ketterä ohjelmistokehitys on ohjelmistokehitysmenetelmä, joka syntyi 1990-luvun lopulla ja 2000-luvun alussa vastauksena perinteisten, raskaiden ohjelmistokehitysmallien ongelmiin. Ketterä ohjelmistokehitys keskittyy työtä tekeviin ihmisiin ja heidän yhteistyöhönsä. Ratkaisut kehittyvät itseohjautuvien, monialaisten tiimien välisen yhteistyön kautta, hyödyntäen kulloinkin tilanteeseen sopivia käytäntöjä. (Agile Alliance n.d.) Menetelmän keskiössä ovat sen neljä ydinarvoa, jotka 17 ohjelmistokehittämissen ammattilaista kirjasivat vuonna 2001 luodessaan Ketterän ohjelmistokehityksen julistuksen:

- *Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja*
- *Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota*
- *Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja*
- *Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa. (Beck ym. 2001.)*

Ketterän ohjelmistokehityksen julistuksen ja muotoiluajattelun sekä palvelumuotoilun välillä on useita yhtäläisyyksiä. Molemmissa painotetaan asiakkaiden tarpeiden ymmärtämistä ja niihin vastaamista, vaikkakin hieman eri näkökulmista. Ketterässä kehityksessä keskitytään asiakkaan tarpeet täyttävän ohjelmiston toimittamiseen, kun taas muotoiluajattelu ja palvelumuotoilu korostavat käyttäjäkokemusta ja asiakasarvon luomista. Lisäksi kaikissa näissä lähestymistavoissa korostetaan moniammatillista yhteistyötä: ketterässä kehityksessä liiketoiminnan edustajien ja kehittäjien päivittäistä yhteistyötä, muotoiluajattelussa ja palvelumuotoilussa puolestaan eri alojen asiantuntijoiden välistä yhteistyötä. Muutokseen vastaaminen on myös keskeinen teema kaikissa näissä menetelmissä, sillä ratkaisuja kehitetään ja muokataan jatkuvasti saadun palautteen perusteella. Lopuksi, sekä ketterässä kehityksessä että muotoiluajattelussa painotetaan liiketoiminnan menestystä ja kilpailukykyä. Yhtäläisyydet osoittavat, että ketterät ohjelmistokehittämisen menetelmät ja muotoiluajattelun sekä palvelumuotoilun lähestymistavat tukevat toisiaan ja voivat toimia synergiasa ohjelmistokehitysprojekteissa.

2.4 Lean Startup

Lean Startup on Eric Riesin kehittämä lähestymistapa, joka soveltaa lean-ajattelua innovoinnin prosesseihin (Ries 2011, 6) ja se painottaa nopeaa tuotekehitystä, asiakaspalautteen keräämistä ja iteratiivista oppimista. Lean-ajattelussa pyritään jatkuvasti kehittämään arvon tuotantoa asiakkaille tunnistamalla ylimääräistä resurssien kulutusta työprosesseissa ja minimoimalla tätä tuhlausta (Ries 2011, 47). Tämä resurssien käytön tehostaminen yhdessä mahdollisimman varhaisen oppimisen maksimoinnin kanssa muodostaa Lean Startup -menetelmän keskeisen ytimen.

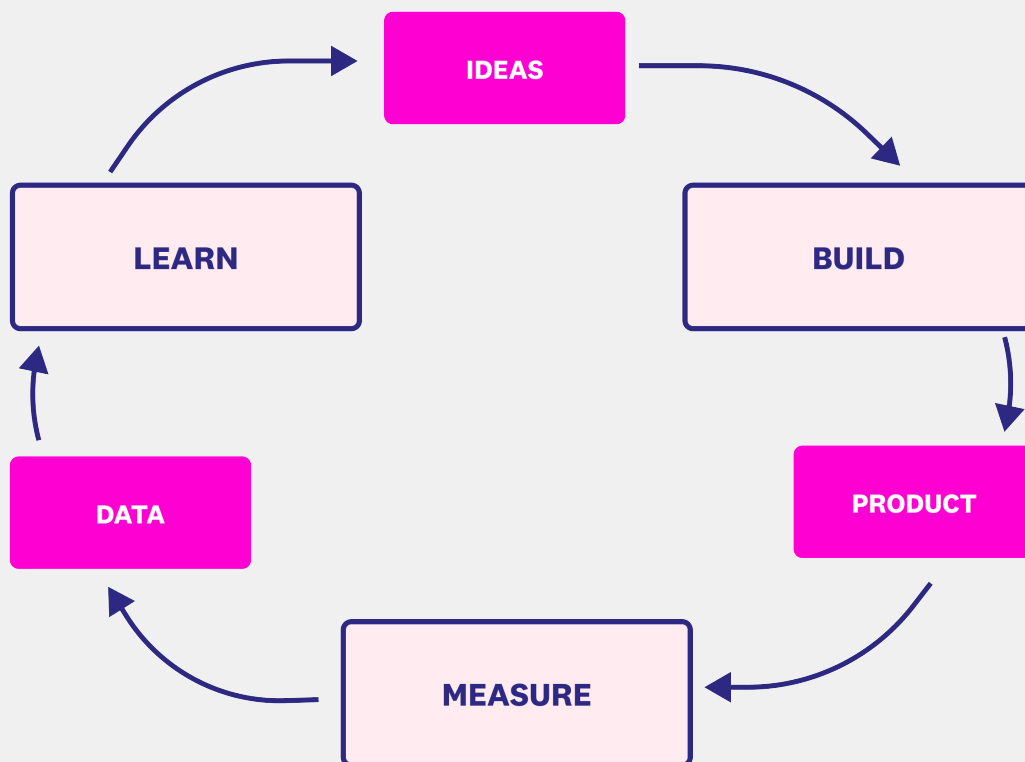
Keskeinen osa Lean startup -menetelmää on "rakenna-mittaa-opi" -sykli (englanniksi *build-measure-learn loop*). Tämä iteratiivinen prosessi koostuu kolmesta vaiheesta:

1. Rakenna: Kehitä minimaalinen toimiva tuote (MVP, eli *Minimum*

Viable Product) tai prototyyppi mahdollisimman nopeasti ja pienillä resursseilla.

2. Mittaa: Testaa tuotetta oikeilla käyttäjillä ja kerää dataa sen toimivuudesta ja käyttäjien reaktioista.
3. Opi: Analysoi kerättyä dataa ja tee sen perusteella päätöksiä jatkokehityksestä tai suunnan muutoksesta.

Tämän syklin tarkoituksena on nopeuttaa oppimisprosessia ja vähentää resurssien hukkaamista kehittämällä tuotteita, joille ei ole todellista kysyntää. Sykli mahdollistaa nopean palautteen saamisen ja tuotteen jatkuvan kehittämisen asiakkaiden todellisten tarpeiden mukaisesti. (Ries 2011, 75–78.) Prosessi on kuvattu kuvassa 2. Lean Startup pyrkii yksinkertaistamaan lean- ja systeemiajattelun periaatteita ennen kaikkea liiketoimintaorientoituneille yleisöille enemmän kuin suunnittelijoille (Wendt 2015, 72), mutta tämä voidaan nähdä menetelmän käytäntöönvientiä helpottavana tekijänä.

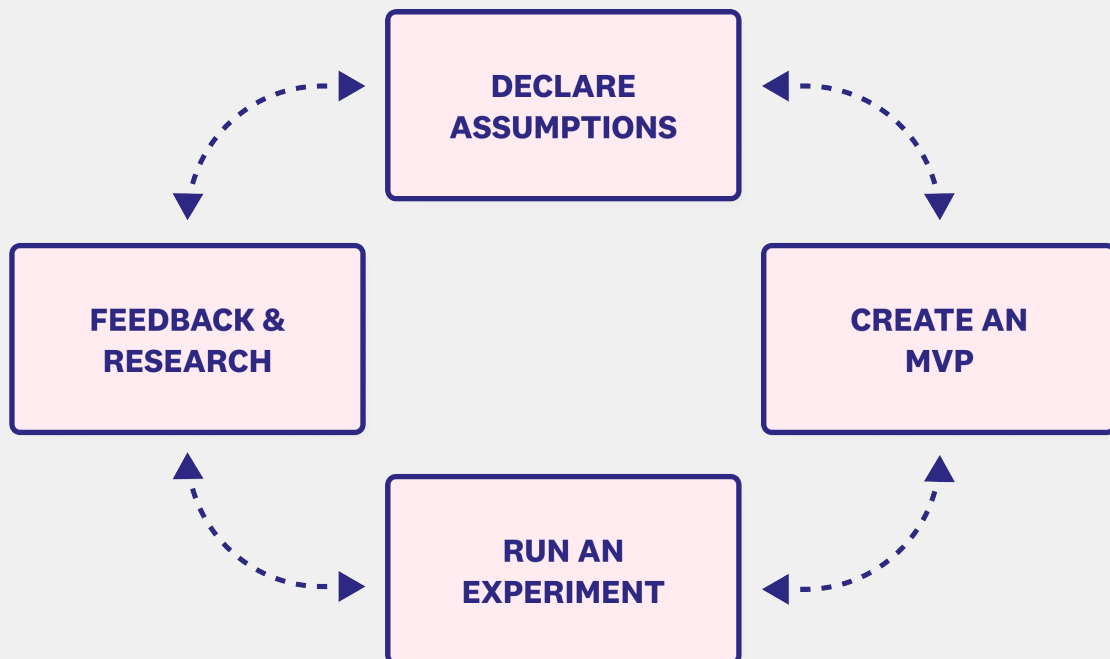


Kuva 2. Rakenna-mittaa-opi -sykli (Mukaelma: Ries 2011, 75).

2.5 Lean UX

Lean UX on lähestymistapa käyttäjäkokemuksen suunnitteluun, joka yhdistää Lean-ajattelun periaatteet ja ketterän kehityksen menetelmät. Se painottaa nopeaa iterointia, jatkuvaa käyttäjäpalautetta ja MVP:n kehittämistä. Lean UX:n ytimessä on kolmen metodologian muodostamat peruspilarit: muotoiluajattelu, ketterä ohjelmistokehitystä ja Lean startup.

Muotoiluajattelusta Lean UX ammentaa muotoilumenetelmien hyödyntämistä kokonaisvaltaisesti kaikille liiketoiminnan osa-alueille ja moniammatillisesti toimimisesta ongelmien ratkaisemiseksi. Ketterän ohjelmistokehityksen neljän ydinarvon tavoitteet lyhentää toimitussyklejä ja toimittaa jatkuvaa arvoa asiakkaille muodostavat Lean UX:n toisen peruspilarin. Lean Startupista hyödynnetään useita menetelmiä, kuten oletusten määrittely suunnitteluvaihetta aloitettaessa sekä jatkuvalla yhteydellä asiakkaaseen tehtävää ratkaisujen validointia kehittämällä prototyyppisiä ja minimaalisia toimivia tuotteita minimoidaksesi hukkatyötä, joka ei johda tavoitteiden saavuttamiseen. (Gothelf & Seiden 2013, luku 2.)



Kuva 3. Lean UX -mallin vaiheet (Mukaelma: Gothelf & Seiden 2013, luku 3).

Lean UX -malli koostuu neljästä vaiheesta, jotka on kuvattu kuvassa 3. Ensimmäisessä vaiheessa kehitystiimi määrittää oletuksensa siitä, miten he ovat käsiteltävää ongelmaa ratkaisemassa ja kenelle. Tiimi asettaa yhdessä priorisoiden hypoteesin ratkaisusta ja tavoitelluista lopputuloksista, joiden toteutumista prosessissa testataan. (Gothelf & Seiden 2013, luku 3.)

Oletusten määrittämisen jälkeen seuraa Lean Startup -mallin mukainen rakentamisvaihe, jossa tiimi kehittää MVP:n hypoteesinsa testaamiseksi. MVP voi olla monenmuotoinen – esimerkiksi paperille luonnosteltu käyttöliittymäkuvaus, myyntipitchilla varustettu verkkosivu, johon kerätään ideasta kiinnostuneiden yhteystiedot tai koodattu sovellusprototyyppi. MVP:n tärkein tehtävä on todentaa tiimin ratkaisuehdotuksen tarpeellisuus, käytettävyys ja kyky tuottaa tavoiteltua arvoa käyttäjille. Tätä tiimi testaa mallin kolmannessa vaiheessa toteuttamalla kokeita MVP:n avulla. (Gothelf & Seiden 2013, luku 5.)

Viimeiseksi tiimi kerää palautetta ratkaisuehdotuksesta käyttäjiltä. Tämä voi tapahtua esimerkiksi käyttäjähaastatteluin tai käyttäjätestaussessioita toteuttaen. Vaiheen tavoitteena on todentaa käyttäjillä, että ratkaisuehdotus toimii. (Gothelf & Seiden, luku 6.) Tämä vastaa Lean Startup -mallissa validoidun oppimisen vaihetta.

Lean UX -mallin vaiheet eivät rajaudu selkeiksi kokonaisuuksiksi, vaan kuviossa 3 kuvatun mukaisesti vaiheiden välillä on liukumaa, ja eri vaiheissa voi syntyä iterointia prosessin aikana. Esimerkiksi MVP:n kehittämisen yhteydessä voidaan palata uudelleenmäärittelemään oletuksia, kun toteutuksessa tehdään uusia havaintoja. Prototyypin testausvaiheessa MVP:nä toimivaa prototyyppiä voidaan kehittää edelleen. Tämä noudattaa ketterien menetelmien periaatetta, jossa mallin toteuttamisessa vastataan joustavasti muutoksiin.

2.6 Shape Up -malli

Shape Up on yhdysvaltalaisen Basecampin nimitys omalle tuotekehitysprosessilleen. Kuten ketterässä kehityksessä laajasti käytetyssä Scrum-

mallissa, Shape Upissa tavoitteena on toimittaa toimivia ohjelmistotoiminnallisuuksia rajatun aikaikkunan, Shape Upissa nimeltään sykli, puitteissa. Shape Up -malli koostuu kolmesta vaiheesta:

1. Muovaus (*Shaping*)

Ennen kuin ominaisuus on valmis vietäväksi kehityssykliin, sen tarve tunnistetaan esimerkiksi asiakaspalautteen tai muilla keinoilla liiketoiminnassa havainnoiden. Muovauksen aikana ominaisuudet määritellään sellaisella tasolla, että syklissä toteutettavan ratkaisun keskeiset elementit voidaan kuvailla yleisellä tasolla tiimille. Lisäksi muovausvaiheessa pyritään tunnistamaan kehitysvaiheen riskejä, jotta kehitystiimi voi välttää ajan tuhlaamista ja rajata toteutustaan. (Singer 2019, 14–15.)

2. Panostus (*Betting*)

Kun tarve on muovattu esiteltäväksi ominaisuudeksi, muovaustiimi tekee ominaisuudelle pitchin, eli myyntipuheen, joka esitellään kehitystiimeille. Näiden pitchien perusteella yritys tekee päätöksen siitä, mihin kehityskohteisiin sillä on eniten halukkuutta panostaa alkavassa syklissä. Panostusvaihe on kriittinen osa Shape Up -mallia, sillä se määrittää, mihin projekteihin ja ominaisuuksiin tiimit keskittyvät seuraavan syklin aikana. Tämä vaihe mahdollistaa resurssien tehokkaan kohdentamisen ja varmistaa, että tiimit työskentelevät sellaisten projektien parissa, joilla on suurin potentiaalinen arvo yritykselle ja asiakkaille. (Singer 2019, 73–78)

3. Rakentaminen (*Building*)

Rakentamisvaiheessa kehitystiimi työskentelee itsenäisesti toteuttaakseen valitun ominaisuuden syklin aikana. Tiimi hyödyntää muovausvaiheessa määriteltyjä rajauksia ja keskeisiä elementtejä ohjaamaan työtään. Tässä vaiheessa

korostuu tiimin kyky tehdä itsenäisiä päätöksiä ja ratkaista ongelmia luovasti annetun ajan puitteissa. (Singer 2019, 96–97.)

Kehitysykleihin sisällytettävää työmäärää ei arvioida, vaan kehitettävälle toiminnallisuuksille määritetään muovausvaiheessa ennen sykliä keskeiset elementit ja rajaukset. Ennen sykliä muovatuista toiminnallisuuksista annetaan pitchit, eli myyntipuheet, ja tiimit päättävät yhdessä, mihin toiminnallisuuksiin syklin aikana panostetaan. Toiminnallisuudet kehittävä tiimi on tämän jälkeen vastuussa toiminnallisuuden toimittamisesta kuudessa viikossa. Mallissa korostetaan tiimin autonomiaa ja vastuuta lopputuloksesta, eli tiimi saa päättää, miten he toteuttavat ominaisuudet annetussa ajassa. Tämä kannustaa tiimiä innovatiivisiin ratkaisuihin ja työskentelemään tehokkaasti.

Shape Up -malli pyrkii tasapainottamaan kehitystyön joustavuuden ja ennustettavuuden. Se tarjoaa selkeän rakenteen tuotekehitykselle, mutta antaa tiimeille vapauden päättää, miten he saavuttavat tavoitteensa syklin aikana. Tämä lähestymistapa kannustaa innovointiin ja ongelmien luovaan ratkaisuun, samalla kun se pitää projektin aikataulussa ja budjetissa.

2.7 Keskeiset käytettävät työkalut ja teknologiat

Tämä työ ei pyri kuvaamaan tarkasti toimeksiantojen teknisen kehityksen vaiheita. Koska kyseessä on ohjelmiston muotoilusta, on kuitenkin tärkeää avata tiettyjä yleisiä työkaluja ja teknologioita, joita muotoilussa ja toimeksiantojen toimituksessa on hyödynnetty.

2.7.1 Figma

Figma on ohjelmistosovellus, jolla voidaan mallintaa ohjelmistojen käyttöliittymiä sekä niissä tapahtuvia interaktioita. Sovellus mahdollistaa myös yhteiskäytön, jolloin useampi käyttäjä voi samanaikaisesti tehdä muutoksia käyttöliittymäkuviin.

Figma sisältää myös FigJam-työkalun, joka mahdollistaa virtuaalisen

työtilatyöskentelyyn, jossa käyttäjät voivat yhteisessä sessiossa jakaa muistilappuja, piirtää yksinkertaisia kuvioita ja kaavioita tai kirjoittaa muistiinpanoja. Sakeus hyödyntää FigJamia virtuaalisissa työpajoissa kun mahdollisuutta työskennellä samassa tilassa ei ole. Tässä työssä nähtävät käyttöliittymäkuvat ja kuviot työpajoista ovat Figmasta ja FigJam-työtiloista.

2.7.2 Ruby on Rails

Sakeus kehittää ohjelmistonsa pääsääntöisesti Ruby on Railsilla (myöhemmin tekstissä Rails). Rails on David Heinemeier Hanssonin kehittämä Ruby-ohjelmointikieltä hyödyntävä avoimen lähdekoodin verkkosovelluskehys. Hanson kehitti Railsin kehittäjäystävälliseksi ympäristöksi kehittää sovelluksia. Se pyrkii mahdollistamaan selkeän ja luettavan koodin kirjoittamisen, jotta koodi on ymmärrettävää ja ylläpidettävää, ja sen suunnittelussa on painotettu kehittäjäkokemusta ja tehokkuutta kehittää (Hansson n.d.), minkä vuoksi se on valikoitunut Sakeuden käyttöön.

Työssä kuvatuissa toimeksiannoissa toteutettavat käyttöliittymät on toteutettu Rails-näkymiksi, eli käyttöliittymään ei ole käytetty kokonaan erillistä käyttöliittymäkehystä, kuten React, joka mahdollistaisi esimerkiksi dynaamisesti tietojen hakemista ohjelmiston tietokannasta ja niiden näyttämistä käyttäjälle ilman, että käyttäjän tarvitsee ladata sivua uudelleen. Sen sijaan ohjelmistossa käytetään Stimulusta, joka on käyttöliittymäkehys, jolla voidaan kehittää dynaamisesti tietoa hakevia osia Rails-sovelluksen näkymiin ilman koko sovelluksen kattavaa käyttöliittymäkehystä (Broadbent 2023). Stimuluksen käyttöä tehdään kuitenkin tarkoin harkiten, koska Stimulus-osien kehittämiseen vaadittava työpanos voi olla moninkertainen vaihtoehto ja jotta koodikanta pysyy hallittavana.

2.7.3 Tailwind CSS

Käyttöliittymien visuaalisen ilmeen toteutuksessa on käytetty Tailwind CSS:aa (myöhemmin tekstissä Tailwind). Tailwind on kehys, joka sisältää ennalta asetetut CSS-luokat verkko-ohjelmistojen ulkoasutyöliien

määrittämiseen (Tailwindcss n.d.), jolloin suunnittelija voi käyttöliittymää kehittäessä käyttää näitä CSS-luokkia suoraan näkymien HTML-elementteihin sen sijaan, että ulkoasumäärittelyä tehdään erilliseen CSS-tiedostoon. Tällöin ulkoasutyylit pysyvät rajattuina näkymäkohtaisesti, mikä estää sen, että yhdessä näkymässä tehtävä virhe ulkoasun tyylittelyssä vaikuttaisi muualle ohjelmistossa.

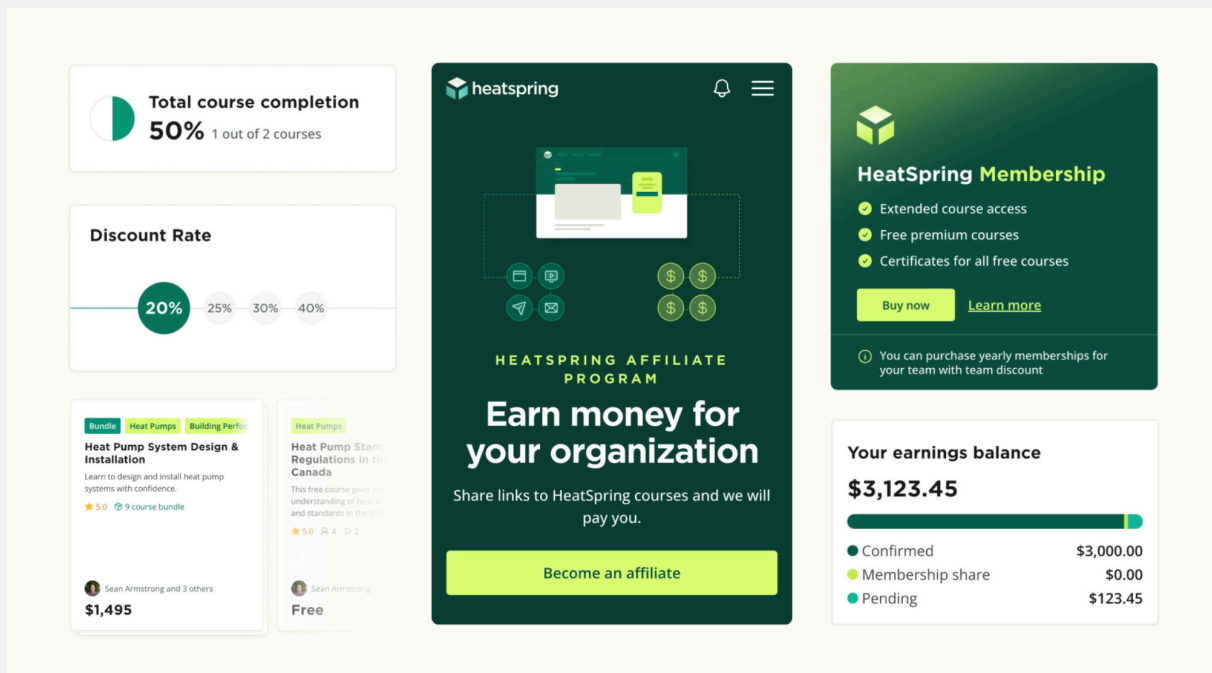
3 Toimeksiannot

Tässä luvussa käsitellään työn aineistona toimivaa kahta toimeksiantoa ja niiden asiakkaana toimivaa HeatSpringia. Toimeksiantojen käsittely aloitetaan tarkastelemalla HeatSpringin taustaa ja sen roolia asiakkaana. Tämän jälkeen syvennytään kahteen keskeiseen toimeksiantoon, jotka muodostavat työn empiirisen aineiston ytimen. Näiden toimeksiantojen kautta pyritään havainnollistamaan Sakeus Oy:n muotoiluprosessia käytännössä ja sen mukautumista asiakastarpeisiin sekä toimeksiantojen aikana tehtävien havaintojen mukaan. Ensimmäinen toimeksianto keskittyy HeatSpringin Team Dashboard -ominaisuuden uudistamiseen. Toinen toimeksianto koskee HeatSpringin kouluttajien työkalujen uudistamista.

3.1 HeatSpring asiakkaana

HeatSpring on vuonna 2007 perustettu yhdysvaltalainen koulutusalan yritys, joka sovelluksensa kautta tarjoaa puhtaan energian sertifiointeihin johtavia kursseja. Palvelussa on tarjolla yli 300 kurssia ja sillä on yli 112 000 rekisteröitynyttä käyttäjää. (HeatSpring n.d.)

Sakeus on toiminut HeatSpringin keskeisenä suunnittelu- ja kehittäjäkumppanina, vastaten alustan uudistamisesta ja uusista ominaisuuksista vuodesta 2022 lähtien. Alussa Sakeuden tiimi toteutti HeatSpringille ensisijaisesti brändiuudistuksen loppuunviennin. Tämän jälkeen tiimi on toteuttanut muun muassa sähköpostiviestinnän automatisointia, tiiminvetäjien ja kouluttajien toiminnallisuuksien jatkosuunnittelua ja kehitystä sekä muita myynninedistämiseen ja käyttäjäkokemuksen parantamiseen tähtääviä ominaisuuksia. (Sakeus n.d.)



Kuva 4. Näyte Sakeuden suunnittelemista käyttöliittymäkomponenteista.

Tarvetta Sakeuden palveluille loi vuonna 2001 alkanut koronaviruspandemia, joka muutti HeatSpringin toimintamalleja. Aiemmin alusta toimi ensisijaisesti ostamis- ja ilmoittautumisalustana kursseille, joissa kurssin kävijät olivat läsnä. Koronaviruspandemian myötä alusta on muuttunut tarjoamaan vain virtuaalisesti toteutettuja kursseja, jotka kurssin ostaja voi suorittaa omaa tahtia. Tämä toiminnan painopisteen muutos on lisännyt alustan käyttöastetta ja HeatSpring on asettanut palvelun käyttökokemuksen kehittämisen aiempaa korkeammaksi prioriteetiksi.

3.2 HeatSpring – Team Dashboard

HeatSpring tarjoaa käyttäjilleen mahdollisuuden muodostaa samassa yrityksessä työskentelevien henkilöiden käyttäjätileistä tiimejä, minkä kautta käyttäjille voidaan tarjota tiimikohtaisia etuja ja työkaluja. Tiimit hyötyvät vuosittaisiin ostomääriin perustuvista alennuksista sekä keskitetystä laskutuksesta ja yksinkertaistetusta ostoprosessista. Tiiminvetäjät pystyvät seuraamaan tiimensä jäsenten edistymistä ja hallinnoimaan tiimin jäsenyyksiä heille kehitettyjen työkalujen avulla.

Team Dashboardin kautta.

Sakeus oli uudistanut Team Dashboardin kerran aiemmin ennen vuoden 2024 uudistusta. Ennen ensimmäistä uudistusta Team Dashboard oli HeatSpringin sisäisesti toteuttama näkymä, joka listasi tiimiin kuuluvat käyttäjät ja kurssit, joille tiimin jäsenet ovat ilmoittautuneet (kuva 5). Tiiminvetäjäkäyttäjä pystyi myös lataamaan raportin tiimin jäsenten suorituksista, tarkastelemaan kurssin etenemistä sekä käyttäjiän ansaitsemia kurssisuoritussertifikaatteja. Sakeuden Henri Nikka suunnitteli ja toteutti brändiuudistuksen yhteydessä uuden version Team Dashboardista 2023 (kuva 6).

team foo Team Page

Generate Training Report

Teammates and Administration

Team Members

Teammate	Unique ID	Team Status	Notified	Total Courses
		Confirmed	10/12/2022	1
		Confirmed	10/12/2022	1
		Confirmed	10/12/2022	1
		Confirmed	10/12/2022	1

You can contact Brian Hayden at bhayden@heatspring.com to add additional people to this team.

Last Enrollment: 10/12/2022
Training Leader: Faustino Blogovich
Total Team Members: 4

View Learning Paths (beta)

Paid Course Enrollments

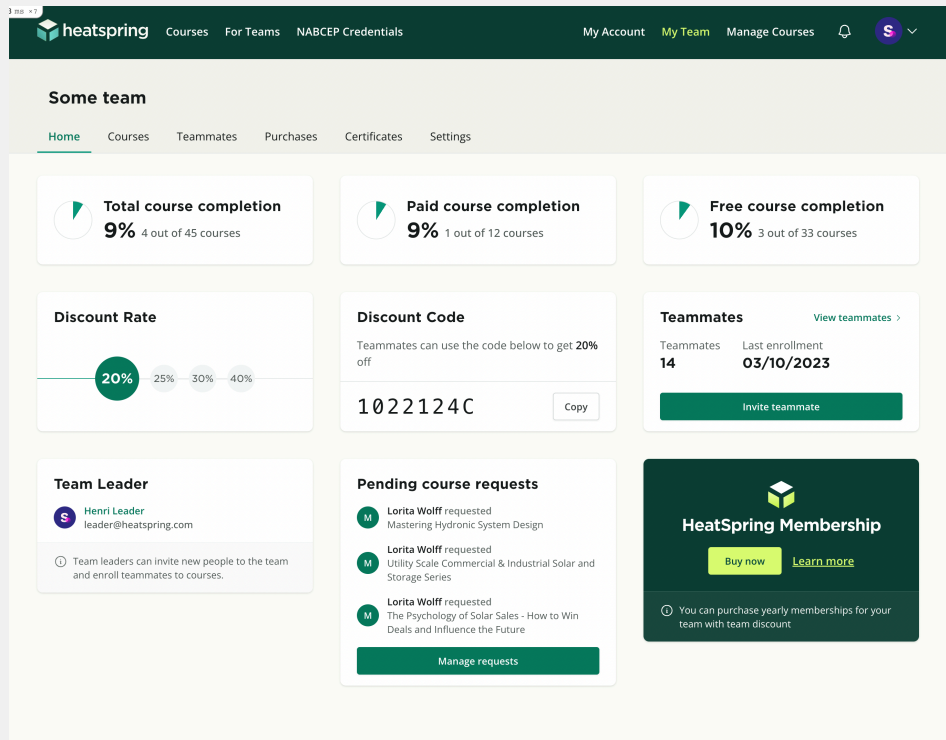
fake course 23

Teammate	Enrolled	Last Visit	Completed Assignments	New / Incomplete Assignments	Progress	Cost	Certificate Earned
	10/12/2022		0	6	view		
	10/12/2022		0	6	view		
	10/12/2022		0	6	view		
	10/02/2022		0	6	view	\$100.00	10/12/2022

Total Team Course Enrollments: 4
Total Active Course Assignments: 6
[Download Spreadsheet](#) | [View Course Information](#)

Kuva 5. Team Dashboard vuonna 2022. Käyttäjätiedot sumennettu.

Vuoden 2023 Team Dashboard (kuva 6) toteutettiin suoraviivaisesti hyödyntämään järjestelmässä jo saatavilla olevaa dataa, joka pystyttiin helposti esittämään numeraalisessa muodossa käyttäjille. Tämä koostui pääasiassa tiimin jäsenten kurssisuoritusasteeseen. Näkymän sisältöjä, kuten listaus tiimin jäsenistä ja kurssihankintahistoria, siirrettiin omille välilehdilleen. Näkymään lisättiin myynnin edistämiseksi helposti kopioitava tiimikohtainen alennuskoodi ja heräte ostaa useita vapaasti suoritettavia sisältävä kausimaksullinen HeatSpring Membership -palvelu.



Kuva 6. Team Dashboard vuonna 2023.

Tiimiominaisuuksilla on merkittävä rooli HeatSpringin liiketoiminnalle. Suurin osa alustan käyttäjistä suorittaa kurssit työnantajinsa vaatimuksesta ja rahoittamina. Tiimiominaisuuksien avulla yritysasiakkaille mahdollistetaan koulutushankintojen tuottaman arvon validointia. HeatSpringin henkilöstö keräsi palautetta Sakeuden toteuttamista uudistuksista vuoden 2023 ja alkuvuoden 2024 aikana. Team Dashboardille lisätty kurssien suoritusaste sai käyttäjät kokemaan, että palvelusta on vain vähän hyötyä, koska suoritusaste oli lähes kaikilla tiimeille matala. Yksi tähän vaikuttava tekijä oli, että ilmaiskurssit vaikuttivat kokonaissuoritusasteeseen ja useat käyttäjät aloittivat kurssin vain silmäillääkseen kiinnostaako kurssin sisältö heitä tai heitä kiinnosti vain yksittäinen osa kurssin sisällöstä. Lisäksi suoritusaste huomioi tiimin koko käyttäjähistorian, mikä saattoi vaikuttaa mataliin lukuihin. Matalat suoritusasteet nähtiin kuitenkin riskinä sille, että yritysten korkeampi johto ei näkisi HeatSpringin kautta tehtävissä koulutusinvestoinneissa riittävästi arvoa jatkaakseen palvelun käyttäjinä. HeatSpringin käyttäjä kuvasi palautteessa Team Dashboardin ongelmia seuraavalla tavalla:

"A lot of people on our team will sign up for free classes or practice exams just to check them out or get a little bit of information. But now our Team Dashboard looks really sad. Our course completion rates are really low. Can we just clean up all of those old courses somehow?"

HeatSpringin perustaja Brian Hayden summasi Team Dashboardin haasteet seuraavalla tavalla:

"Course completion percentages are not useful! They are sad and not actionable. Let's change the Team Dashboard so it shows useful, actionable information that is current! It's going to be awesome and I'm super excited to do this. I've been suspiciously not-proud when I pull up a Team Dashboard and show it to somebody. Let's change that and make them awesome."

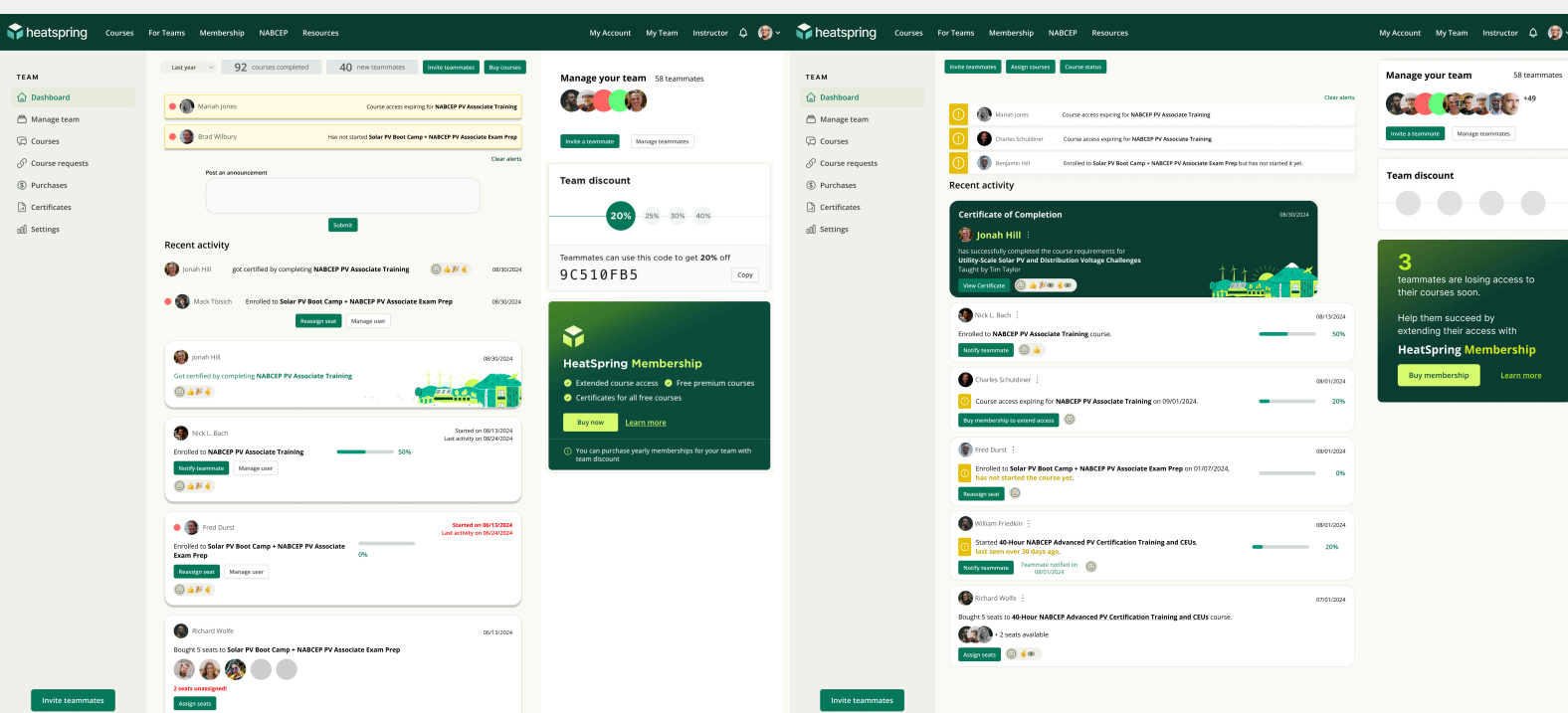
Palautteiden myötä Team Dashboardin uudelleensuunnittelusta tehtiin päätös asiakkaan ja Sakeuden tiimin yhteisessä viikottaisessa kehityspalaverissa maaliskuussa 2024. Haydenin palautteesta nostimme uudelle Team Dashboardille kolme korkean tason suunnitteluohjuria: sen tulee olla hyödyllinen, toimintaan ohjaavaa ja ajantasainen. Suunnitteluohjurit ovat suunnittelua ohjaavia määrittelyjä, jotka kiteyttävät asiakkaiden tarpeet, tavoitteet ja motivaatiot pitämään suunnittelufokuksen selkeänä (Tuulaniemi 2021, 156). Näiden suunnitteluohjureiden pohjalta lähdimme ideoimaan uusia ratkaisuja Team Dashboardin sisältöön ja rakenteeseen. Tavoitteena oli luoda näkymä, joka tarjoaisi tiiminvetäjille selkeän ja ajantasaisen kuvan tiimin toiminnasta sekä mahdollistaisi tehokkaan päätöksenteon ja toiminnan ohjaamisen.

3.2.1 Suunnittelutyö

Sekä loppuasiakkailta että HeatSpringilta kerätyn palautteen pohjalta teimme yhdessä asiakkaan kanssa seuraavia oletuksia pohjustamaan ideointia ja käyttöliittymän suunnittelua:

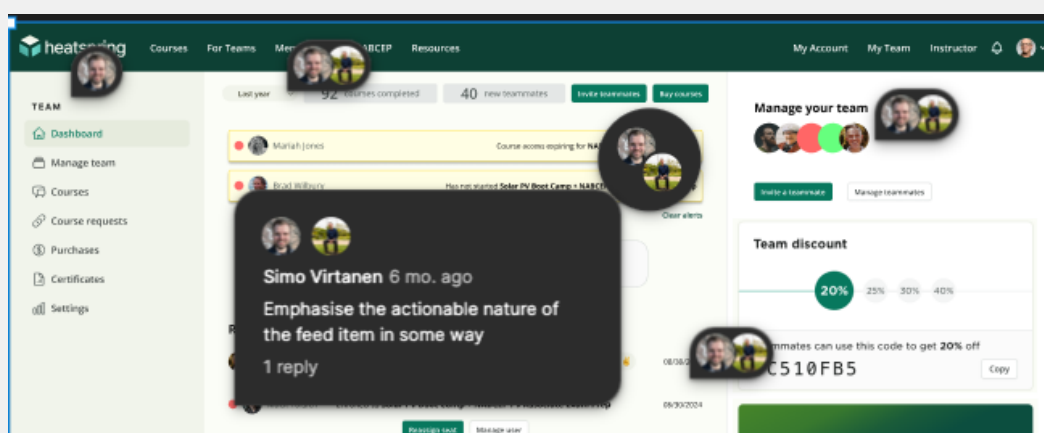
- Suoritusprosentit ja muut näytettävät metriikat eivät ole hyödyllisiä. Mitä käyttäjä tekee tiedolla, että mikä osuus kurseista on jätetty suorittamatta? Miten tämä ohjaa käyttäjää?
- Tiimin jäsenten aktiivisuutta ja osallistumista kurseille tulisi korostaa positiivisella tavalla. Näin voidaan kannustaa tiimin jäseniä jatkamaan oppimistaan ja osallistumistaan.
- Tiiminvetäjille tarvitaan työkaluja, joilla he voivat tukea ja ohjata tiimin jäseniä oppimisprosessissa. Tällaisia voivat olla esimerkiksi keinot havaita kun tiimin jäsenet eivät ole edistyneet kurseissaan tai kun pääsyoikeus kurssille on umpeutumassa, jolloin tiiminvetäjä voi toimia ja motivoida tiimin jäseniä suorittamaan kurssinsa.

Ideointia ja suunnittelutyötä lähdettiin toteuttamaan yhteiskehittämismallilla siten, että ideoita kerättiin tiimin käyttämään Asana-tehtävähallintatyökaluun ja suunnittelussa mahdollistettaisiin kommentointi suunnitelmiin niin pian kuin mahdollista. Koska Sakeus oli tehnyt HeatSpringin kanssa jo aiempaa yhteistyötä, oli suunnittelutyöhön Figmaassa käytettävissä valmiita käyttöliittymäelementtejä aiemmista suunnittelutiedostoista.



Kuva 7. Käyttöliittymäprototyyppien versioita.

Tämä mahdollisti nopeamman siirtymisen korkeamman tarkkuustason käyttöliittymäkuvauksiin, mikä konkretisoi ideoita rautalankamalleja paremmin HeatSpringin henkilöstölle. Simo Virtasen toteuttamien käyttöliittymäluonnosten tavoitteena oli prototypoida Team Dashboardin uutta rakennetta, siihen sisällytettävää informaatiota ja mahdollistaa palautteen kerääminen ja päätöksenteko toteutettavista toiminnallisuuksista. Kuten Brown (2019, 98.) kirjoittaa, tulee prototyyppeihin panostaa vain sen verran aikaa ja resursseja, kuin palautteen keräämiseksi ja idean eteenpäin viemiseksi on tarpeen, joten toteutuksessa pyrittiin välttämään yksityiskohtien hiomista.



Kuva 8. Kommentointia Figma-työkalussa

Ensimmäisiin käyttöliittymäkuvauksiin kerättiin asiakaspalautetta HeatSpringilta Figma-kommentointi-ominaisuutta käyttäen (kuva 8). Koska HeatSpring toimii Yhdysvalloissa, asynkroninen viestintä on aikaeron vuoksi edellytys kommunikation toimivuudelle. Figmalla työskentely mahdollistaa HeatSpringille suunnitelmien kommentoinnin silloin kun se heille itselleen parhaiten sopii. Lisäksi se mahdollistaa, että Sakeus voi antaa lisäkontekstia suunnitelmille ja kuvaamaan asiakkaalle miksi kuvattua ratkaisua ehdotetaan. Käyttöliittymän rakenteeseen otettiin mallia aiemmin uudistetuista ylläpitäjäkäyttäjän näkymistä, mikä suoraviivaisti toteutusta.

Ratkaisussa keskityttiin tekemään tiimin jäsenten toiminta HeatSpringissa näkyväksi. Tähän ratkaisuksi ehdotettiin sosiaalisen median palveluissa konventiksi muodostunutta sisältösyötettä. Tämän toimivuudesta esitettiin hypoteesi, että sisältösyötteen konsepti olisi laajan

levinneisyytensä ansiosta käyttäjille entuudestaan tuttu ja helposti käyttöönotettava tapa esittää tietoa. Käyttäjien suorittamat aktiviteetit, kuten tiimiin liittyminen, kurseissa eteneminen ja kurssihankintapyyntö tuotaisiin erityisesti tiiminvetäjille helposti nähtäville. Tämä johti kahden erilaisen aktiviteettityypin määrittämiseen, joista toinen edellyttää tiiminvetäjältä toimintaa ja toinen ei. Toimintaa edellyttävät tulevat näkyviin vain tiiminvetäjille erillisenä syötteenä kaikille näkyvien sisältöjen yläpuolelle. Tällaisia herätteitä toimintaan ovat esimerkiksi ilmoitukset tiimiläisen kurssin käyttöoikeuden tai vuosimaksullisen jäsenyyden umpeutumisen sekä heräte siirtää käyttöoikeus kurssiin toiselle tiimiläiselle, mikäli alunperin kurssille ilmoitettu käyttäjä ei ole aloittanutkaan kurssia. Näin pyrittiin siihen, että Team Dashboard kykenisi kerralla vastaamaan jokaiseen sille määritellyyn suunnitteluohjuriin.

Team Dashboardin lisäksi myös muista tiiminäkymistä kuvattiin Figmaan päivitettyt näkymät, koska uudelleensuunnittelun myötä näkymien rakenne tulisi muuttumaan. Toteutettujen puolivalmiiden käyttöliittymäkuvien pohjalta pystyttiin yhdessä HeatSpringin kanssa tekemään päätökset siitä, mistä toiminnallisuuksista rakentaminen voidaan aloittaa ja mitä ideoita jätettäisiin toteutuksesta pois, jotta Team Dashboardin käyttöä voidaan lähteä koestamaan käytännössä. Käyttöliittymän ilmettä kohennettaisiin vielä myöhemmässä vaiheessa lisää.

3.2.2 Rakentaminen

Team Dashboardin uudistamisen tekninen toteutus alkoi noin kaksi viikkoa suunnittelun aloittamisesta. Virgil Mocanu vastasi taustajärjestelmään tehtävästä ominaisuuskehityksestä ja Simo Virtanen käyttöliittymästä. Samalla kun toiminnallisuuksia toteutettiin, Henri Nikka sai vastuulleen iteroida käyttöliittymäluonnoksista lopulliset versiot, jotka toimisivat lopullisen käyttöliittymän tavoitereferenssiä, jotta uudistettu käyttöliittymä on visuaaliselta ilmeeltään linjassa sovelluksen muiden näkymien kanssa.

Kehitystyön prosessina oli rakentaa uudet toiminnallisuudet niin, että kokonaisuus pilkottiin osiin, jossa sekä taustajärjestelmän että

käyttöliittymän kehitystä tehtiin kunkin osan kohdalla samanaikaisesti. Näin kutakin ominaisuutta pystyttiin testaamaan asiakkaan toimesta kun kehitys siirtyy muihin ominaisuuksiin. Singer (2019, 104) kuvaa tilannetta, jossa käyttöliittymä ja taustajärjestelmä eivät ole vielä kytköksissä toisiinsa, hypoteettiseksi ja spekulatiiviseksi tilaksi, koska ominaisuuksien toimivuutta ja liiketoiminnallista hyötyä ei ole mahdollista todentaa. Kun ominaisuuksien toimivuuteen keskitytään siivuissa, saadaan toimivia ja testattavia ohjelmiston osia sekä tehdään projektin eteneminen näkyväksi. Virgil Mocanu loi toiminnallisuuksille Rails-näkymät, joiden määritellyt sisällöt Mocanu tarkisti Figma-prototyypistä (Kuva 9). Kun näkymiin saatiin luotua dataa, Simo Virtanen tyyllitteli ne käyttäen Tailwind-luokkia. Vaikka tyyllittelyä ei olisikaan tehty, oli toiminnallisuutta mahdollista jo testata ja kerätä palautetta.

Action Required



Amelia Bergbower * 3 hours ago

Requested a course [OSHA 10-Hour Construction](#).

[Action link](#)

Team Activity



Virgil Mocanu * 9 minutes ago

Bought 1 seat to [Leadership in Utility-Scale Solar Construction \[Career Bridge Program\]](#)



Virgil Mocanu * 2 hours ago

Started [40-Hour NABCEP Advanced PV Certification Training and CEUs](#)



Brent Vaughn * 3 hours ago

Completed and earned a certificate for [18-hour Solar PV Boot Camp](#) taught by Sean White



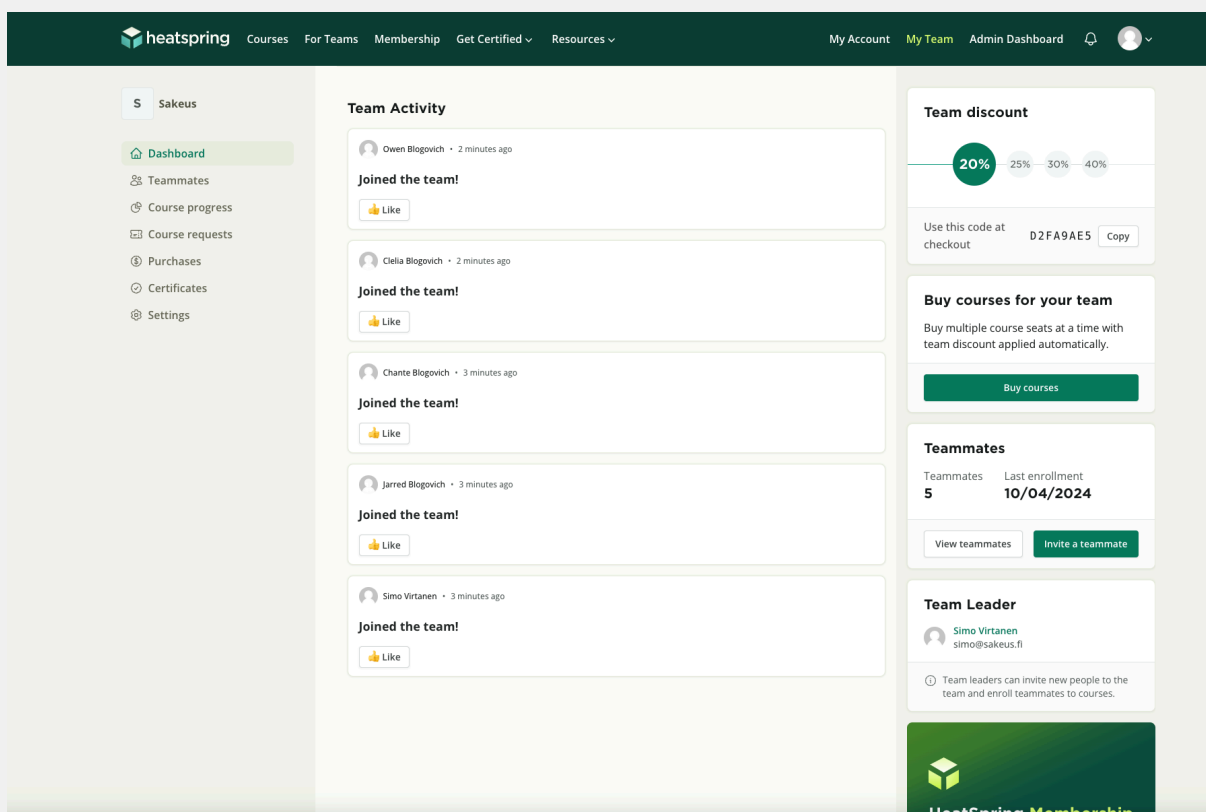
Simo Virtanen * 3 hours ago

Joined the team!

Kuva 9. Tiimin aktiivisuussyöte ennen ulko-osun tyylien asettamista

Kun ominaisuuksia saatiin toteutettua sekä taustajärjestelmän että käyttöliittymän osalta, julkaistiin ne HeatSpringin henkilöstön testattaviksi verkossa saatavilla olevaan testiympäristöön, missä heillä

oli mahdollisuus testata toiminnallisuuksia asettuen itse loppukäyttäjien rooliin. Näin pystyttiin hyödyntämään HeatSpringin kokemusta asiakkaistaan toiminnallisuuksien hyödyllisyyden validoinnissa. Jotta ensimmäinen versio saatiin nopeasti testattavaksi, hyödynnettiin aiemman Team Dashboardin version käyttöliittymäelementtejä lähes sellaisenaan. Lisäksi pieniä ominaisuuksia lisättiin jo ensimmäiseen versioon kehityksen aikana, kuten useissa sosiaalisen median palveluissa hyödynnetty reagointi-ominaisuus aktiviteetteihin, jolla pyrittiin vastaamaan tiiminjäsenten toiminnan positiivisen korostamisen suunnitteluohjuriin. Kun toiminnallisuudet ja Nikan Figmaassa viimeistelemät käyttöliittymien ulkoasut saivat hyväksynnän, toteutti Nikka viimeistelyt sovellukseen ja ominaisuus julkaistiin loppukäyttäjien käytettäväksi.

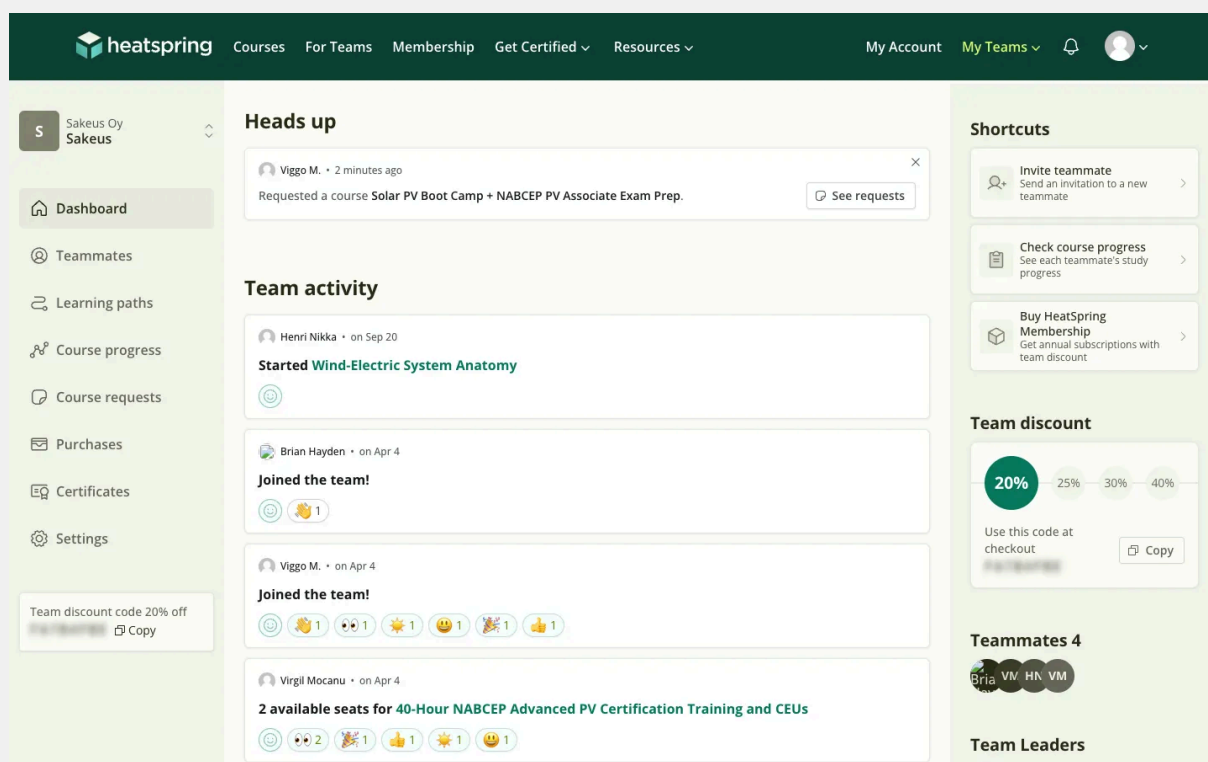


Kuva 10. Uusitun Team Dashboardin versio 1.

3.2.3 Tulokset ja retrospektiivi

Viimeistelty Team Dashboard, esitettynä kuvassa 11, sai loppukäyttäjiltä positiivisen vastaanoton. Ominaisuuden julkaisun jälkeen HeatSpring lähetti sähköpostia tiimien vetäjille ja pyysi palautetta uudistuksesta.

Sakeus ei osallistunut palautteen keräämiseen loppukäyttäjiltä. Kahden viikon kuluessa julkaisusta saatiin kolme palautetta, jotka kaikki olivat myönteisiä uutta Team Dashboardia kohtaan. Vaikka otos on pieni, on merkittävää, ettei uudistus saanut yhtään kielteistä palautetta. HeatSpringissa on yli 500 tiimiä, joten kielteisten palautteiden puuttumista pidettiin vahvana osoituksena uudistuksen onnistumisesta.



Kuva 11. Loppukäyttäjille julkaistu Team Dashboard.

Tiimin viikkopalaverissa ominaisuuden kehitystä käsiteltiin lyhyellä retrospektiivillä, eli tiimi keskusteli siitä, mikä Team Dashboardin uudistuksessa oli onnistunutta ja mitä kehitysprosessissa voitaisiin kehittää. Onnistuneena mallina nähtiin suunnittelijoiden työn porrastaminen, jossa alkuun pyritään prototyyppejä ja MVP:n kehittämällä määrittelemään lopputuloksen lopullinen muoto ilman, että kehitys hidastuu visuaalisten yksityiskohtien hiomiseen, mitä voitiin tehdä kehityksen myöhemmässä vaiheessa. Ominaisuuden kehitys päätöksestä aloittaa uudistuksen suunnittelu loppukäyttäjille julkaisuun kesti noin kuusi viikkoa ja toteutuksen kustannukset pysyivät Sakeuden ja HeatSpringin välisen noin 10–15 henkilötyöpäivän kuukausiallokaation

puitteissa. Kuukausiallokaatioon sisältyy myös muuta pienkehitystä että ylläpitotoimenpiteitä, joten Team Dashboardin laajuuden hallinnassa onnistuttiin tiimin näkemyksen mukaan hyvin.

Kehityskohteena tiimille nousi käyttäjätarpeiden selkeämpi tarkastelu ja kartoittaminen. Team Dashboardin kohdalla tekeminen kohdistui vahvasti ideoiden keräämiseen ja niiden muuttamiseksi toiminnallisuuksiksi. Tätä osaa prosessista päätettiin parantaa keräämällä enemmän ymmärrystä käyttäjä- ja asiakastarpeista seuraavan toimeksiannon kohdalla keräämällä enemmän tietoa HeatSpringilta toimeksiannon alkuvaiheessa.

3.3 HeatSpring – Kouluttajien työkalut

Kouluttajien työkalujen käyttöliittymän ja uusien ominaisuuksien muotoilu aloitettiin kehitystiimin toimesta heti Team Dashboard -ominaisuuksien julkaisun jälkeisessä kehityssprintissä. Kouluttajien työkalut toimivat käyttöliittymänä HeatSpringissa myytävän koulutussisällön tuottamiseen. Ne ovat siis liiketoiminnan kannalta kriittisessä roolissa, jotta yrityksellä on uusia ja ajantasaisia myyntiartikkeleja.

HeatSpringissa myytävä kurssisisältö on pääasiassa yrityksen ulkoisten toimijoiden toteuttamaa. Kurssien kouluttajat syöttävät kurssien sisällöt HeatSpringin alustalle ja he saavat osuuden kurssien myyntituotoista. Kurssien sisällöt koostuvat pääsääntöisesti videoista, mutta ne sisältävät myös tekstiä, linkkejä, PDF-muotoisia liitetiedostoja, kyselyitä ja tehtäviä, joissa opiskelijat lataavat itse tiedostoja kouluttajan arvioitavaksi. Itse kurssisisältöjen lisäksi merkittävä rooli kurssien myynnille on niiden myyntisivu, jonka kautta kurssin ostot tapahtuu.

3.3.1 Suunnittelutyöpaja

Toisin kuin Team Dashboardin kehityksen aloituksessa, kouluttajien työkalujen uudelleensuunnittelua ei aloitettu välittömästi viikottaisessa kehityspalaverissa. Projekti päätettiin aloittaa pitämällä erillinen, pelkästään kouluttajien työkalujen suunnittelulle omistettu työpaja. Työpajan pidettiin Google Meet -videopuhelun välityksellä ja sen kestoksi sovittiin kaksi tuntia. Työpajan asialista oli seuraava:

Tiedon haltija	Sisältö	Muotoiluajattelun osa-alue
HeatSpring	What are we trying to accomplish?	Kannattavuus
HeatSpring	Review of how we want instructor tools to work and stories about personas	Haluttavuus
Sakeus	Guidance about what decisions we need to make	Toteutettavuus
Sakeus & HeatSpring	Agreement about next steps, who is responsible for what, and the scope of the upcoming sprint	

Työpajan aikana Sakeus hallinnoi kaikille avoimena olevaa FigJam-työtilaa, jonne kerättiin muistiinpanoja. Ensimmäisen osion aikana kirjattiin seuraavia tavoitteita:

Työpajassa kirjattu tavoite	Selitys
Build a system for continuous creation	Mahdollistetaan sisältöjen jatkuva kehittäminen kouluttajille.
Discovery leads to Success, Success leads to Profit	Kehitetään työkaluja siten, että kurssien löydettävyyks paranee. Oletuksena on, että jos ostajalle relevantin kurssin löydettävyyks paranee, tämä lisää myyntiä.
Keep it lightweight – follow good conventions	Ei pyritä uudistamaan koko järjestelmää kerralla, vaan priorisoidaan olemassaolevan parantamista ja päivittämistä noudattamaan hyvän käytettävyyden käytänteitä niiltä osin kuin se on järkevää. Parannetaan jatkossa jatkuvalla iteroinnilla.
Easy to organise your content as an instructor	Sisällön kokonaisvaltaisen hallinnan helpottaminen.
Get instructors to build their first assignment – Get them started after the initial excitement.	Mahdollistetaan, että kurssisisällön syöttäminen saadaan käyntiin helposti kun kouluttaja saa innostuksen toteuttaa kurssin. Ylläpidetään kouluttajan motivaatiota.

Remove friction	Poistetaan esteitä ja hidasteita prosessista.
Celebrate the progress of getting things done	Tehdään kurssin kokoamisen eteneminen näkyväksi.

Tavoitteissa voidaan nähdä, että ne ovat varsin korkealla tasolla ja niille on haastavaa asettaa onnistumisen mittareita. Useat tavoitteet keskittyvät kouluttajien motivointiin kurssien toteuttamisessa sekä työkalujen käytön helppouteen. HeatSpringin henkilöstö jakoi tähän liittyen kokemuksiaan kouluttajien kanssa työskentelystä. Tyypillisesti kouluttajat valmistelevat kurssien sisällöt HeatSpringin ulkopuolella, työstäen niitä omilla tietokoneillaan esimerkiksi Word-ohjelmalla tekstitiedostoihin. Tämän jälkeen he toimittavat sisällöt HeatSpringille, missä henkilökunta rakentaa kurssin HeatSpringissa olevilla työkaluilla, jotka ovat sekä ylläpitäjille että kouluttajille saatavilla. Joskus tämä työ sisältää myös kurssisisältönä toimivien videoiden pilkkomista lyhyemmiksi kun yksittäisen videon pituus on ollut jopa yli tunnin mittainen. Työkalujen uudistamisen onnistuminen näissä tavoitteissa vähentäisi tätä HeatSpringin sisäisten resurssien käyttöä, mikä mahdollistaisi työajan käyttöä muihin, mahdollisesti tärkeämpiin työtehtäviin.

Tämän myötä pyrimme ajattelemaan kouluttajien työkaluja uudella tavalla. Lähtötilanteessa ne olivat ylläpidon päivittäisessä käytössä olevia työkaluja. Projektin tavoitteena tiimillämme on kehittää niistä helpommin lähestyttäviä käyttäjille, jotka käyttävät niitä ylläpitäjiä harvemmin, mutta joille ne ovat silti pääasialliset työkalut. Kun työkaluja käytetään harvoin, ei niiden hallinnassa voida luottaa pelkästään käyttäjän muistiin, vaan työkalu tulee järjestää siten, että niiden käyttö on intuitiivista ja loogista. Tämä selkeyden ja tavoitteellisuuden periaate toimisi projektin suunnitteluajurina.

Työpajan toisessa osiossa HeatSpringin eniten kouluttajien parissa työskentelevät asiantuntijat jakoivat ehdotuksia ja ideoita siihen, mitkä ominaisuudet ja muutokset uudistettuihin kouluttajien työkaluihin heidän kokemustensa ja havaintojensa perusteella tuottaisivat käyttäjille arvoa

ja lisäisivät täten palvelun käytön haluttavuutta. Ehdotetut ideat olivat konkreettisia ja niissä useimmiten toistuvana teemana oli kouluttajan sisällönhallinnan rajoitteiden minimoiminen. Tähän asti kouluttajilla ei ollut mahdollisuutta päivittää tuottamiensa kurssien hintaa tai lisätä markkinointivideoa kurssin myyntisivulle, vaan heidän tuli lähettää nämä ylläpitäjille, jotta tiedot voitiin päivittää. Lisäksi kouluttajan ei ollut mahdollista esikatsella kurssia HeatSpringissa ennen kuin kurssi oli julkaistu kaikkien nähtävillä, mikä lisäsi kouluttajien epävarmuutta sisällönsyötössä.

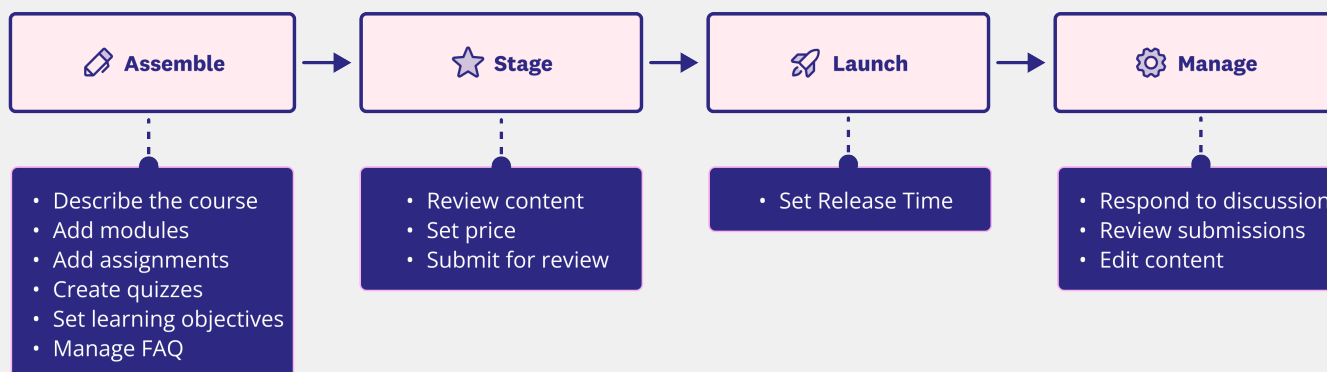
Kurssien sisällönsyötön motivoimiseksi ideoitiin mahdollisuus asettaa kurssit ennakkoon myytäväksi ja julkaistavaksi kouluttajan asettamana päivämääränä. Tämä nähtiin myös liiketoimintaa tukevana uudistuksena kun kurseja voitaisiin markkinoida ja myydä ennakkoon pienellä alennuksella, mahdollisesti kasvattaen kurssien myyntimääriä. Ennakkomyynti nähtiin myös keinona motivoida kouluttajia saamaan kurssinsa valmiiksi tiettyyn määräpäivään mennessä.

Sakeus nosti työpajassa uusien ideoiden toteutettavuuden kannalta merkittävän asian. Olemassa oleva tekninen toteutus kouluttajien työkaluille oli rakennettu siten, että samat tekniset toteutukset olivat sekä kouluttajien että kurssien kävijöiden käytössä samanaikaisesti. Käyttäjäroolista riippuen sovelluskoodiin oli asetettu tarkistuksia, jolloin käyttäjä näki omaa rooliaan koskevaa eriävää sisältöä. Tämä kuitenkin tarkoitti sitä, että ominaisuuksien toimintaa määrittelevä koodi olisi vaarassa paisua jatkokehityksen myötä ja tämä kasvattaisi virheiden muodostumista ja vaikeuttaisi niiden löytämistä. Jotta ohjelmistokoodi pysyisi hallittavana, päätettiin työpajassa, että kurssien kävijöitä koskeva ohjelmistokoodi erotettaisiin kouluttajien työkaluista. Tämä myös mahdollisti sen, että kouluttajien ja kurssien kävijöiden käyttöliittymät voisivat jatkossa poiketa toisistaan selvemmin.

3.3.2 Suunnitteluvaihe

Työpajan jälkeen tehtävänä oli purkaa työpajassa tehtyjä havaintoja, koota ideat ja pyrkiä muodostamaan niistä käyttäjiä palveleva kokonaisuus. Tätä

varten suunnittelija Virtanen pyrki kuvaamaan kouluttajan työvaiheita, joiden tehtävät pohjautuivat sekä sen hetkisiin kouluttajan ominaisuuksiin sekä työpajassa HeatSpringin keräämän asiakaspalautteen ja havaintojen tuottamista ideoista. Vaiheistus on kuvattu kuvassa 12.

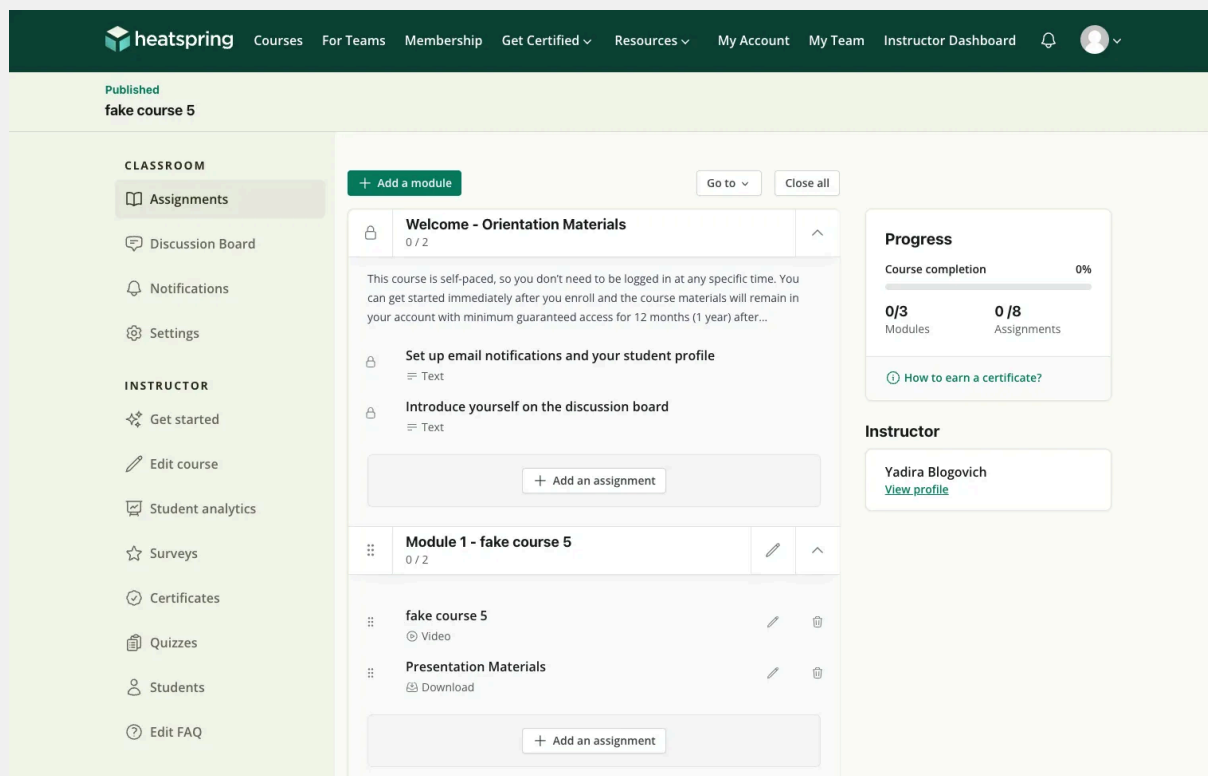


Kuva 12. Kouluttajan työvaiheet tehtävineen kuvattuna suunnittelun alkuvaiheessa.

Vaiheistuksen luomisessa hyödynnettiin *Jobs-To-Be-Done* -viitekehyksestä inspiraatiota ammentanutta lähestymistapaa. *Jobs-To-Be-Done* on käyttäjälähtöisen suunnittelun menetelmä, joka keskittyy asiakkaan tarpeisiin ja tavoitteisiin tuotteen tai palvelun käytössä pelkkien ominaisuuksien sijaan. Tämä lähestymistapa auttaa ymmärtämään syvällisemmin, mitä asiakkaat todella haluavat saavuttaa ja miten tuote tai palvelu voi auttaa heitä siinä. (Laubheimer 2017.) Kouluttajan työvaiheita kuvaavassa kaaviossa pyrittiin tunnistamaan ne kriittiset tehtävät, jotka kouluttajan on suoritettava kurssin luomisessa ja hallinnassa. Näin pystyttiin myös konkretisoimaan toteutettavien ja muutosta vaativien ominaisuuksien vaatimuksia ohjelmistokehityksen viitekehykseen sopivampaan muotoon tehtävänhallintaa kirjattavia tehtäviä varten.

Kouluttajien työkalujen käyttöliittymää tarkastelemalla voitiin havaita useita kohteita, jotka olivat kouluttajien työskentelylle epäolennaisia ja kohteita, jotka eivät tukeneet hahmoteltuja työvaiheita ja -tehtäviä. Tällaisia olivat muun muassa kouluttajalle näkyvät kurssisuorituksen etenemisen indikaattorit. Nämä olivat seurausta ominaisuuksien jakamisesta kurssin suorittajien kanssa. Kuvassa 13 on esitetty kuvakaappaus toimeksiannon alkuvaiheessa käytössä olleesta kouluttajien

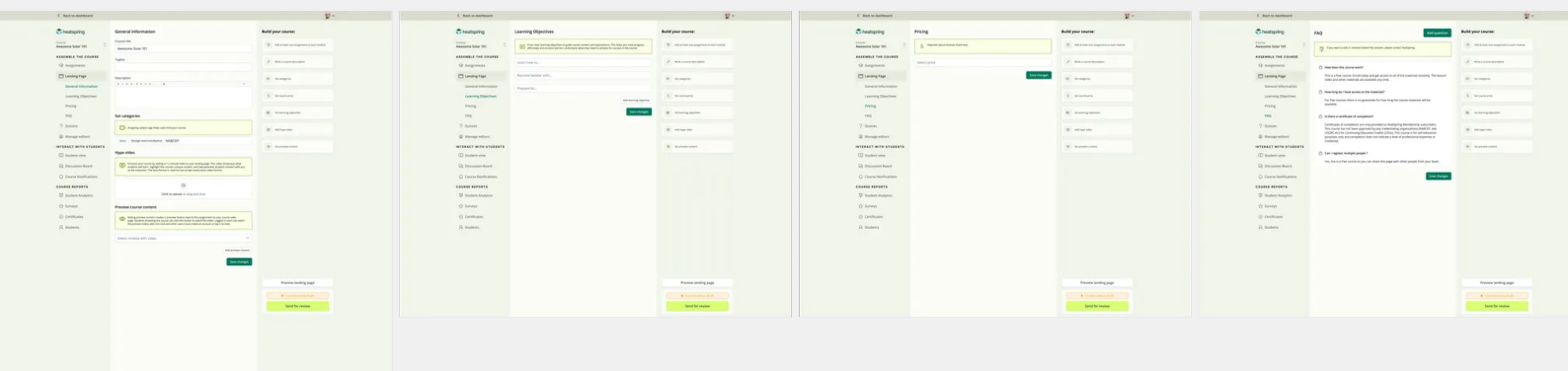
työkalujen etusivusta. Näkymän vasemmassa laidassa on kouluttajan työkalujen valikko. Valikon "Classroom"-otsikon alla olevat näkymät ovat sellaisia, joiden sisältö on sama myös kurssin suorittajille. "Instructor"-otsikon alla taasen on nähtävillä sekalainen lista kaikista muista työkaluista, joita kouluttajille on käytössä. Asetimme tavoitteeksi uudistaa sekä näkymiä että valikkoa siten, että ne ohjaavat kouluttajaa tekemään oikeita toimenpiteitä kurssin rakentamisen eri vaiheissa.



Kuva 13. HeatSpring-kouluttajien työkalujen oletusnäkökulma ennen uudelleensuunnittelua.

Ideoita alettiin visualisoida Team Dashboard -toimeksiannon tapaan Figmalla toteutettavilla käyttöliittymäkuvilla, jotka linkattiin toisiinsa klikkailtavaksi prototyypiksi HeatSpringin henkilöstön testattavaksi. Myös palautteen keräämisessä hyödynnettiin edellisen toimeksiannon tapaan Figman sisäisiä kommentointityökaluja palautteen dokumentoimiseksi. Kun Team Dashboardin kohdalla toimeksiannossa huomio keskittyi suurimmaksi osaksi yhteen laajaan ominaisuuteen, oli kouluttajien työkaluissa useista pienemmistä toiminnoista kytkeytyvä joukko näkymiä, jotka tuli hahmottaa. Tämä vaati siis hieman erilaista lähestymistapaa. Prosessia pyrittiin pilkkomaan vielä pienempiin vaiheisiin ja nämä vaiheet

määrittäisivät näkymät, joiden kautta kurssin tietoja syötettäisiin.



Kuva 14. Kouluttajien työkalujen varhaiset käyttöliittymäkuvaukset.

Kuvassa 14 on kuvattu varhaisia suunnitelmia kurssin myyntisivun tietojen syötön jakautumisesta usean näkymän välille. Myyntisivun tiedot jaettiin useampaan näkymään, jotta täytettävät lomakkeet olisivat vähemmän kuormittavia. Oletuksenamme oli myös, ettei kouluttajilla olisi kurssin rakentamisvaiheen alussa kaikkea tarvittavaa tietoa valmiina, joten jaottelulla ohjattaisiin tietojen syöttöä kurssin rakentamisen tietyissä vaiheissa. Kuviossa on nähtävissä myös kouluttajien työkalujen navigaation muutokset, jotka heijastavat työpajan pohjalta tehtyä vaihteistusta. Näkymien oikeassa laidassa nähtävällä työlista-ominaisuudella tavoiteltiin parempaa näkyvyyttä kouluttajalle siihen, mitä tietoja kurssin toteuttamiseksi vielä edellytetään. HeatSpringilta kerätyn palautteen pohjalta tämä ominaisuus kuitenkin hylättiin, koska asiakkaalla oli kokemusta työlistan kaltaisen ominaisuuden testaamisesta aiemmin ja tämä ei ollut osoittautunut tehokkaaksi. Tämä osoitti prototyypin hyödyntämisen hyötyjä kun nopean palautteen pohjalta kykenimme säästämään kehitysresursseista ja priorisoimaan muita vaihtoehtoja. Päätimme rakennusvaiheeseen jättää näkymien oikean palstan tyhjäksi edetäksemme kehityksessä ja voidaksemme testata pääasiallisia toiminnallisia uudistuksia käytännössä.

3.3.3 Rakentaminen

Rakentamisvaihe aloitettiin samoin menetelmin kuin Team Dashboard noin kaksi viikkoa suunnitteluvaiheen aloituksesta. Toteutustyötä helpotti

muun muassa se, että toisin kuin edeltävässä toimeksiannossa, kouluttajien työkalujen näkymien yleinen asettelu ei edellyttänyt laajoja muutoksia, vaan tiimi pystyi keskittymään sisältömuutoksiin ja hyödyntämään jo toteutettua asettelua tarvittaviin uusiin näkymiin. Näin tiimillä oli mahdollisuus keskittyä koestamaan miten näkymien sisältö toimii käytännössä ja uudistaa sisältörakennetta palautteen mukaan. Samalla kun näkymien rakennetta ohjelmoitiin, Henri Nikka jalosti Team Dashboardin kehitysvaiheen tapaan kehittäjien työkalujen visuaalista ilmettä Figmassa.

Toteutusvaiheen aikana toiminnallisuuksia julkaistiin HeatSpringin testattavaksi verkosta saatavilla olevaan testiympäristöön. Tämä osoittautui jälleen arvokkaaksi välivaiheeksi, koska HeatSpring antoi palautetta useista uusien toiminnallisuuksien vaiheista, joihin pystyttiin reagoimaan nopeasti ja parantamaan käytettävyyttä hyödyntäen heidän tietämystään asiakkaista. Sen lisäksi että he kommentoivat ominaisuuksia kouluttajien näkökulmasta, kouluttajien työkalujen uudistukset vaativat uudistuksia myös HeatSpringin itse käyttämiin ylläpitäjien työkaluihin, mistä saatiin myös työskentelyä ohjaavaa palautetta.

3.3.4 Tulokset ja tavoitteissa onnistuminen

Kouluttajien työkalujen julkaisun jälkeen HeatSpringin työntekijät keräsivät palautetta kouluttajilta. Heti julkaisun jälkeen kerätyn palautteen perusteella kouluttajat kokivat uudistetut työkalut selkeiksi käyttää ja niiden käyttöönotto sujui helposti. Negatiivista palautetta työkalujen uudistuksesta ei kertynyt, minkä HeatSpring näkee myös onnistumisena.

Uusia kursseja toteutetaan nykyään noiden viisi kuukaudessa kun aiemmin alustalla oli uusia kursseja kaksi kuukaudessa, eli kurssien julkaisutahti kasvoi uudistuksen myötä 250%. Tähän on vaikuttanut kuitenkin käyttöliittymä uudistuksen lisäksi HeatSpringin oman panostuksen lisääminen kurssisisältöjen tuottamiseen yhteistyössä kouluttajien kanssa. Tämä panostus on osittain yhteydessä uudistuksessa kehitettyihin ennakkomyyntiominaisuuksiin, jotka mahdollistavat uusia

markkinointikeinoja uusille kursseille. Vaikka tulokset ovat positiivisia, HeatSpringin työntekijöiden roolin vähentämisessä kurssisisältöjen tuottamisessa ei täysin onnistuttu. Uudet työkalut kuitenkin nopeuttavat ja helpottavat kurssisisältöjen syöttämistä myös henkilöstölle ja avaavat mahdollisuuksia laajentaa kouluttajien työkalupakkia jatkossa tehokkaammin.

Toimeksiannon prosesseja kehitettiin tiimin mielestä oikeaan suuntaan edellisestä toimeksiannosta nousseiden oppien mukaisesti. Kouluttajien työkalujen tarpeiden kartoittamiseen työpajamenetelmin nähtiin toimivana tapana asettaa selkeitä yhteisiä tavoitteita ja vastaavia menetelmiä pyritään hyödyntämään tarpeen mukaan myös tulevaisuudessa. Toimeksiannossa myös vahvistui suunnittelijoiden työn porrastamisen toimivuus ketterän kehittämisen sekä nopean validoinnin ja oppimisen mahdollistajana. Myös toimeksiannon laajuuden rajaaminen onnistui ja kokonaisuus toteutettiin Team Dashboardin tapaan kuudessa viikossa ja tiimin kuukausiallokaation puitteissa.

4 Sakeuden muotoiluprosessit ja toimeksiannoissa tehdyt havainnot

HeatSpringille toteutetut kaksi toimeksiantoa mahdollistivat laajuudeltaan samankaltaisina sen, että Sakeuden työskentelyprosessia voitiin vertailla ja kehittää niiden välillä. Vertailussa havaittiin prosesseissa yhtäläisyyksiä useisiin menetelmiin, jotka olivat Sakeuden tiimin jäsenille entuudestaan tuttuja. Näitä menetelmiä ei kuitenkaan ollut aiemmin järjestelmällisesti sisällytetty osaksi kehitettävien palveluiden muotoiluprosessia, vaan ne olivat työn kautta opittuja hyväksi havaittuja menetelmiä. Tässä osiossa kokoan yhteen toimeksiantojen kautta, miten Sakeus toteuttaa muotoiluajattelun ja palvelumuotoilun periaatteita sekä miten prosessissa havaitut menetelmät viedään yrityksen toiminnassa käytäntöön.

4.1 Asiakastarpeiden kartoittaminen

Toimeksiannot aloitetaan aina asiakastarpeiden kartoittamisella. Kartoituksen toteuttamiseksi järjestetään asiakkaan kanssa Sakeuden fasilitoima työpaja, missä tiimi selvittää asiakkaan liiketoiminnan ja käyttäjien tarpeita. Tavoitteena on paitsi kerätä tietoa, myös luoda yhteinen ymmärrys projektin tavoitteista ja haasteista asiakkaan kanssa ja pyrkiä taklaamaan näitä haasteita jo ennalta. Työpajan kesto ja siinä käytettävät menetelmät kartoitustyöhön riippuvat monista ennakkotietona kerättävistä tekijöistä, kuten toimeksiannon budjetin suuruudesta ja onko asiakkaan liiketoiminta-ala tiimille ennestään tuttu. Ennalta määritettyjen menetelmien sijaan voidaan työpajan aikana pivotoida hyödyntämään sellaisia menetelmiä, joita ei alunperin oltu suunniteltu työpajassa hyödynnettävän. Liiallinen ennakkovalmistelu voi osoittautua jopa haitalliseksi ja työpajan lähteminen sivuraiteille voi tuottaa uusia oivalluksia kehitettävien ratkaisujen osalta (Stickdorn ym. 2019, 398).

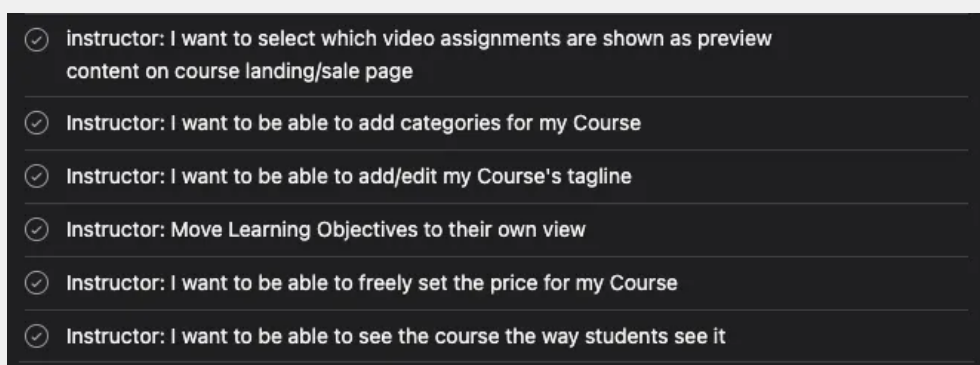
Tässä työssä hyödynnettiin työpajatyöskentelyä kouluttajien työkalujen toimeksiannossa jakamalla asialista muotoiluajattelun osa-alueita vastaaviin teemoihin. Koska Sakeus tunsi asiakkaan liiketoiminnan ennestään, menetelmänä käytettiin pääasiassa keskustelevaa aivoriihityöskentelyä näiden teemojen mukaisesti. Työpajoissa voidaan kuitenkin lisätä asiakkaan osallistamista hyödyntämällä liiketoimintaa ja käyttäjäpolkuja kartoittavia menetelmiä. Tällaisia hyväksi havaittuja työkaluja ovat esimerkiksi *User Journey Mapping*, *Business Model Canvas* ja *Value Proposition Canvas*.

4.2 Prototyypaus

Asiakastarpeiden perusteella pyrimme kehittämään prototyypin. Prototyypaaminen on Sakeudelle keino ideoida ja koestaa mahdollisia ratkaisuja, tai kuten Wendt (2015, 49) ilmaisee asian, ideoiden tutkimista fyysisessä tilassa. Prototyypaus on jatkuva osa Sakeuden palvelunkehitysprosessia ja yksittäisen toimeksiannon aikana voidaan toteuttaa useita prototyyppejä erilaisissa muodoissa. Prototyyppi

mahdollistaa lopputuotteen laajuuden määrittelyn kun toteutushaasteita voidaan tarkastella matalalla kustannuksella. Näin Sakeus voi yhdessä asiakkaan kanssa priorisoida toteutuksen pääpainopisteitä ja rajata toteutuslaajuutta asiakkaan budjetin ja liiketoiminnan tarpeiden mukaan.

Yksinkertaisimmillaan prototyyppi voi olla paperille piirrettyjä karkeita luonnoksia, joiden avulla määritetään ideoitavan ratkaisun keskeiset elementit. Luonnoksissa toteutus voidaan määritellä hyvin abstraktilla tasolla, esimerkiksi kuvaamalla elementtejä sanoilla piirrosten sijaan. Shape Up -mallissa ratkaisujen luonnostelussa hyödynnetään *breadboarding*-konseptia, jossa käyttäjäpolkua mallinetaan kuin sähkötekniikan koekytkenälevyille. *Breadboardingissa* määritetään kolme pääelementtiä: paikat (*places*), joihin käyttäjä navigoi; affordanssit (*affordances*), eli käyttäjän vaikutusmahdollisuudet, kuten painikkeet ja syöttökentät sekä ohjaava tekstisisältö; ja yhteydet (*connection lines*), jotka osoittavat, miten affordanssit ohjaavat käyttäjää paikasta toiseen. *Breadboarding* mahdollistaa käyttäjäpolun keskeisten vaatimusten määrittelyn ilman visuaalisia valintoja ennen kuin suunnitelmia konkretisoidaan tarkemmin. (Singer 2019, 36–41.)



Kuva 15. Kouluttajan työkaluille määritettyjä toiminnallisuuksia pilkottuna pienempiin

Sakeus ei toteuta *breadboardingia* tarkalleen samalla tavalla kuin Shape Up -mallissa, mutta prototyyppauksessa toiminnallisuuksien ja käyttäjäpolun karkea kuvaaminen tekstimuotoisina listoina ja vuokaavioina toimii usein osana prosessia, jotta abstraktion taso on varhaisessa vaiheessa korkealla mahdollistamaan tarvittaessa nopea suunnitelmien pivotointi ja kehitettävän ominaisuuden systeeminen tarkastelu. Listaamista

toteutetaan esimerkiksi tehtävähallintajärjestelmään kuvaamaan mitä tietoja käyttäjän tulee lomakkeella syöttää, kuten nähdään kuvassa 15.

Sakeudelle tyypillisin tapa kehittää ja toimittaa prototyyppi on piirtää käyttöliittymäkuvat Figmalla. Tämä mahdollistaa ideoiden toteutettavuudesta keskustelun asiakkaan kanssa sekä liiketoiminnallisen hyödyn että teknisen toteutettavuuden näkökulmasta konkreettisella tasolla. Prototyypauksessa hyödynnetään asiakkaan brändi-ilmettä, mutta prototyypillä ei pyritä kuvaamaan lopullista tuotetta – lopputuotteen viimeistely jätetään koodausvaiheeseen. Tästä viestitään myös asiakkaalle selkeästi.

Sakeuden tiimi pyrkii tällä lähestymistavalla siihen, että palautteen ja validaation kohteena olisivat kehitettävien asioiden rakenne ja toiminnallisuuden hyödyllisyys, eikä aikaa käytettäisi visuaalisiin yksityiskohtiin. Ennen kuin näkymien keskeisten elementtien hyödyt on todennettu, yksityiskohdat voivat muuttua merkittävästi. Näihin keskittyminen varhaisessa vaiheessa olisi vastoin Lean-mallien mukaista hukkatyön välttämistä. Lisäksi, koska Sakeuden suunnittelijat toimivat myös käyttöliittymän toteuttajina, yksityiskohtainen visuaalinen mallintaminen ei ole tarpeen käyttöliittymäkehittäjää varten, vaan suunnittelijoiden kantama hiljainen tieto siirtyy myös suoraan kehitysvaiheeseen.

Myös varhaiset kehitysvaiheen tuotokset voidaan nähdä prototyyppeinä, joilla toiminnallisuuden toteutettavuus ja hyödyllisyys voidaan Lean Startupin Rakenna-mittaa-opi -mallin mukaisesti validoida. Sakeuden työssä Rakenna-mittaa-opi ja Lean UX -syklit toteutuvat useina iteraatioina prototyypeistä kohti MVP:ta kun palautetta asiakkaalta kerätään jokaisen syklin jälkeen ja toiminnallisuuden kehitetään tämän palautteen perusteella eteenpäin lisäten uusia ideoita ja tarvittaessa korjatessa kehityksen suuntaa.

4.3 Työvaiheiden porrastaminen

Työn porrastamisen tavoitteena on, että asiakkaalle toimitetaan jatkuvaa

arvoa ketterästi. Työskentely jakautuu suunnittelijoiden välillä löyhästi kahdelle polulle: toiminnallisuuksien suunnitteluun ja kehitykseen. Kuten Shape Up -mallissa (Singer 2019, 26), kahden polun mallilla pyritään siihen, että kehitystyö voidaan aloittaa tehokkaasti, kun toteutettavuuteen liittyvät keskeiset haasteet on selvitetty jo suunnitteluvaiheessa esimerkiksi prototyypeillä validoiden tai asiakkaan kanssa keskustelemalla.

Sakeuden mallissa Simo Virtanen määrittelee käyttäjäpolut, prototyypaa näkymien rakennetta ja toteuttaa yhdessä Virgil Mocanun kanssa toimivan ohjelmiston ensimmäisen version. Toimivan ohjelmiston toteutusvaiheessa Henri Nikka iteroi suunnitelmien käyttöliittymän käytettävyyttä sekä visuaalisia yksityiskohtia muuttamatta aiemmassa vaiheessa määritettyjä ydintoiminnallisuuksia. Suunnittelijat siis vuorottelevat suunnittelu- ja kehityspolkujen välillä, mutta suunnittelun painopiste eroaa: Virtasella se on rakenteessa ja toiminnallisuudessa, Nikalla käytettävyydessä ja käyttäjäinteraktiossa.

Ketterän ohjelmistokehityksen periaatteiden ja Lean UX:n mukaisesti tiimi jakaa koko prosessinsa aikana tietoa keskenään ja voi myös työskennellä yhdessä eri vaiheissa, eli roolijako ei ole tiukasti määritelty. Tiimi tarkastelee ja mukauttaa tarvittaessa toimintaansa ja tiimiläisten vastuualueita maksimoidakseen asiakkailleen toimitettavan hyödyn. Tässä työssä kuvatuissa toimeksiannoissa kuitenkin noudatettiin pitkälti vastuunjakoa edellä kuvatulla tavalla.

4.4 Toiminnallisuuksien testaus ja julkaisu

Työ ei ole valmis ennen kuin se on julkaistu ja käyttäjien käytettävissä. Kun toiminnallisuuksia kehitetään, kehittää tiimi ne Shape Up -mallin kaltaisesti siivu kerrallaan. Singer (2019, 103–105) jakaa ohjelmistokehityksen karkeasti kahteen kerrokseen, *front-endiin*, eli käyttöliittymäkehitykseen, ja *back-endiin*, eli taustajärjestelmäkehitykseen. Kehityksen jakaminen siivuihin mahdollistaa, että asiakkaalle saadaan toimivia ohjelmiston osia nopeasti testattavaksi.

Toiminnallisuuksien siivuttaminen edellyttää suunnittelijan ja kehittäjän tiivistä yhteistoimintaa. Vaikka suunnittelija toteuttaisi koodiin useita näkymiä, ilman taustajärjestelmää nämä eivät ole käytettäviä ohjelmiston osia. Samalla tavalla, mikäli taustajärjestelmä kehitetään eri osille toimeksiantokokonaisuutta kuin käyttöliittymä, ei tällöinkään synny toimivaa ohjelmistoa, vaikka ohjelmistokoodia syntyykin. Toimivaa ja testattavaa ohjelmistoa syntyy vain jos sekä käyttöliittymä että taustajärjestelmä on kehitetty samalle siivulle, kuten kuvassa 16 on kuvattu. (Singer 2019, 103–105.)



Kuva 16. Kehitettävien ominaisuuksien pilkkominen (Mukaelma: Singer 2019, 104–105).

Kun ohjelmiston osia saadaan toteutettua, ne julkaistaan asiakkaan testattavaksi testipalvelimelle. Näin asiakas pystyy testaamaan ominaisuuksia, antamaan Sakeudelle palautetta ominaisuuksien toimivuudesta niin liiketoiminnan kuin omien asiakkaidensa näkökulmasta sekä vahvistamaan tai tyrmäämään Sakeuden tiimin tekemiä oletuksia suunnittelussaan. Testaamisen pohjalta tiimi toteuttaa parannuksia ohjelmistoon ja asiakkaan hyväksynnän saatuaan julkaisee palvelun tai ominaisuudet loppukäyttäjien käyttöön.

4.5 Työn jakautuminen ja toteutusosaaminen muotoiluprosessissa

Koska käyttäjälähtöisessä suunnittelussa ja käyttökokemussuunnittelussa vaaditaan laajaa osaamista, koetaan tiimissä, että suunnittelussa työtehtävien jakaminen mahdollistaa tehokkaamman työskentelyn ja vähentää suunnittelijan kognitiivista taakkaa. Sakeuden kohdalla työskentely jakautuu siten, että Simo Virtasen rooli on vastata

ensisijaisesta asiakas- ja käyttäjätarpeiden kartoituksesta ja informaatioarkkitehtuurisuunnittelusta palvelumuotoilun keinoin. Henri Nikka hyödyntää Virtasen löydöksiä ja vastaa käyttöliittymäsuunnittelusta, jotka sisältävät interaktiosuunnittelun ja ulkoasusuunnittelun. Jotta palvelut voidaan toteuttaa toimivaksi ohjelmistoksi, Virgil Mocanu vastaa palveluiden taustajärjestelmän ohjelmistokehityksestä ja ohjelmistoarkkitehtuurista. Käytettävyys ja koko käyttäjäkokemus taasen ovat koko tiimin vastuulla. Näin tiimi mahdollistaa jokaisessa projektissaan kokonaisvaltaisen käyttäjälähtöisen kehityksen. Näitä Sakeuden tarjoaman osa-alueita on tarkoitus kuvata yrityksen verkkosivuilla tulevaisuudessa kuvan 17 mukaisesti.



Palvelumuotoilu

Etsimme kanssasi uusia mahdollisuuksia ja testaamme prototyypin keinoja toiminnan tehostamiseen tai uuteen liiketoimintaan.



UX/UI-suunnittelu

Digitaaliset palvelut on tehty ihmisten käytettäväksi. Suunnittelemme ja kehitämme palvelut käyttäjien tarpeet huomioiden.



Ohjelmistokehitys

Etsitkö kumppania rakentamaan palvelusi? Toteutamme web-selaimessa toimivia palveluita viimeisimmillä verkkoteknologiolla.

Kuva 17. Tiivistelmä Sakeuden palveluista.

Vaikka Sakeudella on roolitettu tekijät ohjelmistokehittäjiin ja suunnittelijoihin, nähdään tiimissä, että työtehtävien tiukka roolittaminen on ongelmallista kun pyritään tuottamaan arvoa ja tuloksia tehokkaasti. Aiheesta on kirjoittanut myös muun muassa Madsen (2023) seuraavasti:

”Kriittisin virhe suunnittelijoiden ja ohjelmistokehittäjien välisessä yhteistyössä tapahtuu, kun sekoitamme työkalujen jakautumisen tehtävien välillä tarpeeseen selvästä työnjaosta. Suunnittelun ja teknisen toteutuksen käsitteleminen kahtena täysin erillisenä prosessina johtaa

eristettyyn vesiputousmalliseen työnkulkuun, jossa suunnittelijat ensin piirtävät haavekuvia ideoista staattisissa työkaluissa, ja ohjelmistokehittäjät sitten toteuttavat halutut ominaisuudet, kun ne ovat valmiita kehitettäväiksi. Suunnitelmien hand-over, tämä oletetusti taianomainen hetki, jolloin suunnittelu on valmis ja tekninen toteutus alkaa, on monien tuotetiimien yhteistyöongelmien perimmäinen syy.” (Madsen 2023.)

Madsen (2023) ehdottaa ratkaisuksi suunnittelu- ja ohjelmistokehitysprosessien välisen näkyvyyden lisäämistä. Hän suosittelee myös suunnittelutyön toteuttamista koodissa aiemmin palvelun kehityksessä kuin organisaatioissa tyypillisesti tehdään. Sakeudella nämä ratkaisut toteutuvat suunnittelijoiden työelämässään saavuttaman ohjelmistokehitysoosaamisen kautta. Suunnittelijoilla on itsellä kyvykyys toteuttaa ja iteroida Figmalla toteuttamia suunnitelmia ohjelmistokoodissa tai hyödyntää pelkkää ohjelmointia suunnittelutyökalunaan ja kehittää ratkaisut suoraan tuotteen koodiin.

Suunnittelun ja ohjelmistokehityksen välistä jakoa on kaventanut myös tekoälyavusteisten työkalujen, kuten *GitHub Copilot* ja *Cursor*-koodieditori, saapuminen markkinoille. Vaikka kaikilla tiimin jäsenillä ohjelmistokehitystaitoja, mahdollistavat nämä työkalut työskentelyä myös niillä ohjelmoinnin osa-alueilla, joilla heillä ei ole omakohtaista osaamista. Näillä työkaluilla suunnittelijat voivat esimerkiksi kehittää MVP:n, jotta toiminnallisuutta voidaan testata ilman kehittäjän työpanosta. Kun toiminnallisuus on validoitu, voidaan palautteen pohjalta kehittää toiminnallisuus joko kehittäjän toimesta parempaa ohjelmistokoodia tuottaen tai jatkaen jo olemassaolevasta tekoälyavusteisesti tuotetusta koodista.

Tekoälyavusteiset työkalut nopeuttavat työskentelyä myös muun muassa avustamalla näkymien tyylittelyssä mikäli samat Tailwindilla toteutetut CSS-tyylit toistuvat eri näkymien välillä. Tällöin tekoäly kykenee oppimaan aiemmin syötetyistä tyylittelyistä ja täyttämään ne uuteen näkymään.

Tämä vähentää kitkaa paitsi suunnittelijan työskentelystä, mutta myös mahdollistaa kehittäjille nopean näkymien tyylyttelyn ja arvon toimittamisen asiakkaalle ilman, että suunnittelijan saatavuus olisi hidastava tekijä.

Ohjelmistokehitysosaamisella on suunnittelijalle myös valtava etu kun yhtä muotoiluajattelun peruspilaria, toteutettavuutta, kartoitetaan. Kun suunnittelijalla on itsellä kykyä hyödyntää ohjelmistokoodia työkalunaan, pienenee toteuttamisen kartoittamiseen tarvittava aika ja epävarmuus huomattavasti.

4.6 Havaitut haasteet muotoiluprosessin täytäntöönpanolle

Vaikka Sakeus saa asiakkailtaan testauspalautetta ja näkemyksiä loppukäyttäjien tarpeista, toimeksiannoissa on havaittavissa puute loppukäyttäjiltä systemaattisesti kerättävästä palautteesta. Tämä on yksi alihankintana toteutettavan ohjelmistokehityksen haasteista: miten kerätä dataa ja mitata toteutusten onnistumista, jotta palvelua voidaan muotoilla ja kehittää käyttäjä- ja liiketoimintalähtöisesti? Jotta Sakeus voisi kerätä itse dataa loppukäyttäjiltä, tulisi tälle olla asiakkaan mandaatti.

Tämän lisäksi Sakeudella tulee olla liiketoiminnallinen insentiivi toteuttaa käyttäjätutkimusta ja kehittää asiakkaan tuotetta eteenpäin, mikä toteutuisi laskuttamalla käyttäjätutkimukseen käytetyistä työtunneista. Asiakas ei kuitenkaan ole välttämättä halukas tähän tai haluaa toteuttaa käyttäjäpalautteen keräämisen itse. Erityisen haastavaa käyttäjätutkimukseen investoimista on perustella silloin, kun liiketoiminta kehittyy positiivisesti siitä huolimatta ja ohjelmistokehitys pysyy kustannustehokkaana. Tämä tekee kuitenkin Sakeuden toiminnan suorien vaikutusten mittaamisen haastavaksi. Ilman systemaattista palautteen keräämistä ja uusien palveluiden tai ominaisuuksien tehokkuuden ja vaikuttavuuden mittaamista kehityksessä ei synny muotoiluprosesseille ominaista syklisyyttä, kuten "Rakenna-mittaa-opei" tai Lean UX -malleissa.

Käyttäjätutkimuksen puute ei tuota haasteita pelkästään ratkaisujen

testaamisessa, vaan myös ongelman määrittelyvaiheessa. Kun käyttäjien toimintaa ei havainnoida suunnittelutiimin toimesta, ei tiimillä ole myöskään mahdollisuutta havaita ennalta käyttäjien tarpeita ja liiketoiminnallisia mahdollisuuksia. Mikäli Sakeudella olisi mandaatti ja mahdollisuus toteuttaa käyttäjätarpeiden selvittämistä ennalta, voisi myös Shape Up -mallin kaltainen jatkuva uusien ratkaisuiden muovaus sekä asiakkaan kanssa tehtävä panostaminen eri ominaisuuksiin mahdollistua. Nykyisellään muovaus kuitenkin kohdistuu aina asiakkaan toimeksiantona ja kohdistuu pääasiassa yhteen ominaisuuteen kerralla. Panostaminen myös tapahtuu toimeksiannon yhteydessä, eli päätös toteuttaa ominaisuudet tapahtuu jo ennen kuin suunnittelua aletaan tekemään.

Toimeksiantopäätöksiin liittyy viimeinen käsiteltävä haaste: liian pienet budjetit. Kun ominaisuuksien kehittämiseen on mahdollista tehdä vain pieniä rahallisia panostuksia, tämä johtaa muotoiluajattelulle tärkeän kokeellisen tekemisen vähenemiseen. Se myös heikentää mahdollisuuksia muuttaa kehityksen suuntaa tai keskeyttää huonojen ratkaisujen kehitys kokonaan – usein siksi, että kehitykseen sijoitetun rahan ei haluta näyttävän valuneen hukkaan. Brown (2019) kirjoittaa, että luovalla tiimillä tulee olla aikaa, tilaa ja budjetti tehdä virheitä (Brown 2019, 75). Kun asiakkaille toteutetaan räätälöityä ohjelmistoa, suunnittelijat ja kehittäjät ovat aina kokeellisen ja uuden kehittämisen äärellä. Budjetille voi asettaa rajoja, ja se voi osaltaan toimia myös luovien ratkaisujen katalyyttinä, mutta se ei saa olla niin pieni, että se estää ratkaisujen testaamisen, virheellisten oletusten korjaamisen ja iteratiivisen kehittämisen.

5 Yhteenveto

Tässä työssä tarkasteltiin Sakeuden toimeksiannoissa toteuttamia muotoilun ja muotoiluajattelun menetelmiä. Työ ei kuitenkaan välttämättä anna kaikenkattavaa kuvaa Sakeuden muotoilutyöskentelystä sen kohdistuessa yhteen asiakkuuteen, jolla on omat erityispiirteensä. Tällaisia ovat muun muassa se, että kehittäminen HeatSpringin kanssa on

jatkuvaa ja tiimi suunnittelee palvelun kehittämistä viikottain yhteisissä palavereissa kun useiden asiakkaiden kanssa voi olla pitkiä jaksoja, jolloin kehitystä ei tehdä. Lisäksi HeatSpring sijaitsee ja harjoittaa liiketoimintaansa toisella mantereella, mikä vaikuttaa työskentelyn aikatauluihin ja siihen, ettei Sakeudella ole mahdollisuutta toteuttaa paikan päällä tapahtuvaa työpaja- tai käyttäjätutkimustyöskentelyä. Työ kuitenkin kuvaa Sakeudelle ominaista tapaa kehittää työskentelytapojaan asiakkaan ja toimeksiannon tarpeiden mukaan, kuten toimeksiantojen välillä tapahtuvassa menetelmien kehittämisessä erityisesti asiakastarpeen kartoituksen osalta voidaan havaita.

Sakeus toteuttaa muotoilua käytäntölähtöisesti, pyrkien validoimaan asiakkaiden esittelemiä liiketoiminta- ja käyttäjätarpeita mahdollisimman pian prototyypin ja kehittämällä MVP-toteutuksia. Ideat realisoimalla voidaan tuoda ideointi konkreettisemmalle tasolle, yhtenäistää ajatusta lopputuloksesta asiakkaan kanssa, rajata lopputuotoksen vaatimuksia ja oppia käytännön kautta, mitkä asiat todella toimivat ja mitkä eivät. Sakeuden malli seuraa näiltä osin hyvin läheisesti Lean Startup ja Lean UX -malleja. Molemmissa malleissa kerättävä palautteen ja datan kerääminen ovat kuitenkin keskiössä, mutta nämä jäävät toimeksiannoissa pääsääntöisesti asiakkaan vastuulle. Tämän muuttaminen on haastavaa, koska käyttäjätutkimuksen mahdollistaminen Sakeudelle nostaisi palvelukehitysinvestointeja.

Mikäli Sakeudelle kuitenkin annettaisiin mandaatti ja resurssit toteuttaa käyttäjätutkimusta sekä toimeksiantojen alkuvaiheessa että tuotteen jatkuvan kehityksen aikana, voisi se parantaa asiakkaille tarjottavia suunnittelupalveluita kun Sakeus kykenisi käyttäjähaastattelujen ja -palautteen kautta kehitettyä omaa näkemystään asiakkaan liiketoiminnan kehittämiseksi. Tällöin Sakeudella tulisi olla myös mahdollisuus proaktiivisesti suunnitella ja ehdottaa asiakkailleen datan perusteella liiketoimintaa ja käyttäjiä hyödyttäviä kehityskohteita. Tällainen malli mahdollistaisi myös Shape Up -mallin kaltaisen tuotekehityksen, joka on kenties tyypillisempää *in-house*-tiimien toteuttamalle palveluiden kehittämislle kuin alihankintana toteutettavalle ohjelmistokehitykselle.

Sakeudelle tämä työ tekee näkyväksi omien toimintaprosessien toimivia osa-alueita sekä herättää keskustelua siitä, miten yritys voi asemoida omaa tarjootaan alansa liiketoimintakentässä. Pienenä yrittäjävetoisena yrityksenä Sakeudelle ei ole tärkeää pelkästään tarjota asiakkaidensa tarpeisiin sopivia muotoilu- ja ohjelmistokehityspalveluita, vaan myös kehittää omaa toimintaansa ja solmia vahvoja kumppanuuksia, joissa sillä on mahdollisuutta aidosti syventyä asiakkaansa liiketoimintaan ja toteuttaa tässäkin työssä kuvattuja muotoiluprosesseja laajemmin. Tässä työssä kuvatut toimeksiannot osoittavat Sakeudelle, että muotoiluprosessi ja palveluiden kehittämisen tehokkuus ovat kehittyneet varsin kypsälle tasolle, josta tason noston edellytyksenä on asiakkaiden toimintaympäristön ymmärryksen kehittäminen tiiviimmän asiakasyhteistyön kautta. Nykyisellä tehokkuudellaan Sakeudella on mahdollisuudet rakentaa asiakkaisiinsa luottamusta, jonka myötä uskon avaavan Sakeudelle mahdollisuuksia suurempaan rooliin asiakkaidensa liiketoiminnan kehittämisessä.

Lähteet

Agile Alliance. N.d. Agile 101. [verkkosivu]. Luettavissa osoitteessa: <https://www.agilealliance.org/agile101/> (viitattu 28.09.2024)

Beck, Kent & Beedle Mike & van Bennekum, Arie & Cockburn, Alistair & Cunningham, Ward & Fowler, Martin & Grenning, James & Highsmith, Jim & Hunt, Andrew & Jeffries, Ron & Kern, Jon & Marick, Brian & Martin, Robert C. & Mellor, Steve & Schwaber, Ken & Sutherland, Jeff & Thomas, Dave. 2001. Ketterän ohjelmistokehityksen julistus. [verkkosivu]. Luettavissa osoitteesta: <https://agilemanifesto.org/iso/fi/manifesto.html> (viitattu 25.08.2024)

Broadbent, Harrison. 2023. Your first Stimulus Controller – Learn Stimulus in Ruby on Rails by building a toggle. [blogi-teksti]. Luettavissa osoitteessa: <https://railsnotes.xyz/blog/your-first-stimulus-controller-learn-stimulus-ruby-on-rails-by-building-a-toggle-beginners-guide> (viitattu 22.09.2024)

Brown, Tim. 2019. Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation, Revised and Updated. E-kirja. Yhdysvallat: HarperCollins Publishers.

Brown, Tim. 2008. Definitions of Design Thinking. [verkkosivu]. Luettavissa osoitteessa: <https://designthinking.ideo.com/blog/definitions-of-design-thinking> (viitattu 29.09.2024)

Dam, Rikke Friis & Siang, Teo Yu. 2024. What is Design Thinking and Why Is It So Popular? [verkkosivu]. Luettavissa osoitteessa: <https://www.interaction-design.org/literature/article/what-is-design-thinking-and-why-is-it-so-popular> (viitattu 28.09.2024)

Gothelf, Jeff & Seiden, Josh. 2013. Lean UX: Applying Lean Principles to Improve User Experience. E-kirja. Sebastopol, CA: O'Reilly Media, Inc.

Hansson, David Heinemeier. N.d. The Rails Doctrine. [verkkosivu]. Katsottavissa osoitteesta: <https://rubyonrails.org/doctrine> (viitattu 28.09.2024)

21.9.2024)

HeatSpring. N.d. About HeatSpring: Workforce solutions for Climate Action. [verkkosivu]. Katsottavissa osoitteesta: <<https://www.heatspring.com/about>> (viitattu 27.08.2024).

HeatSpring. N.d. HeatSpring for Teams. [verkkosivu]. Katsottavissa osoitteesta: <https://www.heatspring.com/for_teams> (viitattu 27.08.2024)

Knight, Westley. 2019. UX for Developers: A Practical Guide to Implementing UX Principles in Your Work. E-kirja. Birmingham, UK: Packt Publishing.

Krug, Steve. 2014. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. E-kirja. San Francisco, CA: New Riders.

Laubheimer, Page. 2017. Personas vs. Jobs-to-Be-Done. [verkkosivu]. Luettavissa osoitteessa: <<https://www.nngroup.com/articles/personas-jobs-be-done/>> (viitattu 4.10.2024)

Lowdermilk, Travis. 2013. User-Centered Design: A Developer's Guide to Building User-Friendly Applications. E-kirja. Sebastopol, CA: O'Reilly Media, Inc.

Madsen, Rune. 2023. The gulf between design and engineering. [verkkosivu]. Luettavissa osoitteessa: <<https://designsystems.international/ideas/the-gulf-between-design-and-engineering/>> (viitattu 29.09.2024)

Nielsen, Jakob & Norman, Don. 1998. The Definition of User Experience (UX). [verkkosivu]. Luettavissa osoitteessa: <<https://www.nngroup.com/articles/definition-user-experience/>> (viitattu 4.10.2024)

Norman, Don. 2013. The Design of Everyday Things: Revised and Expanded Edition. E-kirja. New York: Basic Books.

Norman, Don. 2010. Design Thinking: A Useful Myth. [verkkosivu].

Luettavissa osoitteessa: <<https://www.core77.com/posts/16790/Design-Thinking-A-Useful-Myth>> (viitattu 28.09.2024).

Polaine, Andy & Løvlie, Lavrans & Reason, Ben. 2013. Service Design: From Insight to Implementation. E-kirja. Brooklyn, New York: Rosenfeld Media.

Ries, Eric. 2011. The Lean Startup: How Constant Innovation Creates Radically Successful Businesses. Yhdysvallat: Penguin Business.

Sakeus. N.d. HeatSpring: Yhdysvaltojen johtava koulutusala vihreän energia-alan ammattilaisille. [verkkosivu]. Katsottavissa osoitteessa: <<https://sakeus.fi/tyomme/heatspring>>(viitattu 27.08.2024).

Service Design Network. 2019. Service Design Glossary. [verkkosivu]. Luettavissa osoitteessa: <https://www.service-design-network.org/service-design-glossary> (viitattu 28.09.2024).

Singer, Ryan. 2019. Shape Up: Stop Running in Circles and Ship Work that Matters. E-kirja. Yhdysvallat: Basecamp.

Stickdorn, Marc & Hormess, Markus & Lawrence, Adam & Schneider, Jakob. 2018. This is Service Design Doing: Applying Service Design Thinking in the Real World. Kanada: O'Reilly Media, Inc.

Tailwindcss. N.d. [verkkosivu]. Katsottavissa osoitteessa: <<https://tailwindcss.com/>> (viitattu 22.09.2024).

Tuulaniemi, Juha. 2021. Palvelumuotoilu. 4. painos. Helsinki: Talentum Media. Luettavissa osoitteessa: <<https://bisneskirjasto-almatalent-fi.ezproxy.metropolia.fi/teos/BABBGXETEB#/kohta:Palvelumuotoilu/piste:tbw>> (viitattu 29.9.2024).

Wendt, Thomas. 2015. Design for Dasein: Understanding the Design of Experiences. E-kirja. CreateSpace Independent Publishing Platform.