

Olli-Pekka Riikola

**Digitaalisen kurssipohjan suunnittelu ja toteuttaminen
koodausleirien käyttöön**

Tietotekniikan pro gradu -tutkielma

16. joulukuuta 2024

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Olli-Pekka Riikola

Yhteystiedot: olli-pekka.v.riikola@jyu.fi

Ohjaaja: Petri Ihantola

Työn nimi: Digitaalisen kurssipohjan suunnittelu ja toteuttaminen koodausleirien käyttöön

Title in English: Design and Implementation of a Digital Course Template for Coding Camps

Työ: Pro gradu -tutkielma

Opintosuunta: Ohjelmisto- ja tietoliikennetekniikka

Sivumäärä: 52+3

Tiivistelmä: Tutkielmassa kuvataan, miten kehitettiin digitaalinen oppimisalusta tukemaan koodausleirien läpivientiä. Koodausleirien suosio on viime vuosina kasvanut ja samoin on kasvanut tarve koodausleiritoiminnan eri vaiheita tukevalle oppimisalustalle. Tämän suunnittelutieteellisen tutkielman artefaktina on koodausleireille suunnatun kurssipohjan prototyyppi, joka toteutettiin Jyväskylän yliopistossa kehitetyn TIM-alustan päälle. Kurssipohja tarjoaa kahdeksan ominaisuutta koodausleirin eri toimijoiden käyttöön.

Avainsanat: kurssipohja, koodausleiri, hackathon, hybridiopetus

Abstract: This thesis explores the development of a learning platform designed to support the implementation of coding camps. With the rising popularity of coding camps, there is a growing demand for a dedicated digital tool to assist in managing various phases of these camps. The core artifact of this Design-Based thesis is a prototype course template built on the TIM platform developed at the University of Jyväskylä. The prototype consists of eight features tailored to meet the needs of different coding camp stakeholders.

Keywords: course template, coding camp, hackathon, hybrid teaching

Esipuhe

Ensimmäiseksi haluan kiittää teitä, jotka sanoillanne ja neuvoillanne autoitte tutkielman alkumetreillä ja sen aikana. Kiitos Ilenia ja Tommi, saitte kaiken vaikuttamaan rohkaisevan yksinkertaiselta! Totta vai tarua, tässä sitä kuitenkin ollaan. Kiitos Petri graduni ohjaamisesta. Arvostan sitä, miten sait usein kiireen keskelle tuotua kiireettömyyden. Se rauhoitti työskentelyäni ja auttoi tarttumaan tekstiin uudella otteella. Kiitos myös Ristolle ja kaikille kanssani gradutunneilla istuneille. Kiitos Lauralle graduni lukemisesta ja kommentoimisesta!

Kiitos vaimoni Linda, ilman sinun korvaamatonta apuasi istuisin gradutunneilla todennäköisesti vielä lumien sulaessa! Ja kiitos lapsilleni jokaisesta hymystä, millä te pitkiä työskentelyiltojani piristitte. Niiden arvoa ei voi mitata millään menetelmällä.

Kiitos myös sinulle lukijani. Voit lukea tutkielman vaikka pala kerrallaan, siten se on kirjoitettu. Hyviä lukuhetkiä! :)

Jyväskylässä, 16. joulukuuta 2024

Olli-Pekka Riikola

Termiluettelo

Etäopetus	Opetusmuoto, jonka keskeinen ominaisuus on paikkariippumattomuus eli opettaja ja oppija voivat opetushetkellä olla eri paikoissa. Usein tässä käytetään apuna jotain teknologiaa.
Hackathon	Lyhyt yhteistoiminnallinen ongelmanratkaisuun keskittyvä koulutus- tai verkostoitumistapahtuma.
Hybridiopetus	Etäopetuksen ja perinteisen opetuksen yhdistetty muoto, jossa molempia opetusmuotoja sovelletaan samanaikaisesti jonkin teknologian avulla.
Iteratiivinen prosessi	Toistuva prosessi, jossa haluttu lopputulos saavutetaan toistamalla prosessin vaiheita uudelleen samalla esim. tuloksen vaatimuksia tarkentaen.
Kurssipohja	Digitaaliseen kurssialustaan toteutettu uudelleenkäytettävä kurssipohja.
Koodausleiri	Käytännönläheinen ohjelmoinnin opettamiseen keskittyvä hackathon-tyylinen intensiivikurssi.
LMS	Engl. Learning Management System tarkoittaa digitaalista oppimisalustaa, ohjelmistoa, joka tukee oppimista tarjoamalla työkaluja materiaalien jakamiseen, tehtävien hallintaan, arviointiin sekä vuorovaikutukseen opettajien ja opiskelijoiden välillä. Ne mahdollistavat usein myös ajasta ja paikasta riippumattoman oppimisen.
TIM	The Interactive Material on Jyväskylän yliopiston kehittämä interaktiivinen LMS-alusta.

Kuvat

Kuva 1. <i>Samoin kuin hackathon-tapahtumilla, myös koodausleireillä on kolme vaihetta. Alimmalla rivillä osoitetaan koodausleirin vaihe. Keskimmäisen rivin toiminta kuuluu osallistujalle ja ylimmällä rivillä kuvataan opettajan toiminta. Lisäksi vertaisohjaaja voi suorittaa opettajan toimintoja toteutusvaiheen kohdalla.</i>	6
Kuva 2. <i>Suunnittelututkimuksen eteneminen mukaillen kehyksiä, joita Hevner 2007 ja March ja Smith 1995 ovat esitelleet.</i>	12
Kuva 3. <i>Kuvassa C++ -ohjelmointikurssin harjoitustehtävä.</i>	13
Kuva 4. <i>TIM-alustan dokumenttirakenne ja navigointielementit voivat olla ajoittain vaikeaselkoisia.</i>	14
Kuva 5. <i>Prototyypin kehityksen prosessi sisälsi 14 vuorokauden iteraatiot.</i>	17
Kuva 6. <i>Kurssipohjan prototyypin rakenne. Kuva esittää hallintadokumentin ja molemmat kojelautanäkymät, niiden sisältämät ominaisuudet sekä materiaali- ja tehtävädokumentit ominaisuuksineen. Ominaisuuden nimen perään on kirjattu, mihin vaatimukseen se vastaa.</i>	20
Kuva 7. <i>Kuva esittää kurssipohjan hakemistorakenteen TIM-alustalla. Rakenne voidaan kopioida sellaisenaan uuden koodausleirin pohjaksi.</i>	21
Kuva 8. <i>Prototyypin kehitystä ohjattiin ja seurattiin Kanban-näkymän avulla.</i>	21
Kuva 9. <i>Luonnosteluvaiheessa havaittiin, että osallistujan ja opettajan kojelaudat voi kehittää pitkälti samanlaisista komponenteista. Ne kuitenkin toimivat käyttäjäroolista riippuen eri tavoin.</i>	22
Kuva 10. <i>Digitaalisen kurssipohjan näkymät luonnosteltiin aluksi paperille. Kuvassa on visualisoitu koodausleirin projektityön askelten seuranta.</i>	23
Kuva 11. <i>Osallistujan kojelauta sisältää kolme komponenttia, joiden avulla osallistuja voi seurata kurssin etenemistä.</i>	25
Kuva 12. <i>Askelkomponentin avulla opettaja voi visualisoida esim. projektin vaiheet. Koodausleirin osallistuja voi vaihtaa askelen tilaa, jonka perusteella opettaja voi seurata projektin etenemistä.</i>	26
Kuva 13. <i>Osallistujan käyttöliittymä sisältää interaktiivisen askelkomponentin, jonka avulla osallistuja voi seurata esim. projektin vaiheiden edistymistä.</i>	26
Kuva 14. <i>Opettajan ja osallistujan kojelaudat muistuttavat toisiaan, mutta opettajan kojelaudalla on yhteensä neljä komponenttia. Osallistujalistaus on tarkoitettu vain opettajan nähtäville.</i>	27
Kuva 15. <i>Materiaalidokumentti sisältää myös usein kysytyt kysymykset.</i>	29
Kuva 16. <i>Tehtävädokumentti sisältää konfiguraatiopohjat erityyppisille harjoitustehtäville.</i>	30
Kuva 17. <i>Kalenteri visualisoi koodausleirin aikataulun.</i>	31
Kuva 18. <i>Hallintadokumentissa on mahdollista perustaa uuden koodausleirin pohja kopiointityökalulla. Dokumenttien yläosaan lisättiin myös valikko navigointia varten.</i>	33

Sisällys

1	JOHDANTO	1
2	KOODAUSLEIRIT JA OPPIMISALUSTAT	4
2.1	Koodausleirien kohderyhmä ja tavoitteet	4
2.2	Koodausleirien toteutustapa	5
2.2.1	Esivaihe	6
2.2.2	Toteutusvaihe	7
2.2.3	Jälkivaihe	7
2.3	Oppimisalustat opetuksen tukena	7
3	TUTKIMUSMENETELMÄ JA TUTKIMUSONGELMA	10
3.1	Tutkimuksen lähtökohdat	10
3.2	Suunnittelutieteellinen tutkimus	11
3.3	Menetelmän soveltaminen	11
3.4	Konteksti	13
4	DIGITAALINEN KURSSIPOHJA KOODAUSLEIREILLE	16
4.1	Kurssipohjan kehittämisprosessi	16
4.2	Vaatimusmäärittely	16
4.3	Toteutus	19
4.3.1	Komponentit yleisesti	23
4.3.2	Osallistujan kojelauta	24
4.3.3	Opettajan kojelauta	27
4.3.4	Materiaalidokumentti	28
4.3.5	Harjoitustehtävädokumentti	28
4.3.6	Hallintadokumentti ja navigointi	29
4.4	Tuloksen arviointi	31
5	POHDINTA	34
5.1	Ominaisuuksien hyödyllisyys	35
5.2	Vaatimusmäärittelyn analysointi	37
5.3	Pienin toimiva tuote	38
5.4	Tulosten luotettavuus	39
5.5	Jatkotutkimus	40
6	YHTEENVETO	42
	LÄHTEET	43
	LIITTEET	47
	A Vaatimuslista	47

1 Johdanto

"Everyone should have at least a basic understanding of technology so they can look under the hood and see how things work. Coding should be part of general knowledge."

– Linda Liukas

Koodausleirit (engl. coding camps) ovat lyhyitä, tyypillisesti muutaman kymmenen tunnin mittaisia intensiivikursseja, jotka on suunnattu nuorille aikuisille (Fronza ja Corral 2024; Porras ym. 2018). Koodausleirit ovat luova tapa lisätä ohjelmoinnin opetuksen mielekkyyttä muuttamalla formaaleita opetustilanteita non-formaaliin suuntaan kokeilevalla ja käytännönläheisellä otteella. Ne ovat teoriapainotteisia toteutuksia mielekkäämpiä ja tarjoavat osallistujilleen luonnollisemman vuorovaikutusympäristön. Ne myös sitouttavat osallistujia paremmin kuin formaalit opetustilanteet (Gama, Alencar Gonçalves ja Alessio 2018).

Koodausleirien konseptin ymmärtämiseksi lienee helpompaa aloittaa hieman yleisemmästä konseptista, *hackathon*-tapahtumista. Hackathon-nimi on leikittelevä sanojen yhdistelmä, joka tulee toisiinsa sulautuneista englannin kielen sanoista *hack* ja *marathon*; suomalaisittain voisi sanoa *hakkerointimaraton* (Komssi ym. 2015). Tutkielmassa pitäydytään vieraskielisessä vastineessa.

"Hackathons are events where developers, programmers, designers and computer amateurs in general meet and work intensively to create software projects"

(Zapico ym. 2013).

Vaikka Hackathon-tapahtumille ei ole täsmällistä ja yleisesti hyväksyttyä määritelmää, kirjallisuudesta löytyy useita toisiaan tukevia määritelmiä. Yleisesti ottaen hackathon on intensiivinen tapahtuma, jossa osallistajat työskentelevät projektimaisesti ryhmissä yhteisen teeman parissa, ratkovat ongelmia ja tuottavat tyypillisesti lopputuloksena prototyypisovelluksen (Komssi ym. 2015). Samaan ajatukseen perustuvat myös koodausleirit ja muut vastaavat intensiiviset innovointi- ja kehittämistapahtumat, jotka voidaan kategorisoida yleisemmän hackathon-käsitteen alle (Porras ym. 2018). Ne on kuitenkin mahdollista erottaa toisistaan kontekstinsa perusteella.

Hackathon-tapahtumia järjestävät tahot ovat usein kunnallisia toimijoita tai yrityksiä, joilla on erilaisia tarpeita. Nämä toimijat hyödyntävät tapahtumia tyypillisemmin innovaatiotoimintaan tai jonkin liiketoiminnan ongelman ratkaisemiseen (Komssi ym. 2015). Koodausleirejä puolestaan järjestetään osallistujien kouluttamiseksi ja siellä heille tarjotaan erilaisia teknologioita luovan työskentelyn tueksi (Porras ym. 2018). Koodausleirien aikana käytetään erilaisia graafisia ohjelmointityökaluja, joilla tuetaan ongelmanratkaisua, parannetaan työskentelyn tehokkuutta ja saadaan yksilön potentiaali paremmin esille. Kokeneen ohjaajan johtamana koodausleirit rikastuttavat osallistujien oppimiskokemuksia, kannustavat ryhmätyöskentelyyn sekä ohjaavat kompleksisten haasteiden ratkaisemiseen monialaisen yhteistyön voimin (Fronza ja Corral 2024).

Koodausleirejä toteutetaan usein osana esim. yliopiston säännöllistä kurssitarjontaa, mutta ne voivat myös olla irrallaan järjestävän instituution muusta toiminnasta (Porras ym. 2019). Se voi haastaa koodausleirien järjestämistä, koska tällöin esim. osallistajat eivät välttämättä ole entuudestaan sidoksissa järjestävään instituutioon. Lisäksi tapahtuman läpivientiä haastaa sopivan digitaalisen alustan puute. Instituution ulkopuolisten osallistujien rekisteröimiseksi sekä muiden koodausleirin läpivientiin liittyvien toimien suorittamiseksi olisi hyödyllistä käyttää jotain digitaalista alustaa. (Fronza ym. 2024).

Viime vuosina koodausleirien suosio on kasvanut ja osallistujamäärät ovat olleet nousussa. Toteutustavasta riippuen määrät voivat olla jopa 150–200 henkilön luokkaa (Fronza ja Corral 2024). Suurempien osallistujamäärien sekä edellä kuvattujen haasteiden myötä koodausleirin digitaalisten työkalujen kehittäminen on tullut ajankohtaiseksi. Tässä tutkielmassa esitetään yksinkertainen ratkaisu siihen, miten koodausleirien tarpeisiin suunnattu digitaalinen alusta voisi tarjota ominaisuuksia helpottamaan tapahtuman koordinoimista sekä osallistujien pääsyä kurssin materiaaleihin ja aktiviteetteihin.

Tutkielman tavoitteena on suunnitella, kehittää ja arvioida digitaalinen kurssipohja koodausleirien tarpeisiin osana vuoden 2023 lopulla käynnistettyä EU-rahoitteista *OSCAR Coding camps* -hanketta (Promoting cross-cutting digital Skills through Europe-wide non-Conventional Learning experiences)¹. Hankkeessa kehitetään koodausleiritoimintaa niin käytäntöjen kuin digitaalisen oppimisalustankin kautta. Hankkeen työryhmätyöskentelyn tulok-

1. <https://oscar-codingcamps.eu/>

sena muodostettua oppimisalustan vaatimusmäärittelyä A ja ideapaperia (Fronza ym. 2024), käytetään aineistona tässä tutkielmassa. Tutkielman konsepti koodausleireistä pohjautuu Bolzanon vapaan yliopiston järjestämään vuosittaiseen *MobileDev²*-koodausleiriin.

Koodausleirejä ja digitaalisia oppimisalustoja taustoitetaan tiiviillä kirjallisuuskartoituksella luvussa 2. Luku 3 käsittelee tutkielman menetelmää ja tutkimuskysymystä. Luku 4 tarkastelee tutkielman empiiristä osuutta. Luvussa 5 tarkastellaan tutkielman rajoitteita sekä pohditaan tuloksia ja jatkotutkimusaiheita. Luku 6 sisältää yhteenvedon.

2. <https://mobiledev.inf.unibz.it/>

2 Koodausleirit ja oppimisalustat

"People rarely succeed unless they have fun in what they are doing"

– *Unknown*

Tässä luvussa käydään läpi koodausleirien käsitteistöä ja toteutusta sekä digitaalisten oppimisalustojen käyttöä opetuksessa. Koodausleirien kohderyhmä ja tavoitteet on kuvattu kappaleessa 2.1. Kappale 2.2 kuvaa koodausleirien toteuttamiseen liittyvät vaiheet. Kappale 2.3 käsittelee digitaalisia oppimisalustoja.

2.1 Koodausleirien kohderyhmä ja tavoitteet

Usein koodausleireillä tuetaan laaja-alaisesti erilaisten taitojen kehittymistä. Koodausleireihin sisällytettyjen aktiviteettien avulla osallistujat voivat kehittää henkilökohtaisia kykyjään eli ns. pehmeitä taitojaan (engl. soft skills) sekä substanssiosaamistaan eli ns. kovia taitojaan (engl. hard skills). Koodausleirien oleelliset aktiviteetit koostuvat sen aikana opiskeltavasta materiaalista ja harjoitustehtävistä sekä koodausleireillä suoritettavasta ryhmäprojektityöstä (Fronza ja Corral 2024).

Aktiviteettien vaikeustasoa kannattaa soveltaa kohderyhmälle sopivaksi (Porras ym. 2018). Kohderyhmänä voi olla ohjelmoinnista kiinnostuneet lukioikäiset tai jo perustason ohjelmointitaidon omaavat korkeakouluopiskelijat (Fronza ym. 2022; Porras ym. 2018). Esimerkiksi Italiassa järjestetyllä MobileDev-koodausleirillä osallistujat ovat olleet 15–19-vuotiaita nuoria monipuolisella taustalla eri kouluista (Fronza ym. 2022). Meillä Suomessa LUT yliopistossa järjestetyt vastaavat toteutukset on puolestaan suunnattu tietotekniikan opiskelijoille (Porras ym. 2018).

"Whether they are called camps, hackathons, or anything else with the same basic meaning, these intensive events bring together individuals from diverse backgrounds and collaboratively tackle complex challenges within a limited timeframe."(Fronza ja Corral 2024)

Koodausleirien projektityölle valitaan usein jokin todellisen elämän ongelmaan tai kehitys-

kohtaan liittyvä teema osallistujien motivaation ja suoriutumistason parantamiseksi (Porras ym. 2018; Fronza ym. 2022). Projektityön tarkempi aihe kannattaa jättää osallistujien itsensä valittavaksi, sillä sekin sitouttaa osallistujia paremmin koodausleirin toimintoihin ja tukee taitojen kehittymistä (Shakil ja Denny 2024).

Pääpiirteittäin koodausleireillä tuetaan samojen taitojen kehittymistä, mitä hackathon-tapahtumissa yleisemmin (Porras ym. 2019). Niihin lukeutuvat:

- Projekti- ja ryhmätyöskentelytaidot sekä yhteistyökyky.
- Puhe- ja esiintymistäidot.
- Ohjelmointitaidot ja kohdealueen tunnistaminen.
- Kekseliäisyys ja luovuus.
- Liiketoimintaosaaminen.
- Kriittinen ajattelu.
- Ongelmanratkaisutaidot.

2.2 Koodausleirien toteutustapa

Komssi ym. (2015), Page ja Sweeney (2016) ja Porras ym. (2019) esittivät hackathon-tapahtuman rakenteen kolmivaiheisena kokonaisuutena. Myös koodausleirit voivat koostua näistä kolmesta vaiheesta (Fronza ym. 2024). Vaiheet ovat *esivaihe*, *toteutusvaihe* sekä *jälkivaihe* ja ne on havainnollistettu kuvassa 1. Esivaihe voi sisältää erilaisia valmistavia toimia, kuten aiheen ja toteutuksen ideointia sekä projektiryhmien muodostamista. Varsinainen toteutusvaihe sisältää kaiken työkalujen ja teknologian haltuunoton ja soveltamisen valmiin lopputuloksen saavuttamiseksi. Jälkivaihe puolestaan sisältää arviointia siitä, jatketaanko lopputuloksen kehittämistä vai joutaako raakileeksi jäänyt toteutus pöytälaatikkoon. Jatkokehitys voi käynnistyä myös uudella ryhmäkokoontamalla ja eri toteutusteknologioilla (Page ja Sweeney 2016, kuva 1).

Koodausleirin toteuttamiseen osallistuu tyypillisesti useampi henkilö. Toteuttamiseen voidaan tarvita yksi tai useampi *opettaja* kurssin toimintojen läpivientiin sekä *vertaisohjaajia* avustamaan projektitöissä. Opettajat vastaavat koodausleirin suunnittelusta ja järjestämisestä esivaiheesta jälkivaiheeseen. Vertaisohjaajat voivat olla esim. koodausleirille aiemmin osal-

Logistiikka	Materiaali	Ryhmänmuodostus	Ohjaus	Tekninen tuki	Todistukset	Arviointi
Ennakkotehtävät		Ideointi	Projektityö	Harjoitustehtävät	Jatkokehitys	
Esivaihe			Toteutus		Jälkivaihe	

Kuva 1. Samoin kuin hackathon-tapahtumilla, myös koodausleireillä on kolme vaihetta. Alimmalla rivillä osoitetaan koodausleirin vaihe. Keskimmäisen rivin toiminta kuuluu osallistujalle ja ylimmällä rivillä kuvataan opettajan toiminta. Lisäksi vertaisohjaaja voi suorittaa opettajan toimintoja toteutusvaiheen kohdalla.

listuneita opiskelijoita, jotka auttavat osallistujia leirin projektityössä. Koodausleirin suurin henkilöryhmä muodostuu usein *osallistujista*, jotka suorittavat koodausleirille järjestettyjä aktiviteetteja (Fronza ja Corral 2024).

2.2.1 Esivaihe

Esivaiheessa *järjestäjä* voi määrittää koodausleirin tehtäviä ja tavoitteita, mitä taitoja tarvitaan tai mitä taitoja tapahtumassa voi kartuttaa. Järjestäjä voi myös selvittää ennen tapahtuman alkua mahdolliset viralliset dokumentit ja logistiset kysymykset, kuten luokkatilan, laitteiden ja viestivälineiden tarpeen sekä määrittää aikataulun. Järjestäjä huolehtii koodausleirin ohjaukselliset yksityiskohdat mukaan lukien avustajien rekrytoimisen, materiaalien laittamisen ja muiden tarvittavien resurssien valmistelemisen. Esivaiheessa täytyy varmistua henkilökunnan olevan valmiina tapahtuman toteutusvaiheeseen (Fronza ym. 2024).

Vielä ennen koodausleirin alkua, *osallistujat* voivat valmistautua toteutusvaiheeseen seuraamalla järjestäjän antamia ohjeita ja tutustumalla mahdollisesti jo julkistettuun materiaaliin. Heidän on myös mahdollista harjoitella toteutusvaiheessa suoriutumista ja työkalujen käyttöä ennakkotehtävien avulla. Voi olla myös muita toimia, joita esivaiheessa olisi hyödyllistä tehdä, kuten oppimistavoitteiden asettaminen (Fronza ym. 2024).

2.2.2 Toteutusvaihe

Toteutusvaihe on koodausleirien rakenteen ydin. Tässä vaiheessa *järjestäjä* panee toimeen suunnitellut harjoitustehtävät luentoineen sekä ohjaavat harjoitustyön. Toteutusvaiheessa järjestäjän on myös tärkeää tarjota riittävä ohjauksellinen ja tekninen tuki niin harjoitustehtävien kuin projektityönkin pulmiin, sekä tarkkailla ryhmien työskentelyä palautteen antamiseksi ja vaiheen läpiviennin hienosäätämiseksi (Fronza ym. 2024).

Harjoitustehtävien, käytännön työskentelyn ja saadun palautteen avulla *osallistujat* hankkivat uusia taitoja ja oppivat ryhmätyöskentelytaitoja. Oleellinen aktiviteetti koodausleirillä on harjoitustyö, jonka osallistujat tekevät yhteistyössä esivaiheessa muodostetun ryhmänsä kanssa. Toteutusvaiheen aikana annetun palautteen myötä heidän on mahdollista myös sopeuttaa projektityönsä etenemistä iteratiivisesti (Fronza ym. 2024).

2.2.3 Jälkivaihe

Jälkivaiheessa oleellista on katsahtaa menneeseen ja arvioida koodausleirin toteutusta. *Järjestäjä* arvioi osallistujien suoritukset ja kirjaa ne mahdollisesti todistukseen. Jälkivaiheessa voidaan kerätä osallistujapalautte, joka muun kerätyn datan ohella analysoidaan esim. tilannekatsausten, tapahtuman raportoinnin ja kehityskohtien tunnistamisen tueksi. Käytännön syistä jälkivaiheessa ei osallistujapalautteen lisäksi ole helppoa toteuttaa muuta vuorovaiikutusta järjestäjien ja osallistujien välillä. Tästä syystä tuki ja ohjaus koodausleirin aikana opittujen taitojen kehittämiseen katkeaa tyypillisesti kokonaan. Lisäksi projektityön mahdollinen jatkekehittäminen jää osallistujien oman aktiivisuuden varaan. Usein *osallistujien* ainoa aktiviteetti jälkivaiheessa onkin osallistujapalautteen antaminen (Fronza ym. 2024).

2.3 Oppimisalustat opetuksen tukena

Nykyään on tyypillistä, että korkeakouluopetusta järjestetään *etäopetuksena* käyttäen apuna jotain digitaalista *oppimisalustaa* (Neuwirth, Jović ja Mukherji 2021). Etenkin COVID-19 pandemia vauhditti etätöön ja etäopetuksen yleistymistä, ja sen myötä on yleistynyt erilaisen etätöskentelyä tukevien sovellusten käyttäminen (Ulla ja Perales 2022). Esimerkiksi tietotekniikkaa opetetaan useimmiten verkossa, erilaisilla digitaalisilla oppimisalustoilla (Pra-

seeda ym. 2023). Lisäksi alustojen tuella myös tietotekniikan opetuksessa käytettävät oppimateriaalit ovat kehittyneet (Brusilovsky ym. 2023).

Oppimisalustojen käyttö on mahdollistanut tietotekniikan opetuksen tavoilla, joita aiemmin ei kirjojen, luentoesitysten, harjoitustöiden tai kysymyspattereiden avulla voinut toteuttaa. Parin vuosikymmenen aikana tietotekniikan opetus on kehittynyt huomattavasti ja aiempien opetusmateriaalien lisäksi opiskelijoille tarjotaan videoita, animaatioita sekä automaattitarkisteisia ohjelmointitehtäviä oppimisen ja harjoittelun tueksi (Brusilovsky ym. 2023). Tehtävien automaattinen tarkistaminen ja arviointi sekä palautteen antaminen suoritettujen tehtävien jälkeen on koettu hyödylliseksi ominaisuudeksi oppimisalustoilla. Hyvin toteutetulla automaattisella arvioinnilla parannetaan oppimista ja vähennetään opettajan työkuormaa (Hooper ym. 2024).

Opetus voidaan yhä useammin toteuttaa *hybridiopetuksena*, joka tarkoittaa opetuksen toteuttamista samanaikaisesti sekä lähiopetuksena että etäopetuksena (Ulla ja Perales 2022). Näin toimitaan esimerkiksi koodausleireillä; digitaaliset työkalut ja hybridiopetus ovat usein keskeisessä roolissa koodausleirejä järjestettäessä (Fronza ja Corral 2024; Happonen, Tikka ja Usmani 2021).

Jotta digitaalinen alusta tukisi etä- tai hybridiopetusta riittävällä tavalla, siltä vaaditaan tiettyjä ominaisuuksia. Viestinvälitys, sisällön tuottaminen ja sen hallinta ovat digitaalisen oppimisalustan keskeisiä piirteitä (Moore, Dickson-Deane ja Galyen 2011; Meylani, Bitter ja Legacy 2015). Samoin keskeistä on toteuttaa tuki interaktiivisille tutoriaaleille, harjoitustehtäville ja kyselyille. Voidaan myös tukea erilaisia oppijoita, eri opetusmuotoja ja alustan muokattavuutta (Meylani, Bitter ja Legacy 2015). Interaktiivinen digitaalinen alusta tarjoaa paremmat lähtökohdat opettaa ja oppia ohjelmointia (Robinson ja Carroll 2017).

Erityyppisiä digitaalisia oppimisalustoja on melko paljon ja niiden tekniset toteutustavat vaihtelevat. Moore, Dickson-Deane ja Galyen (2011) mainitsevat ainakin oppimisenhallintajärjestelmän (engl. learning management system eli LMS), kurssinhallintajärjestelmän (engl. course management system eli CMS), virtuaalisen oppimisympäristön (engl. virtual learning environment eli VLE) ja vielä tietämyksenhallintajärjestelmän (engl. knowledge management system eli KMS). Alusta voi toimia myös online-oppimisympäristönä (engl. online

learning environment eli OLE) (Meylani, Bitter ja Legacy 2015). Huomataan, että tilanteesta riippuen voidaan puhua järjestelmästä tai laajemmin kokonaisesta ympäristöstä. Kaikista esitellyistä alustoista voidaan käyttää yleisempää termiä *oppimisalusta* (engl. learning platform) ja sitä käytetään myös tämän tutkielman kontekstissa.

3 Tutkimusmenetelmä ja tutkimusongelma

"Planning problems are wicked problems."

– H. Rittel

Tutkielman tavoitteena on selvittää suunnittelutieteen keinoin, miten voitaisiin toteuttaa digitaalinen kurssipohja koodausleireille. Tässä luvussa esitellään tutkimuksen lähtökohdat ja tutkimuskysymys sekä käsitellään suunnittelutieteellistä tutkimusmenetelmää. Tutkimusasetelma ja tutkimuskysymys esitellään kappaleessa 3.1. Menetelmää kuvataan yleisesti kappaleessa 3.2. Kappale 3.3 käsittelee tutkielmatyöhön sisältyvän artefaktin kehityksen vaiheita suunnittelutieteellisen tutkimusprosessin aikana. Lisäksi tutkielman konteksti esitellään tarkemmin kappaleessa 3.4.

3.1 Tutkimuksen lähtökohdat

MobileDev koodausleiriä on järjestetty vuosittain toistuvana yli kymmenen vuoden ajan tarjoamaan osallistujille oppia Agile-pohjaisesta sovelluskehityksestä. Tapahtumaan on tarkoituksena saada monialainen osallistujajoukko, jotta osallistujilla olisi kattavat mahdollisuudet toteuttaa ohjelmistotekninen ryhmätyö sekä harjoitella projektityöskentely- ja ongelmanratkaisutaitoja monialaisessa ympäristössä (Fronza ym. 2020).

Koodausleiritoiminta kehittyy jatkuvasti ja tavoitteena on muokata toimintaympäristö entistä digitaalisemmaksi yhtä aikaa säilyttäen mm. online-koodausleirien vetovoimaisuus ja hauskuus samalla tasolla verrattuna aiempaan kontaktiopetukseen. Etäpainotteisempi opetusmuoto ei saisi esim. vaikuttaa negatiivisesti yhteenkuuluvuuden tunteeseen tai osallistujien sitoutumiseen eikä se saisi johtaa pitkittyvän tietokoneenkäytön aiheuttamaan väsymiseen (Fronza ym. 2022).

Toistaiseksi koodausleiritoiminnan kehitystä haastaa koodausleirien irrallisuus koulutusinstituution muusta toiminnasta tai kurssitarjonnasta. Tällaisen vakiintumattoman koodausleirin osallistajat eivät välttämättä tule järjestävän koulutusinstituution sisältä (Porras ym. 2018). Se voi aiheuttaa järjestäjälle ylimääräistä hallinnollista työtä esimerkiksi henkilötietojen ke-

räämisen ja hallinnoinnin muodossa.

Lisäksi etä- tai hybridiopetuksena järjestettävillä koodausleireillä ei ole käytössään sopivaa digitaalista alustaa, jonka avulla läpivienti onnistuisi sujuvasti. Alustan puuttuessa materiaalin jakaminen, koodausleirin projektiryhmien muodostaminen ja työskentelyn seuraaminen on haasteellista. Myös yksilöllisten projektitodistusten tuottaminen tai projektin mahdollisen jatkokehityksen seuraaminen voi työkalujen puuttuessa olla hankalaa (Fronza ym. 2024). Sopivan alustan kehittämiseksi tarvitaan jokin koodausleirien tarpeisiin soveltuva innovatiivinen ratkaisu. Tämä voitaisiin saavuttaa esimerkiksi kehittämällä uusi digitaalinen alusta koodausleirien käyttöön tai jatkokehittämällä jotain tarkoitukseen soveltuvaa oppimisalustaa (Fronza ym. 2024). Tutkimuskysymyksemme on:

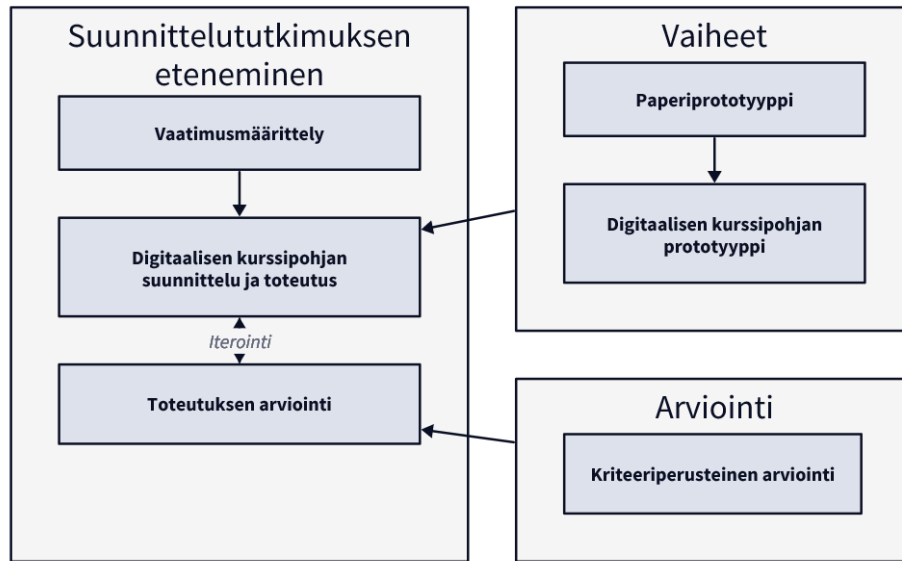
Miten voitaisiin suunnitella ja toteuttaa digitaalinen kurssipohja koodausleirien käyttöön?

3.2 Suunnittelutieteellinen tutkimus

Useimmissa tapauksissa, kuten myös tietojärjestelmien tutkimisessa ja kehittämisessä, suunnittelututkimus toimii pohjimmiltaan ongelmanratkaisuparadigmana. Yleisesti ottaen se sisältää syklejä, jotka puolestaan sisältävät iteratiivisia toimintoja artefaktin kehittämiseksi. Prosessin tuloksena saadaan täsmällisesti suunniteltu ja arvioitu innovatiivinen lopputuote (Hevner ym. 2004; March ja Smith 1995). Tietojärjestelmäkentällä suunnittelutieteellinen tutkimus on luonteeltaan proaktiivista. Sen keinoin pyritään luomaan innovatiivisia artefakteja sekä auttamaan organisaatioita vastaamaan tietotyön haasteisiin (Hevner ym. 2004). Proaktiivisuuden lisäksi Hevner ym. 2004 mukaan tutkimuksen tulisi olla myös reaktiivista. Näin tutkielman aikana kerättyä tietoa voidaan hyödyntää tuotteen kehittämisessä ja arvioinnissa.

3.3 Menetelmän soveltaminen

Tutkimuskysymykseen vastataan suunnittelutieteellisen menetelmän avulla, joka sisältää vaatimusmäärittelyn, iteratiivisen suunnittelun ja toteutuksen sekä arvioinnin. Tämä prosessi on



Kuva 2. Suunnittelututkimuksen eteneminen mukaillen kehäksiä, joita Hevner 2007 ja March ja Smith 1995 ovat esitelleet.

esitetty kuvassa 2. *Vaatusmäärittelyssä* selvitetään millaisia vaatimuksia kurssipohjan toiminnalle on asetettu. Sen jälkeen alkaa *suunnittelu ja toteutus*, joka sisältää artefaktin iteratiivista suunnittelua ja kehittämistä määriteltyjen vaatimusten pohjalta. Toteutuksen jälkeen suoritetaan *arviointi*, jossa pyritään selvittämään miten toteutettu kurssipohja täyttää sille asetetut vaatimukset.

Tutkielmassa vaatimusmäärittely tehtiin *fokusryhmäkeskusteluna*, jossa kaikki digitaalista kurssipohjaa käyttävät roolit huomioitiin. Fokusryhmä muodostettiin kutsumalla koolle kurssipohjaa kehittävän instituution edustajat sekä ne OSCAR Coding Camps hankkeen tutkijat, jotka vastaisivat koodausleirien järjestämisestä omassa instituutiossaan. Fokusryhmäkeskustelussa sovellettiin roolijakoa jakamalla jokaiselle keskustelijalle jonkin kappaleessa 2.2 esitellyistä rooleista.

Vaatusmäärittelyn tekemiseksi tarvitaan keskustelua ja kohdealueen ymmärtämistä. (Stephen Lane ja Sammon 2016). Fokusryhmäkeskustelun tukena käytimme koodausleirin konseptin esittelevää asiantuntijaluentoa. Keskustelun tuloksena vaatimukset kirjattiin ylös ja ne koottiin vaatimuslistaan A. Julkaisimme myös vaatimuksia kokoavan ideapaperin "*Towards s'more connected coding camps*" (Fronza ym. 2024).

Tee aliohjelma `pienimman_paikka`, joka palauttaa kokonaislukutaulukon pienimmän alkion paikan (indeksin). Esim. monisteen luvun 13.1.4 taulukosta

```
int k_pituudet[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
```

`pienimman_paikka` palauttaa arvon 1. Tee myös edellistä aliohjelmaa käyttäen aliohjelma `pienin`, joka palauttaa kokonaislukutaulukon pienimmän alkion arvon.



```
1
2
3 /* Kirjoita tarvittavat aliohjelmat */
4
5 int main(void)
6 {
7
8 /* Täydennä ohjelman toiminta */
9
10 return 0;
11 }
```

Aja Test Document Tavallinen Copy

Kuva 3. Kuvassa C++ -ohjelmointikurssin harjoitustehtävä.

Suunnittelu ja toteutus sisälsivät sekä paperisten luonnosten että varsinaisen kurssipohjan prototyypin kehittämisen. Prototyyppeihin kehitetyt ominaisuudet vastasivat vaatimusmäärittelyssä koottuihin vaatimuksiin. Toteutus sisälsi iteraatioita, joiden aikana pystyttiin havainnoimaan ja arvioimaan kurssipohjaan parhaillaan kehitettäviä ominaisuuksia.

Artefaktin arviointiin sovellettiin keinotekoisia, kriteeriperusteista arviointia, nk. kuvailevaa arviointitapaa (Hevner ym. 2004). Tutkielman aikana prototyypillä ei ollut oikeita käyttäjiä, joten kriteeriperusteinen arviointi oli luonteva valinta tutkielman kontekstissa (Cronholm ja Goldkuhl 2003). Arviointitavan valintaan vaikuttivat myös vähäiset resurssit, ja usein keinotekoinen arviointi säästää aikaa sekä varoja (Venable, Pries-Heje ja Baskerville 2012). Arvioinnissa käytettiin kriteeristönä prototyypille asetettuja vaatimuksia.

3.4 Konteksti

Kuten todettua, erilaisia digitaalisia oppimisalustoja on käytössä paljon, niitä kehitetään ja aiheita tutkitaan jatkuvasti. Joissain tilanteissa nykyiset oppimisalustat eivät kuitenkaan vastaa yksittäisen opetuksen järjestäjän tarpeita. Silloin mahdollisuutena on joko kehittää jokin oma oppimisalusta tai kehittää olemassa olevaa alustaa. Mm. Robinson ja Carroll (2017)

Kuva 4. *TIM*-alustan dokumenttirakenne ja navigointielementit voivat olla ajoittain vaikeaselkoisia.

kehittivät *Moodle*¹-alustaa tukemaan paremmin ohjelmoinnin opetusta ja arviointia.

Tässä tutkielmassa jatkokehitetään Jyväskylän yliopistossa kehitettyä oppimisalustaa, *TIM*-alustaa². *TIM* (The Interactive Material) on avoimen lähdekoodin oppimisalusta, jota voidaan käyttää opetuksen tukena eri opetusmuodoissa. Opettaja voi luoda alustalle harjoitus-tehtäviä ja kurssimateriaaleja. Opiskelijat puolestaan voivat selata sisältöjä ja suorittaa tehtäviä. *TIM* perustuu dokumenttipohjaisuuteen eli käytännössä kaikki alustan sivut ovat *dokumentteja*. Dokumentit koostuvat *lohkoista* ja lohkon sisältönä voi olla teksti, kuva, video tai esim. interaktiivinen *komponentti* kuten kuvassa 3. Useimmat komponentit vaativat konfigurointia toimiakseen. Edellä mainittujen ominaisuuksien lisäksi alusta tukee suoritusten seuranta ja oppimisdatan visualisointia.

Teknologiapinoltaan *TIM* on moderni web-sovellus, jossa kontteihin hajautettu palvelinpuoli tarjoaa selainpuolelle rajapinnan toimintojen suorittamiseksi. Konttiratkaisut on kehitetty käyttäen *Docker*-moottoria ja palvelinpuoli on kehitetty *Python*-ohjelmointikielellä sekä sille kehitetyn *Flask*-sovelluskehityksen tuella. Selainpuolella käytetään *TypeScript*-ohjelmointikieltä ja *Angular*-sovelluskehystä sekä *Jinja*-moottoria templaattien luomiseen. Tietokannanhal-

1. <https://moodle.com>

2. <https://tim.jyu.fi>

laintajärjestelmänä alustalla on *PostgreSQL* ja tietokantapyyntöjen käsittelemiseen käytetään Pythonin *SQLAlchemy*-moduulia pyyntöjen eheyden varmistamiseksi.

TIM-alustan keskeinen ajatus on vähentää opiskelijan ja opettajan kognitiivista kuormaa tarjoamalla kaikki oppimiseen ja opettamiseen tarvittavat toiminnot saman alustan sisällä. Kun materiaalit ja tehtävät jaellaan samalla alustalla, käyttäjän ei tarvitse siirtyä eri alustojen välillä ja tieto on saatavilla keskitetympin. Joskus ongelmana voi kuitenkin olla alustan sisäinen rakenne kuten kuvassa 4. Perustaessaan uutta kurssia kurssipohjan kehittäjä, useimmiten opettaja, on itse vastuussa kurssin rakenteesta alustalla.

4 Digitaalinen kurssipohja koodausleireille

"Talk is cheap. Show me the code."

– *Linus Torvalds*

Luvussa esitellään tutkielmassa kehitetyn artefaktin vaiheet vaatimusmäärittelystä arviointiin. Kappale 4.1 kuvaa kehityksen lähtökohdat ja rajauksen. Vaatimusmäärittely, sen koostaminen ja lopputulos on kuvattu kappaleessa 4.2. Artefaktin toteutus ja siihen sisältyvät komponentit on esitelty kappaleessa 4.3. Artefaktin arviointi on tehty kappaleessa 4.4.

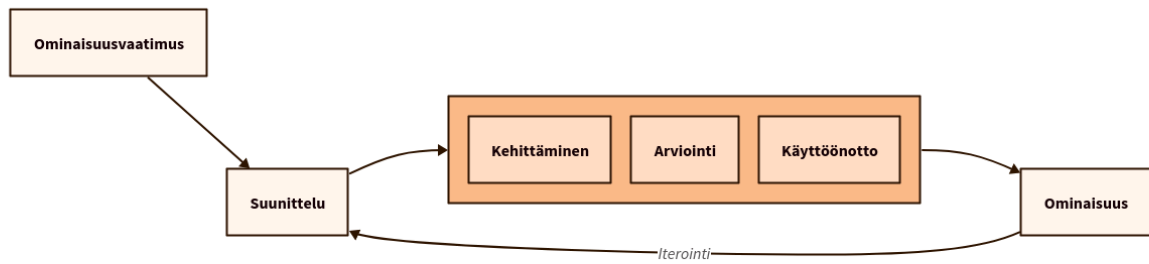
4.1 Kurssipohjan kehittämisprosessi

Tutkielman empiirisessä osassa kehitettiin digitaalisen kurssipohjan prototyyppi kappaleessa 3.4 esitellyn TIM-alustan päälle. Aktiivisessa käytössä olevana oppimisalustana se tarjoaa jo entuudestaan paljon käyttökelpoisia ominaisuuksia erilaisten kurssien järjestämiseen. Täsmällisemmän lopputuloksen saavuttamiseksi alustaan kehitettiin vielä erikseen koodausleireille suunnattuja ominaisuuksia.

Kehitysprosessi kurssipohjan tuottamiseksi noudatteli kuvassa 2 kuvattua prosessia. Työ alkoi vaatimusmäärittelystä, joka toteutettiin fokusryhmäkeskusteluna. Varsinainen ohjelmistokehitys käynnistyi tehtävien määrittämisellä ja paperiprototyypin kehittämisellä, josta edettiin iteratiiviseen ominaisuuksien kehittämiseen kuvan 5 esittämällä tavalla. Ohjelmistokehityksen aloittamisen yhteydessä käynnistettiin 14 vuorokauden ohjelmistokehityksen iteraatiot, jotka tarjosivat tarkistuspisteitä ohjelmiston kehittämisen etenemiselle. Tässä tutkielmassa rajaudutaan neljään ensimmäiseen iteraatioon, joiden jälkeen kurssipohjan prototyyppi arvioitiin ja suunnittelututkimuksen prosessi päättyi. Siitä eteenpäin prototyypin kehitys jatkui tutkielmasta erillään OSCAR Coding Camps -hankkeessa.

4.2 Vaatimusmäärittely

Kuvan 2 mukaisesti tutkielman ensimmäinen osuus oli vaatimusmäärittely. Sen pohjana käytettiin vaatimusmäärittely tilaisuuden alussa nähtyä asiantuntijaluentoa, jonka antoi läs-



Kuva 5. Prototyypin kehityksen prosessi sisälsi 14 vuorokauden iteraatioita.

näöllut MobileDev-koodausleirin järjestäjä *Ilenia Fronza*. Luennolla esiteltiin MobileDev-koodausleirin taustoja, joita on esitelty myös tutkielman luvussa 1. Lisäksi kuvattiin mm. kohderyhmää ja tavoitteita (ks. kappale 2.1) sekä koodausleirin vaiheet ja koodausleireillä toimivat roolit (ks. kappale 2.2. Luennon aikana esiteltiin myös artikkelit "*Keeping Fun Alive: an Experience Report on Running Online Coding Camps*" (Fronza ym. 2022) sekä "*Nurturing Hybrid Work Literacy in Upper Secondary Schools: Selecting the Best Hybrid Work Configuration for Coding Camps*" (Fronza, Iaccarino ja Corral, (Hyväksytty julkaistavaksi)). Luennon ja artikkeleiden tuella pyrittiin muodostamaan yhteinen käsitys koodausleirin konseptista fokusryhmäkeskustelua varten.

Vaatusmäärittelyä varten koottiin kappaleessa 3.4 esitelty fokusryhmä, joka kävi tapauskohtaisesti läpi koodausleirin vaiheet erilaisia skenaarioita tarkastellen ja niistä keskustellen. Järkevän vaatimusmäärittelyn aikaansaamiseksi skenaarioita tarkasteltiin koodausleirin eri rooleissa siten, että jokaisella keskustelijalla oli jokin koodausleirin toimijan rooli. Keskustelun perusteella esiin nousseita vaatimuksia kirjattiin muistiin niin opettajan, vertaisohjaajien kuin osallistujienkin näkökulmasta.

Vaatusmäärittelyn tuloksena saatiin vaatimuslista A digitaalisen alustan kehittämistä varten. Lista priorisoitiin sen mukaan, mitkä vaatimukset mahdollistaisivat pienimmän toimivan tuotteen kehittämisen. Valmis vaatimuslista oli koko kurssipohjan käytännön kehittämistyön lähtökohta.

Fokusryhmäkeskustelun tuella ryhmä julkaisi vaatimuslistan lisäksi ideapaperin, joka sisälsi ehdotelman parannellusta koodausleiristä ja sen toimintaa tukevasta digitaalisesta alustasta. Alustan tulisi jossain muodossa sisältää ideapaperissa esitellyt ominaisuudet (Fronza

ym. 2024). Ideapaperin ja vaatimusmäärittelyn perusteella tutkielmaan koottiin 11 vaatimusta digitaaliselle oppimisalustalle.

Vaatimukset voidaan jakaa kolmeen teemaan. Yksi keskeinen ajatus oli, että alusta sisältäisi *kojelautanäkymät* (engl. dashboard view) opettajalle ja osallistujille koodausleirin yleiskatsaukseen ja perustoimintojen suorittamiseen. Lisäksi alustalta vaadittiin *aputyökaluja sekä oppimateriaalinäkymiä* tukemaan koodausleirin perustamista sekä harjoittelua ja ohjausta. Lisäksi alustan toivottiin tukevan *koodausleirin integroitumista* osallistujien sidosryhmiin tai intresseihin eli esim. kouluihin, työpaikkoihin tai harrastetoimintaan (Fronza ym. 2024). Vaatimukset on listattu alla:

Kojelautanäkymien vaatimukset

- V1 *Aikajanan näyttäminen.* Jana voisi osoittaa mm. tulevat tehtävät ja niiden aikataulun koodausleirillä.
- V2 *Osallistujien tietotaitojen jakaminen* käyttäjäprofiilissa projektiryhmää ja yritysyhteistöitä varten.
- V3 *Työkalut osallistujatietojen keräämiseen.* Nämä tiedot voivat auttaa mm. ryhmänmuodostuksessa ja projektin suunnittelussa.
- V4 *Projektityön suunnittelu ja kehittäminen askel askeleelta.* Tämän tulisi toimia siten, että osallistuja voi seurata ryhmänsä etenemistä koodausleirin projektityössä.

Aputyökalujen ja oppimateriaalin vaatimukset

- V5 *Etätyökalujen tarjoaminen* etä- ja hybridiopetuksessa oleville osallistujille.
- V6 *Yhteydenpidon mahdollistaminen järjestäjän ja osallistujien välillä.* Viestintäkanavat kuten pikaviestin tai bloginäkömä täytyisi olla saatavilla kommunikaatiota varten.
- V7 *Ennakkotehtävien tuottaminen* tukemaan mahdollisia osallistujia koodausleirin alkuvaiheessa.
- V8 *Kurssipohjan uudelleenkäyttäminen.* Koodausleirin järjestäjä voi halutessaan kopioida valmiin kurssipohjan omaan käyttöönsä.

Toiminnan integroinnin vaatimukset

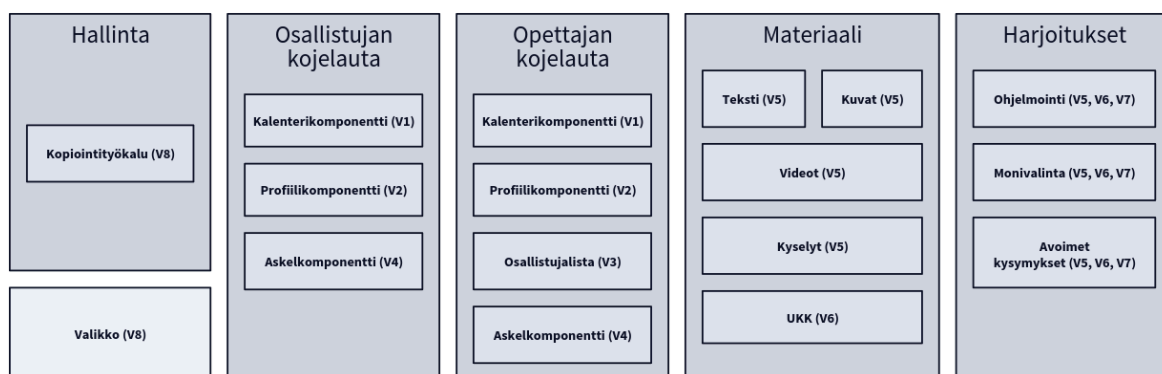
- V9 *Koodausleiriltä saatavien taitojen visualisointi ja niiden arviointi.* Kojelautanäkymässä voisi visualisoida arviointien tuloksia henkilökohtaisesti sekä ryhmätasolla. Lisäksi esim. tehtäväsuorituksista saatua tietoa voisi käyttää opetuksen pelillistämiseksi.
- V10 *Koodausleirin projektien julkistaminen.* Tukee osallistujien pidemmän aikavälin tavoitteita, esim. oman GitHub koodivaraston tai henkilökohtaisen portfolion ylläpitämistä. Osallistujat voisivat julkistaa projektityönsä alustalla ja linkittää ne sieltä portfolioonsa.
- V11 *Raportin generointi koodausleirin jälkeen.* Raportti voisi sisältää kyselydataa ja tarjota tietoa koodausleirin tuloksista organisaatioille.

Tutkielman aikana vastattiin kaikkiaan kahdeksaan vaatimukseen ja jokainen näistä vaatimuksista vastasi yhtä kurssipohjaan sisällytettyä ominaisuutta. Näihin vaatimuksiin vastamalla pyrittiin kehittämään kurssipohjan prototyyppi. Toiminnan integrointiin liittyviin vaatimuksiin ei tutkielman aikana ehditty vastaamaan.

4.3 Toteutus

Tutkielman artefaktina kehitettiin uudelleenkäytettävä kurssipohja; iteratiivisen kehitysprosessin aikana siihen konfiguroitiin ja ohjelmoitiin koodausleirin kannalta oleellisia ominaisuuksia, jotka koodausleirin järjestäjä voisi ottaa kerralla käyttöön kopioimalla kurssipohjan itselleen uutta koodausleiriä varten. Kaikkien kurssipohjaan sisällytettyjen ja kehitettyjen ominaisuuksien tarkoituksena on helpottaa koodausleirien järjestämistä ja niihin osallistumista.

Kurssipohja koostettiin viidestä TIM-dokumentista ja se sisälsi yhteensä kahdeksan vaatimusmäärittelyssä kuvattua ominaisuutta. Ominaisuuksista viisi toteutettiin hyödyntämällä alustan valmiita ominaisuuksia ja kolme kehitettiin suoraan koodausleirien tarpeisiin käyttäen TIM-alustan arkkitehtuurin mukaisia teknologioita (ks. kappale 3.4). Kurssipohja vastasi ominaisuuksillaan kahden ensimmäisenä listatun teeman vaatimuksiin tarjoamalla toteutusratkaisut ja tuen kojelautanäkymille, kurssipohjan aputyökaluille ja koodausleirillä julkaistavalle materiaalille. Vaatimukset koskien koodausleirin integroimista osallistujan sidos-



Kuva 6. Kurssipohjan prototyypin rakenne. Kuva esittää hallintadokumentin ja molemmat kojelautanäkymät, niiden sisältämät ominaisuudet sekä materiaali- ja tehtävädokumentit ominaisuuksineen. Ominaisuuden nimen perään on kirjattu, mihin vaatimukseen se vastaa.

ryhmiin rajattiin pois tutkielman toteutuksesta, mutta niitä käsitellään myöhemmin luvussa 5.

Kuva 7 esittää dokumenttirakenteen, jonka mukaisesti kurssipohjan dokumenttirakenne luotiin TIM-alustalle. Alustaan luotiin *osallistujan kojelauta* ja *opettajan kojelauta* koodausleirin yleisnäkymälle ja perustoiminnoille sekä kaksi muuta TIM-dokumenttia koodausleirin *materiaalille* ja *harjoitustehtäville*. Lisäksi kurssipohjaan sisällytettiin *hallintadokumentti*, joka prototyypissä sisältää koodausleirejä varten kehitetyn kurssipohjan kopiointityökalun. Navigointia varten jokaisen dokumentin alkuun lisättiin *valikko* painikkeineen.

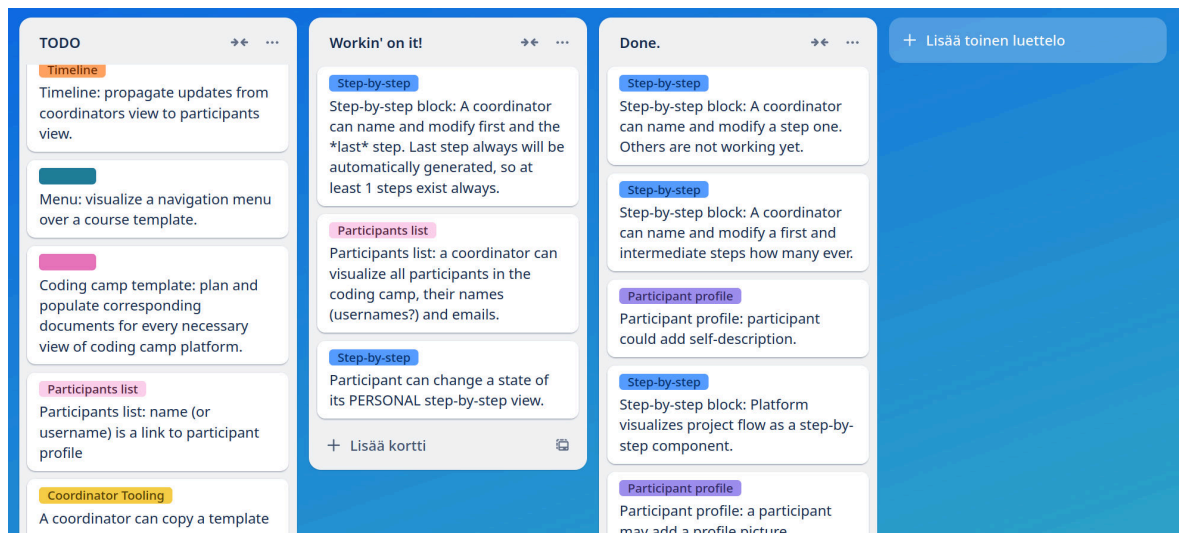
Jotta kurssipohjan rakennetta olisi helpompi hahmottaa, luotiin sen kuvaamisen tueksi korkean abstraktiotason rakennekuva 6. Se esittää hallintadokumentin ja molemmat kojelautanäkymät sekä materiaali- ja tehtävädokumentit. Kuvassa jokaisen näkymän sisään on piirretty erillisinä laatikoina sen sisältämät ominaisuudet. Ominaisuuksien perään on kirjattu, mihin kappaleessa 4.2 esitettyyn vaatimukseen se vastaa. Valikko on piirretty kuvaan omana erillisenä laatikkonaan yksinkertaisuuden vuoksi.

Toteutuksen aloittamiseksi kurssipohjalle asetetut vaatimukset jaettiin osiin. Osat muodostivat tehtäväkokonaisuuksia, joiden etenemistä seurattiin kuvassa 8 esitetyn Kanban-taulun avulla. Luvussa 4.2 esitetyt vaatimukset muotoiltiin taululle tehtäväkortteiksi, joissa vaatimuksen täyttävää ominaisuutta kuvattiin hieman tarkemmin.

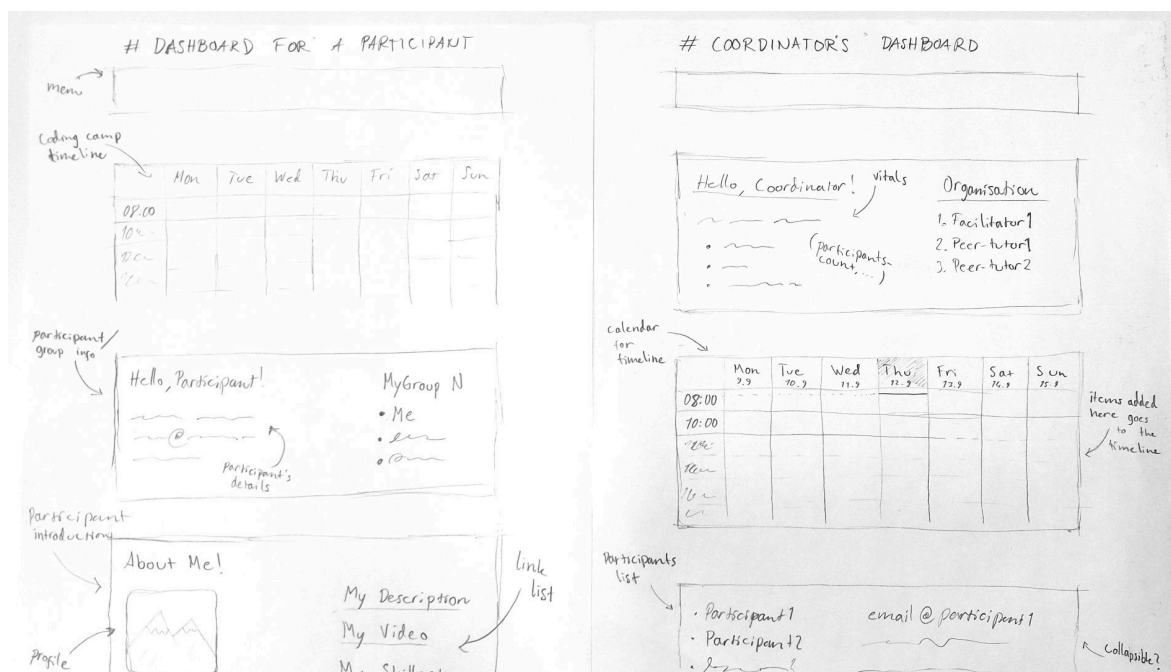
Name	Last modified	Owners	Rights	Id
Go to parent folder				
templates	1 hour ago	op		
camp-exercises	1 hour ago	op		
camp-manager	17 hours ago	op		
camp-material	2 hours ago	op		
coordinator-dashboard	4 minutes ago	op		
participant-dashboard	54 minutes ago	op		

Create a new document Create a new folder

Kuva 7. Kuva esittää kurssipohjan hakemistorakenteen TIM-alustalla. Rakenne voidaan kopioida sellaisenaan uuden koodausleirin pohjaksi.



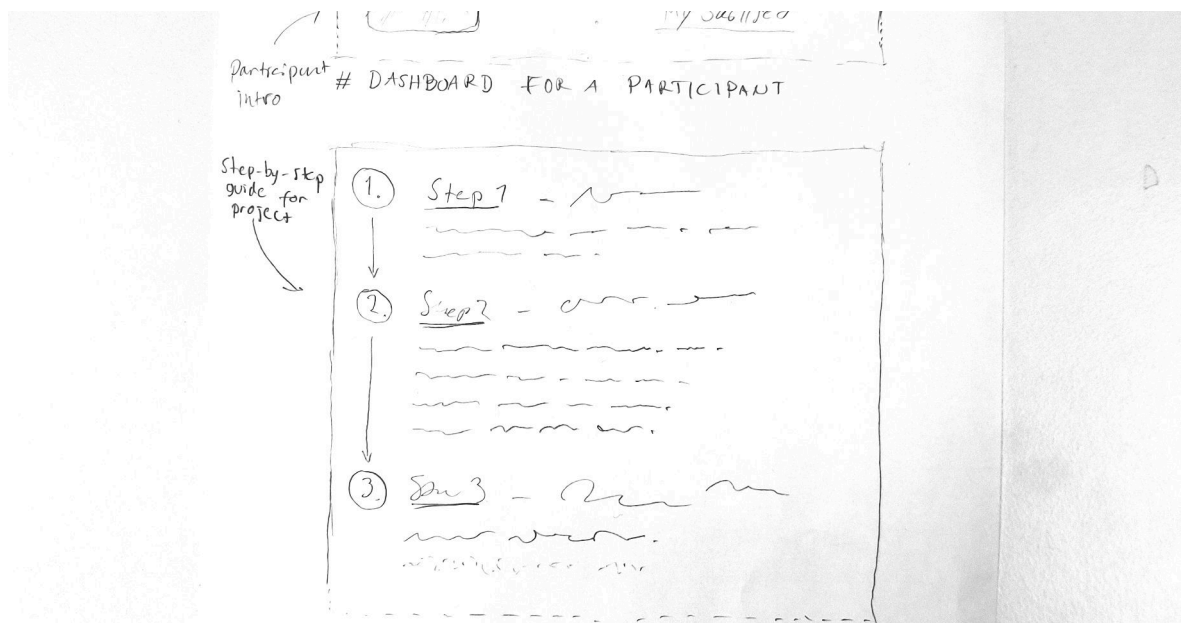
Kuva 8. Prototyypin kehitystä ohjattiin ja seurattiin Kanban-näkymän avulla.



Kuva 9. Luonnosteluvaiheessa havaittiin, että osallistujan ja opettajan kojelaudat voi kehittää pitkälti samanlaisista komponenteista. Ne kuitenkin toimivat käyttäjäroolista riippuen eri tavoin.

Prototyypin kehittämiseksi kurssipohjan rakennetta sekä vaadittujen ominaisuuksien ulkoasua hahmoteltiin luonnostelemalla. Kurssipohjasta luotiin paperiluonnokset, joiden avulla pyrittiin muodostamaan kokonaiskuva kurssipohjaan tulevista kojelautoista ja muista näkymistä. Kojelautojen luonnokset on esitetty kuvassa 9. Luonnosten piirtäminen auttoi myös uusien toiminnallisuuden kehittämisessä. Esimerkiksi kuvassa 10 on esitetty koodausleirin projektintyön seurannan visualisointi.

Suunnittelua ja luonnostelua tehtiin iteratiivisesti paremman tuloksen saavuttamiseksi. Kun kaikista oleellisista näkymistä oli piirretty luonnokset, ne toimitettiin asiantuntijoille nähtäviksi. Piirroksia paranneltiin kommenttien perusteella, jonka jälkeen ne todettiin riittäviksi toteuttamista varten.



Kuva 10. Digitaalisen kurssipohjan näkymät luonnosteltiin aluksi paperille. Kuvassa on visualisoitu koodausleirin projektityön askelten seuranta.

4.3.1 Komponentit yleisesti

Kurssipohjaan luodut dokumentit koostuivat komponenteista, jotka olivat joko alustan tarjoamia tai tutkielman aikana kehitettyjä. Kurssipohjan kojelautanäkymät koostuivat kolmesta samantyyppisestä komponentista, jotka sijoiteltiin allekkain. Opettajan kojelautaan lisättiin vielä neljäs komponentti. Näkymien yläosassa sijaitsi alustan tarjoama *kalenterikomponentti* (V1) koodausleirin tapahtumien seurantaan ja sen alle kehitettiin uutena *profiilikomponentti* (V2) profilitietojen jakamista varten. Alimmaiseksi sijoitettiin uusi *askelkomponentti* (V4) projektityön vaiheiden visualisointiin ja seurantaan kuvan 11 osoittamalla tavalla. Lisäksi opettajan kojelaudalla hyödynnettiin alustan tukemaa *osallistujalista* (V3), joka näytettiin profiilikomponentin ja askelkomponentin välissä, kuten kuvassa 14.

Kojelautanäkymien komponentit toimivat käyttäjäroolista riippuen hieman eri tavoin. Osallistujan kojelaudan komponentit suunniteltiin osittain riippuvaisiksi opettajan kojelaudan näkymiin asetetuista arvoista ja sisällöistä. Esimerkiksi kalenterikomponentin tai askelkomponentin sisältöä oli mahdollista muokata opettajan kojelaudalla. Osallistujan kojelaudalla muokkaaminen oli estetty, mutta siellä oli mahdollista ilmoittautua yksittäiseen aktiviteettiin

tai muuttaa askelkomponentin tilaa.

Muut näkymät sisälsivät hieman vähemmän komponentteja kojelautoihin verrattuna. Materiaalidokumentissa näytettiin opetusmateriaalin lisäksi lohko, joka sisältää *usein kysytyt kysymykset* (UKK, engl. FAQ) (V6). Harjoitustehtävädokumentti kehitettiin tukemaan harjoittelua mm. etätyöskentelyssä (V5, V7). Se koostui alustan tarjoamista *ohjelmointi-* ja *monivalintakomponenteista* sekä *avoimen syötteen komponentista*. Hallintadokumentti sisälsi kurssipohjan uudelleenkäyttöä tukevan *kopiointityökalun* (V8), joka kehitettiin uutena komponenttina tutkielman aikana.

4.3.2 Osallistujan kojelauta

Kalenterikomponentin tehtävänä oli tarjota lineaarinen aikajana koodausleirin toteutusvaiheen aktiviteettien visualisointiin. Osallistujien kojelaudalla oli mahdollista selata aktiviteetteja, tarkastella yksittäistä aktiviteettia tai ilmoittautua mukaan aktiviteettiin.

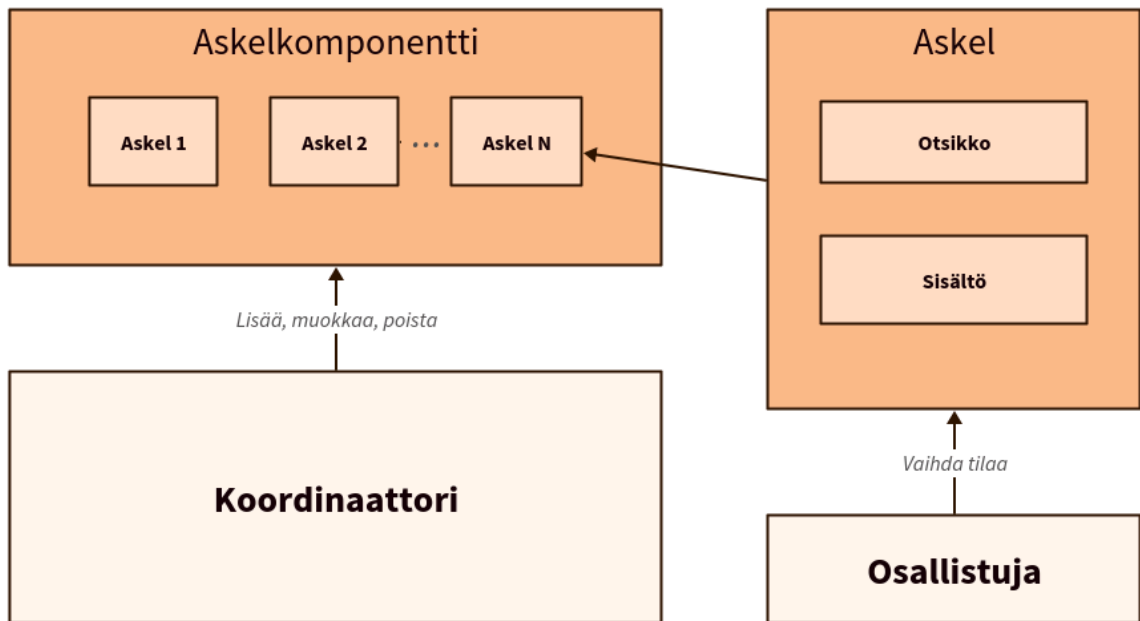
Profiilikomponentin tehtävänä oli mahdollistaa osallistujien profiilikuvien ja henkilökuvauksen lisääminen tukemaan verkostoitumista. Lisäksi tarjottiin mahdollisuus syöttää linkki esim. henkilökohtaisten tietotaitojen tai muun lisätiedon jakamista varten erillisessä dokumentissa tai kolmannen osapuolen sovelluksessa. Profiilikomponentin sisällä suunniteltiin myös näytettäväksi osallistujan projektiryhmän nimi sekä ryhmään kuuluvien jäsenten nimet.

Askelkomponentti kehitettiin osallistujien projektitöiden ohjausta ja seuranta varten. Sen oli myös tarkoitus mahdollistaa opetuksen pelillistäminen koodausleireillä. Komponentti kehitettiin informatiiviseksi ja interaktiiviseksi kuvan 12 esittämällä tavalla siten, että osallistujan oli mahdollista nähdä komponenttiin asetettu sisältö ja vaihtaa askelten tilaa. Tilan vaihtaminen merkitsi askeleen joko odottavaksi tai suoritetuksi, ja näin oli mahdollista seurata projektityön vaiheiden etenemistä kuvan 13 esittämällä tavalla.

The image shows a participant dashboard interface with three main components highlighted by brackets:

- Kalenterikomponentti (Calendar component):** A calendar view for Week 48 / 2024, showing a grid of days from Monday (25.11) to Sunday (1.12) and times from 08:00 to 16:00. Various colored blocks represent scheduled activities.
- Profiilikomponentti (Profile component):** A user profile card for 'Test user 1!'. It includes a profile picture, a bio ('Hey, I am happy participant!'), a video link ('My video: www.vimeo.com'), and a 'Group number' section listing 'member1' and 'member2'. There is also a 'Modify profile' button.
- Askelkomponentti (Step-by-step component):** A 'Step-by-step' guide with five numbered steps:
 - 1 Project Idea:** In this step, brainstorm ideas for the project. Identify the problem you aim to solve, the target audience, and the core objectives. Collaborate with the team to generate creative and innovative concepts. Prioritize ideas based on feasibility, impact, and alignment with goals.
 - 2 Design plan:** Outline a comprehensive design plan that includes user requirements, technical specifications, and a roadmap. Create sketches, wireframes, or diagrams to visually represent the solution. Ensure the plan addresses user experience, scalability, and functionality.
 - 3 Prototype tooling:** Select the tools and technologies necessary to build a prototype. This may include programming languages, development frameworks, hardware components, or design software. Evaluate the tools for compatibility with your project goals and team expertise.
 - 4 Prototype development:** Begin building the prototype based on the design plan. Focus on implementing the core features and functionality. Ensure that the prototype serves as a functional representation of the final product, even if it lacks full polish or features.
 - 5 Fine tuning:** Refine and optimize the prototype based on testing and feedback. Address any issues, enhance performance, and improve usability. Collaborate with stakeholders to ensure the solution aligns with their expectations and needs.
 The final step is marked with a green checkmark and labeled 'Presentation!'.

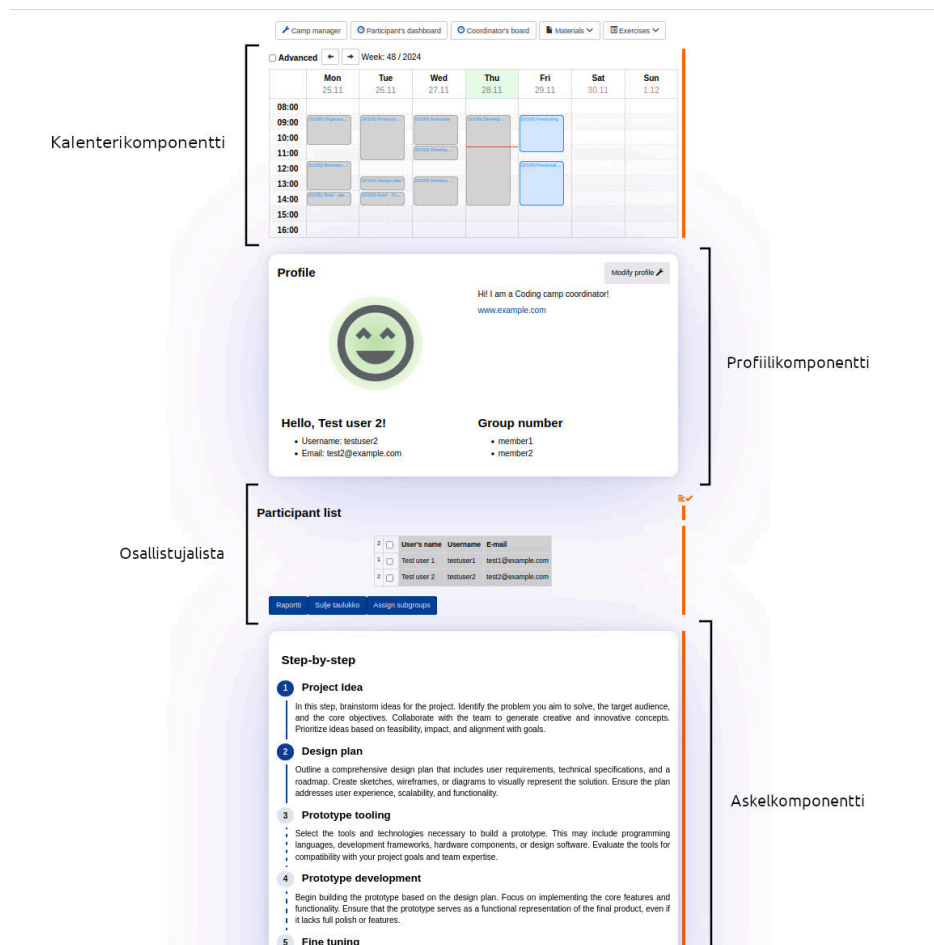
Kuva 11. Osallistujan kojelauta sisältää kolme komponenttia, joiden avulla osallistuja voi seurata kurssin etenemistä.



Kuva 12. Askelkomponentin avulla opettaja voi visualisoida esim. projektin vaiheet. Koo-
 dausleirin osallistuja voi vaihtaa askelen tilaa, jonka perusteella opettaja voi seurata pro-
 jektin etenemistä.



Kuva 13. Osallistujan käyttöliittymä sisältää interaktiivisen askelkomponentin, jonka avulla
 osallistuja voi seurata esim. projektin vaiheiden edistymistä.



Kuva 14. Opettajan ja osallistujan kojelaudat muistuttavat toisiaan, mutta opettajan kojelaudalla on yhteensä neljä komponenttia. Osallistujalistaus on tarkoitettu vain opettajan nähtäville.

4.3.3 Opettajan kojelauta

Kalenterikomponentin täysi muokkaaminen onnistui opettajan kojelaudalta. Sieltä pystyi mm. asettamaan kalenteriin katselijoiksi oikean osallistujaryhmän, lisäämään ja muokkaamaan aktiviteetteja sekä tarkastelemaan ilmoittautumisia.

Profiilikomponentti mahdollisti samat toiminnot opettajille kuin osallistujillekin. Projektiryhmän nimen tilalla opettajan profiilikomponentissa suunniteltiin näytettäväksi organisaatioryhmiä kuten vertaisohjaajat sekä ryhmiin kuuluvien nimet ja yhteystiedot.

Askelkomponentin sisällön tuottaminen oli opettajan toiminnallisuus. Opettajan kojelaudalta

komponenttiin oli mahdollista lisätä askelia otsikoineen ja kuvauksineen *YAML*¹-syntaksia käyttäen. Komponenttiin suunniteltiin myös visualisointi siten, että opettajan olisi mahdollista omalta kojelaudaltaan nähdä eri projektiryhmien eteneminen.

Osallistujalistaus esitti kaikki koodausleirin osallistujat suoraan opettajan kojelaudalla. Se konfiguroitiin listaamaan koodausleirin osallistujien nimet, käyttäjätunnukset ja yhteystiedot taulukkoon, josta tiedot pystyi tarvittaessa viemään esim. *CSV*²-muodossa taulukkolaskentaohjelmaan.

4.3.4 Materiaalidokumentti

Kuvassa 15 esitetyn materiaalidokumentin tarkoituksena oli olla yksinkertainen pohja koodausleirin oppimateriaalille. Toteuttamisen jälkeen se oli käytännössä tynkä, johon opettajan oli mahdollista itse tuottaa sisältöä hyödyntämällä alustan tukemia formaatteja kuten *markdown*³-tekstiä, kuvia, videoita ja vaikkapa *kyselyitä* tai muita alustan tarjoamia interaktiivisia ominaisuuksia.

Muun materiaalin lisäksi dokumenttiin lisättiin *UKK*-lohko formaalin vuorovaikutuksen lisäämiseksi opettajan, ohjaajien ja osallistujien välillä. *UKK*-lohko sisälsi listan kysymys ja vastaus pareja ja se toteutettiin alustan tarjoamalla piilotustoiminnolla. Lohkon ollessa piilotettuna, siitä jäi näkyviin otsikko (*FAQ – Frequently asked questions*) ja avauspainike.

4.3.5 Harjoitustehtävädokumentti

Harjoitustehtävädokumentti sisälsi kolme komponenttia kuvan 16 osoittamalla tavalla. *Ohjelmointitehtäväkomponentti* lisättiin palvelemaan osallistujien harjaantumista koodausleirin tehtävissä. Se konfiguroitiin tehtävädokumenttiin yksinkertaiseksi *C#*-ohjelmointikielen tehtäväpohjaksi, josta opettaja voi kopioida uusia vastaavia ja muokata niitä toteutukseen sopiviksi. Komponentin avulla osallistuja voi ajaa itse kirjoittamaansa koodia ja nähdä miten se toimii. Opettaja voi asettaa koodin tarkistimeen palautteen, joka voidaan koodin ajamisen

1. <https://yaml.org/>

2. <https://fi.wikipedia.org/wiki/CSV>

3. <https://www.markdownguide.org/>

Coding camp materials

This material may contain text, images, videos etc. like smaller interactive exercises.



FAQ - Frequently asked questions

FAQ

1. Where can I find Coding camp exercises?

Top right in the menu, there is dropdown list of exercises.

Add paragraph

Kuva 15. *Materiaalidokumentti sisältää myös usein kysytyt kysymykset.*

jälkeen näyttää osallistujalle.

Monivalintakomponentti voi toimia joko koodausleirin tehtävänä tai esim. kyselylomakkeen osana. Tehtävädokumenttiin konfiguroitiin yksi monivalintakomponentti mallipohjaksi.

Avoimen syötteen komponentti voi monivalintakomponentin tavoin toimia tehtävänä, kyselylomakkeen osana tai muun avoimen syötteen lähetyspaikkana. Tehtävädokumenttia laajennettiin vielä tämän komponentin mallikonfiguraatiolla.

4.3.6 Hallintadokumentti ja navigointi

Hallintadokumenttiin sisällytettiin kuvan 18 mukaisesti kurssipohjan *kopiointityökalu* tehostamaan koodausleirin perustamista. Lisäksi hallintadokumenttiin luotiin tehtävälista, joka kannattaa suorittaa kurssipohjan kopioinnin yhteydessä. Sillä varmistetaan kurssipohjan oikea toiminta koodausleirillä.

Kurssipohjaan lisättiin myös *navigointi*, joka TIM-alustan arkkitehtuurin mukaisesti konfiguroidaan erilliseen *templates*-dokumenttihakemistoon (ks. kuva 7). Sieltä se periytyy kaikkiin kurssipohjan dokumentteihin.

Kopiointityökalu kehitettiin tutkielman puitteissa ainoaksi kurssipohjan aputyökaluksi. Käy-

Camp manager
Participant's dashboard
Coordinator's board
Materials
Exercises

Coding camp exercises

This is a document for exercises, where participant could do hands-on interactive exercises. This document may be duplicated how many different exercises documents is needed/feasible on the camp.

**Ohjelmointi-
tehtävä**

Programming Exercise (csPlugin)

Instructions are written in here. E.g.

Make the program print following:

Hello, World!
I am a developer!

Short instructions could go here. Longer explanations may be given in separate paragraph/block above.

```
System.Console.WriteLine("Hello World");
```

Aja
Näytä koko koodi
Highlight
Copy

**Monivalinta-
tehtävä**

Avoin tehtävä

Multiple choice exercise

It is possible to modify the exercise plugin by clicking the questionmark (?) button down below. Also, the plugin's YAML markup can be modified in edit view.

The title

Description of a choice

An another choice

Save
?

Avoin tehtävä

Avoin tehtävä

Text block

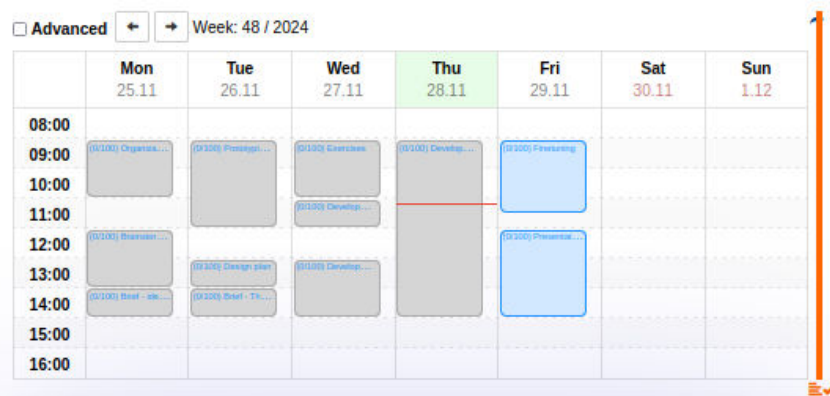
Give your ideas for coding camp projectwork.

Give your answer here

Save
Copy
Wrap
?

Avoin tehtävä

Kuva 16. Tehtävädokumentti sisältää konfiguraatiopohjat erityyppisille harjoitustehtäville.



Kuva 17. Kalenteri visualisoi koodausleirin aikataulun.

tännössä se mahdollisti kurssipohjan sisältämien dokumenttien kopioimisen alustalla uuden koodausleirin pohjaksi. Hallintadokumentin ja sen sisältämän kopiointityökalun käyttöoikeudet tarkoitettiin annettavaksi vain koodausleirin järjestäjälle.

4.4 Tuloksen arviointi

Prototyypille asetettiin yhteensä 11 vaatimusta. Kahdeksaan vaatimukseen vastaamalla tuotettiin viidestä TIM-dokumentista koostuva prototyyppi, joka tukisi koodausleirin osallistujia ja opettajia koodausleirin eri vaiheissa.

Prototyypin arviointi suoritettiin kuvan 2 mukaisesti kriteeriperusteisena arviointina. Arviointiasteikkona on kolmiportainen asteikko, jossa vaatimukseen verrattu ominaisuus joko toteuttaa tai ei toteuta vaatimusta tai toteuttaa osittain. Prototyypin arviointikriteerit ovat luvussa 4.2 listattujen vaatimusten kahdeksan ensimmäistä kohtaa.

V1 *Aikajanan näyttäminen.* **Toteutuu.** Aikajana toteutettiin käyttämällä alustan tarjoamaa kalenterikomponenttia ja konfiguroimalla se koodausleirin tarpeisiin. Kalenteri näyttää tulevat tehtävät ja niiden aikataulun tarkoituksenmukaisella tavalla. Se ei kuitenkaan muistuta visuaaliselta ulkoasultaan aikajanaa (ks. kuva 17). Komponenttia olisi mahdollista jatkokehittää muuttamalla ulkoasua paremmin vaatimuksia vastaavaksi.

V2 *Osallistujien tietotaitojen jakaminen.* **Toteutuu.** Profilissa on mahdollista esittää esim. osallistujan taitamat ohjelmointikielet listaamalla ne henkilökuvauksen yhteyteen. Pro-

filiin on myös mahdollista lisätä linkki johonkin kolmannen osapuolen sovellukseen, jonka avulla taitoja voi visualisoida eri tavoin. Komponenttia olisi luontevaa jatkokehittää siten, että henkilökohtaisille taidoille olisi erillinen visualisointi profiilikomponentin sisällä.

- V3 *Työkalut osallistujatietojen keräämiseen.* **Toteutuu.** Osallistujatietojen kerääminen tapahtuu käytännössä alustalle rekisteröitymisen yhteydessä. Rekisteröitymisen lisäksi osallistuja lisätään koodausleirin käyttäjäryhmään ja kurssipohjan tarjoaman osallistujalistauksen avulla on mahdollista näyttää ryhmän jäsenet sekä viedä käyttäjäryhmän tiedot CSV-muodossa esim. taulukkolaskentaohjelmaan käsiteltäväksi. Jatkoa ajatellen on syytä selvittää tarkemmin, mitä tietoja osallistujista halutaan kerätä ja sen jälkeen listaus voidaan konfiguroida vastaamaan paremmin vaatimuksia.
- V4 *Projektityön suunnittelu ja kehittäminen askel askeleelta.* **Toteutuu osittain.** Askelkomponentin avulla osallistujat voivat omassa kojelautanäkymässään tarkastella esim. projektityön sisältämiä vaiheita tai muuta siihen luotua vaiheistusta. Komponentin tilaa muuttamalla he voivat seurata omaa etenemistään vaiheissa. Komponentti ei vielä tarjoa mahdollisuutta seurata projektiryhmän etenemistä, joten vaatimus ei tältä osin täyty. Alustan käyttäjäryhmiin liittyviä osia tulisi kehittää siihen suuntaan, että ryhmän työvaiheiden seurantaominaisuus voitaisiin toteuttaa.
- V5 *Etätyökalujen tarjoaminen.* **Toteutuu.** Kurssipohjaan lisätyt materiaali- ja harjoituskäsitteet tukevat koodausleiriä esim. hybridiopetuksessa. Alusta ylipäättään tarjoaa riittävät etätyökalut, mutta kurssipohjaa voisi jatkokehittää tarjoamaan osallistujille vielä valmiit työskentelydokumenttipohjat. Luonnollisesti osallistujien on mahdollista luoda itse omat työskentelydokumenttinsa.
- V6 *Yhteydenpidon mahdollistaminen järjestäjän ja osallistujien välillä.* **Ei toteudu.** UKK-palsta mahdollistaa yksisuuntaisen yhteydenpidon opettajalta koodausleirin osallistujille. Se ei kuitenkaan täytä vaatimusta pikaviestimestä tai blogialueesta. Vaatimuksen voisi järkevästi täyttää esim. konfiguroimalla kurssipohjaan keskustelupalstan tai integroimalla alustaan jonkin viestisovelluksen. Myös alustan tarjoamat sähköpostitoiminnot voisivat olla varteenotettava vaihtoehto vaatimuksen täyttämiseksi.
- V7 *Ennakkotehtävien tuottaminen.* **Toteutuu.** Ennakkotehtäviä on mahdollista tuottaa koodausleirin esivaiheessa heti sen jälkeen, kun kurssipohja on kopioitu uuden koodauslei-

Coding camp manager

Starting a new camp

1. Copy a template for a camp
2. Create a group for the camp, e.g. oscar25
3. Create users for each participants
4. Add all camp participants/users to the group
5. Change calendar bookers group to be current camp's group
6. Update Steps-component contents according your needs.
7. Copy the calendar reference to Participant's dashboard.
8. Similarly, copy the Steps-component reference to Participant's dashboard.



Copy tool

Copy a camp or other course from a template.

Copy from path, eg. courses/camp

Name for a copy

Create ↗

Kuva 18. Hallintadokumentissa on mahdollista perustaa uuden koodausleirin pohja kopiointityökalulla. Dokumenttien yläosaan lisättiin myös valikko navigointia varten.

rin pohjaksi. Kurssipohja sisältää harjoitustehtävädokumentin mallitehtävineen. Tehtävien tuottamista voisi helpottaa luomalla tehtäväpankin.

V8 *Kurssipohjan uudelleenkäyttäminen.* **Toteutuu.** Hallintadokumenttiin kehitetty kopiointityökalu mahdollistaa kurssipohjan uudelleenkäyttämisen. Jotta työkalusta saisi enemmän irti, kopioinnin jälkeisiä rutiininomaisia toimia voisi olla järkevää tehostaa. Esimerkiksi koodausleirin käyttäjäryhmän luominen ja kalenterin sekä osallistujalistauksen konfiguroiminen käyttäjäryhmälle kannattaisi automatisoida. Myös muita mm. kuvassa 18 näkyviä tehtäviä olisi järkevää automatisoida.

Kurssipohjaan kehitetyt ominaisuudet vastasivat pääosin niille asetettuja vaatimuksia. Osa ominaisuuksista vaatii kuitenkin jatkokehitystä vaatimusten täyttämiseksi. Lisäksi muita toteutettuja ominaisuuksia voi jatkokehittää laadukkaamman lopputuloksen saavuttamiseksi.

5 Pohdinta

"The expert at anything was once a beginner."

– Helen Hayes

Pohdintaluku käsittelee kurssipohjan ominaisuuksien toteutumista kappaleessa 5.1, vaatimusmäärittelyn onnistumista kappaleessa 5.2 ja prototyypin jatkokehittämistä kappaleessa 5.3. Tulosten luotettavuutta on käsitelty kappaleessa 5.4. Lisäksi kappale 5.5 tarjoaa jatkotutkimusideoita tutkielman aihepiiristä.

Tutkimuskysymyksemme on:

Miten voitaisiin suunnitella ja toteuttaa digitaalinen kurssipohja koodausleirien käyttöön?

Tutkimuskysymykseen vastattiin suunnittelemalla ja toteuttamalla koodausleiritoimintaa tukeva kurssipohja Jyväskylän yliopistossa kehitetyn TIM-alustan päälle (ks. luku 4). Ratkaisussa hyödynnettiin alustan valmiita ominaisuuksia, mutta samalla kehitettiin myös koodausleirin tarpeisiin räätälöityjä uusia ominaisuuksia. Kehitetty kurssipohja vastasi melko hyvin sille asetettuja vaatimuksia tarjoten tukea koodausleireille luvussa 2.2 esitettyihin vaiheisiin.

Kurssipohjan toteuttamiseksi olisi voinut olla myös muita vaihtoehtoja. Nyt se kehitettiin valmiin oppimisalustan päälle, mikä toisaalta nopeutti kehittämistä rajaten kuitenkin samalla toimintaympäristöä mm. arkkitehtuurin ja käyttöliittymien osalta. Lisäksi alustan käytettävyyttä ei ole joitain komponentteja lukuunottamatta kattavasti tutkittu ja nykyiset kehityskäytännöt vaikuttivat perustuvan enemmän kehittäjän kuin käyttäjän näkökulmaan. Alustalle on kuitenkin koottu runsaasti kehitysehdotuksia¹, joista osa koskee käytettävyyttä.

Toisaalta alustaksi olisi voitu valita jokin muu tunnettu ja laajemmin testattu oppimisalusta, joita kirjallisuuden perusteella on tarjolla todella paljon. Koska työ tehtiin Jyväskylän yliopistossa, oli luonnollista valita samassa yliopistossa kehitetty LMS-alusta. Kurssipohja olisi voitu toteuttaa myös kokonaan uuteen oppimisalustaan eri teknologioilla. Tämä olisi tuo-

1. <https://tim.jyu.fi/view/tim/kehitysehdotuksia/parannusehdotuksia>

nut huomattavia vapauksia arkkitehtuurin suunnitteluun ja käyttöliittymien kehittämiseen. Toisaalta työmäärä olisi kasvanut merkittävästi. Myös tutkielman kontekstissa olisi todennäköisesti päädytty rajaamaan toteutusta eri tavalla. Kokonaisuutta ajatellen kurssipohjan kehittäminen TIM-alustan päälle oli toimiva ratkaisu.

5.1 Ominaisuuksien hyödyllisyys

Kurssipohjaan kehitetyt ominaisuudet tukevat koodausleirien toimintaa monella tavalla. Koodausleirin eri vaiheissa tapahtuvan toiminnan tukemisen lisäksi ominaisuudet myös tukevat mm. osallistujan taitojen kehittymistä. Kappaleen 2 kirjallisuudesta havaitaan, että osallistuja voi kehittää sekä pehmeitä että kovia taitojaan osallistumalla koodausleirille. Nyt tutkielman aikana kehitetty kurssipohja tukee profiilitoimintojen avulla erityisesti ryhmätyöskentely-, projekti- ja yhteistyötaitojen kehittymistä. Lisäksi ohjelmointitaidot kehittyvät harjoittelemalla koodausleirin tehtäviä, joiden tuottamista kurssipohja tukee.

Koodausleirin *esivaiheessa* (ks. kappale 2.2.1) kurssipohja tukee *opettajan* työskentelyä tarjoamalla kurssipohjan kopiointityökalun uuden koodausleiripohjan luomiseksi ja sisällöntuotannon aloittamiseksi. Sisällöntuotantoa tukevat myös kurssipohjan sisältämät materiaali- ja harjoitustehtävädokumentit. Kojelautanäkymän avulla opettaja voi luoda askelkomponentin sisällön valmiiksi ennen leirin alkua ja osallistujalistauksen avulla esim. projektiryhmät voidaan muodostaa valmiiksi toteutusvaihetta varten. Harjoitustehtävädokumentti mahdollistaa *osallistujalle* ennakkotehtävien tekemisen. Lisäksi kojelautanäkymän kautta osallistuja voi täydentää profiiliaan jo ennen koodausleirin toteutusvaiheen alkua.

Toteutusvaihe (ks. kappale 2.2.2) on hektinen vaihe ja kurssipohja tukee sen läpivientiä. Kojelautanäkymä tarjoaa *järjestäjälle* yleiskatsauksen koodausleirin osallistujien yhteystiedoista ja projektityön etenemisestä sekä mahdollisuuden aktiviteettien visualisointiin ja hallintaan kalenterin avulla. Jos järjestäjällä ei ole aiempaa kokemusta aktiviteettien ajoittamisesta koodausleirille, kurssipohjan kalenteriin määritettyä malliajoitusta voi käyttää apuna. Lisäksi järjestäjä tai koodausleirin ohjaajat voivat ylläpitää UKK-palstaa ja näin parantaa kommunikaatiota sekä toteutusvaiheen aikaista tukea. Profiilikomponentti auttaa *osallistujia* verkostoitumaan projektiryhmänsä kanssa. Materiaali- ja harjoitusedokumentit tarjoavat osal-

listujille mahdollisuuden perehtyä sisältöihin ja harjoitella projektityötä lähi-, etä- ja hybridiopetuksessa. Lisäksi UKK-palsta tarjoaa heille välittömän tuen koodausleiriin liittyvissä kysymyksissä. Toteutusvaiheen aikana osallistujat voivat myös luoda alustalle omia dokumenttejaan esim. projektityön esittelyä varten.

Koodausleirin aktiviteetit tarjoavat *jälkivaiheeseen* (ks. kappale 2.2.3) dataa *järjestäjälle* analysoitavaksi. Harjoitusdokumenttien sisältämät tehtäväkomponentit tarjoavat mahdollisuuden tarkastella osallistujien suoriutumista. *Osallistujat* voivat hyödyntää materiaali- ja harjoitusdokumentteja vielä jälkivaiheessakin koodausleirillä harjoiteltujen aiheiden kertamiseen. Lisäksi alusta tarjoaa heille mahdollisuuden palata projektityönsä pariin vielä toteutusvaiheen jälkeen.

Kurssipohjalle asetetuista 11 vaatimuksesta vastattiin kahdeksaan. Toteutus täyttää vaatimukset pääosin, mutta toteutettuja ominaisuuksia jatkokehittämällä saadaan kurssipohja tukemaan koodausleirejä nykyistä paremmin. Mm. järjestäjän työtä on mahdollista vielä tehostaa automaation avulla. Kurssipohjan kopiointityökalu automatisoi jo nyt kokonaisen hakemiston rakenteen ja sisällön kopioinnin. Arvioinnin perusteella se vaikuttaa olleen oikeansuuntainen ratkaisu. Käytännössä kopiointityökalu säästää järjestäjältä yksittäisten dokumenttien kopioimisen vaivan, mikä mahdollistaa kurssipohjan sisältämien dokumenttien määrän huomattavankin nostamisen. Toisaalta järjestäjän vastuulle jää yhä ryhmien luominen manuaalisesti, ryhmien konfigurointi kalenterille ja osallistujalistalle sekä aikataulun ym. suunnittelu ja luonti. Kopiointityökalun päivittämisen lisäksi osallistujien rekisteröityminen alustalle kannattaisi tehdä mahdollisimman helpoksi, jotta itse alustan käytön aloittaminen olisi sujuvaa. Myös profiilikomponentin toteutusta voisi parantaa, sillä mm. tietotaitojen visualisointi tekstinä henkilökuvausten yhteydessä ei välttämättä ole tarkoituksenmukaista.

Toteutuksen jatkokehittämiseksi esitämme joitain ehdotuksia. Kaikki rutiinitoimenpiteet voisi yrittää automatisoida siten, että kurssipohjan kopioimisen jälkeen konfigurointi pystyttäisiin hoitamaan ohjelmallisesti koodausleirin järjestäjän antamalla parametreilla. Tämä karsisi manuaalisesti suoritettavien toimenpiteiden määrää ja järjestäjä pääsisi nopeammin oleellimpien tehtävien kuten aikataulun suunnittelun sekä ryhmien muodostamisen pariin. Rekisteröityminen alustalle voisi onnistua erillisen aloituskyselydokumentin avulla, jonka pa-

rametrien perusteella osallistujien käyttäjät luotaisiin ja samalla voitaisiin täydentää profilia kyselydokumentin tietojen perusteella. Käyttäjien luomisen voisi antaa myös opettajan tehtäväksi mahdollistaa sen esim. CSV-datan tuontina taulukkolaskentaohjelmasta tai erillisestä kyselylomakkeesta. Vielä profiilikomponentin osalta tietojen visualisointia voisi parantaa siten, että pelkän tekstikuvauksen sijaan osallistujan tietotaidot näytettäisiin profiilissa sanapilven (engl. tag cloud), ikonien tms. avulla.

5.2 Vaatimusmäärittelyn analysointi

Vaikka suurin osa vaatimuksista täytettiin, *yhteydenpidon mahdollistaminen (V6)* ja *projektityön toteutus askel askeleelta (V4)* eivät täytyneet. Yhteydenpidon osalta kysymys voi olla siitä tulkinnasta, miten yhteydenpito täytyisi toteuttaa. TIM-alusta tukee sellaisenaan yhteydenpitoa mm. sähköpostilla. Lisäksi yhteydenpitoon voisi riittää opettajan luoma dokumentti, johon kaikki koodausleirin osallistujat voisivat luoda tekstiä. Näin saataisiin melko vaivattomasti perustettua toimiva keskustelualusta. Sähköposti-ilmoitusten tilaaminen tosin vaatisi lisäkonfigurointia alustalla. Askelkomponentilta puolestaan vaadittiin selvästi ryhmän projektin etenemisen seuranta, jota ei tutkielman puitteissa saatu toteutettua. Askelkomponentin toteutus on hieman kontroversiaali, sillä muihin ominaisuuksiin verrattuna siihen käytettiin huomattava määrä työaikaa eikä se siitä huolimatta täyttänyt vaatimuksia. Se on kuitenkin riippuvainen alustan ryhmätoiminnoista. Koska koodausleirin projektiryhmien organisointi alustalla ei toiminut askelkomponentin vaatimalla tavalla, ei ryhmätyön seuranta-toiminnallisuuksia voitu toteuttaa valmiiksi.

Kohteen tunteminen on tärkeää vaatimusmäärittelyn ja ohjelmiston kehittämisen aikana. Vaatimusmäärittelyyn ja ohjelmistokehitykseen osallistuneilla kehittäjillä ja sidosryhmällä ei ollut yhtenevää käsitystä TIM-alustasta, mikä aiheutti aika ajoin väärinkäsityksiä kehitettävien ominaisuuksien kohdalla. Näin ollen kysymys esimerkiksi toteutettavan yhteydenpidon riittävydestä olisi voinut ratketa sillä, että ennen kurssipohjan kehittämisen aloitusta sidosryhmällä olisi ollut parempi käsitys alustan toiminnasta ylipäätään ja esim. alustan sisältämistä yhteydenpitovaihtoehdoista.

Kokemuksemme perusteella suosittelimme vastaavanlaisiin hankkeisiin perusteellisempaa

kohdealueeseen tutustumista vaatimusmäärittelyn tueksi. Koodausleirien järjestäjille voisimme suositella jonkin muun kurssin järjestämistä kehitettävällä oppimisalustalla, jolloin vaatimusmäärittelyyn tultaessa heillä olisi kokemusta alustan käytöstä. Tällöin vaatimusten esittäminen voisi olla helpompaa ja ne olisivat todennäköisesti tarkempia. Toisaalta alustan kehittäjille suosittelemme koodausleiritoimintaan tutustumista oikeassa ympäristössä. Se voisi olla hedelmällistä, sillä koodausleirin konseptiesittelyn lisäksi esim. koodausleirin observointi parantaisi käsitystä sen toteuttamisesta käytännössä ja voisi helpottaa alustalta vaadittujen ominaisuuksien kehittämistä.

Kurssipohjan prototyyppi loi kuitenkin toimintaedellytykset Fronza ym. 2024 esittämälle uudistetulle koodausleirille. Varsinainen ehdotelman mukainen koodausleiri vaatii kuitenkin kaikkien luvussa 4.2 esitettyjen vaatimusten täyttämisen. Prototyyppiä jatkokehittämällä *pienimmäksi toimivaksi tuotteeksi* kaaviossa 2 kuvattua prosessia noudattaen voitaisiin saavuttaa tulevaisuuden koodausleirejä paremmin tukeva ratkaisu.

5.3 Pienin toimiva tuote

Ennen kuin kurssipohjaa kokeillaan varsinaisella koodausleirillä se kannattaa kehittää pienimmäksi toimivaksi tuotteeksi. Se olisi nimensä mukaisesti tuote eli tässä tapauksessa kurssipohja, joka sisältäisi pienimmän mahdollisen määrän ominaisuuksia koodausleirin järjestämiseksi alusta loppuun – tapahtuman mainonnasta suoritusten arviointiin ja loppuraportin tulostamiseen saakka.

Tutkielman rajoituksista johtuen pienimmän toimivan tuotteen kehittäminen ei ollut mahdollista niin, että kaikki luvussa 4.2 mainitut vaatimukset olisi täytetty. Toisaalta kuten Fronza ym. 2024 esittivät, toteutuksen riittäisi täyttää vielä puuttuvat vaatimukset *projektityön julkistamisesta* (V10), *loppuraportin generoinnista* (V11) ja *koodausleirillä hankittujen taitojen visualisoinnista* (V9). Lisäksi edellä kappaleessa 5.2 mainitut V6 ja V4 täytyisi kehittää valmiiksi. TIM-alusta saattaisi jo nyt mahdollistaa koodausleirien sidosryhmäintegraatioon vaikuttavien vaatimusten täyttämisen, sillä yksinkertaisimmillaan esim. projektityöt voisi julkistaa kurssipohjaan luodussa dokumentissa. Samoin kaikista alustan dokumenteista on mahdollista tulostaa asiakirjoja. Laadukkaan tulostuksen tekeminen vaatisi kuitenkin,

että alustaan ohjelmoitaisiin koodausleirin asiakirjoihin sopivat tulostusasetukset haluttujen tietojen tulostamiseksi. Alusta tukee myös suoritettujen tehtävien pisteyttämistä ja arviointia tehtäväpalautusten perusteella. Koodausleirin edetessä osallistujien käyttäjäprofiiliin voisi pelillistämisen hengessä visualisoida jonkin virtuaalisen suoritusmerkin tai mitalin palkinnoksi esim. oleellisten tehtäväkokonaisuuksien suorittamisesta tai tietyn pistemäärän saavuttamisesta.

Koodausleirin järjestämiseksi alustalta siis vaadittaisiin, että kaikki vaatimukset täyttyvät. Edelleenkin sen ei välttämättä tarvitsisi sisältää esim. kaikkia iteraatioiden aikana suunniteltuja ominaisuuksia, vaan minimissään riittäisi täyttää kappaleessa 4.2 esitetyt vaatimukset täyttävät ominaisuudet. Niiden turvin olisi mielekästä lähteä toteuttamaan kurssipohjan käyttäjäkokeilua koodausleirillä.

5.4 Tulosten luotettavuus

Suunnittelutieteellinen tutkimusmenetelmä tarjosi luontevan lähestymistavan kurssipohjan kehittämiseksi. Piirteilleen ominaisesti se toimi tutkielmassa ongelmanratkaisuparadigmana koodausleirin eri vaiheiden haasteiden ratkaisemiseksi. Innovatiivista ratkaisua haettaessa se oli oikea valinta tutkimusmenetelmäksi. Tutkielmassa olisi voitu soveltaa menetelmää myös vaihtoehtoisella tavalla esim. lisäämällä nykyisen prosessin ympärille relevanssisykli ja täsmennyssykli. Nyt tutkielma sisältää yhden suoraviivaisen suunnittelusta ja toteutuksesta koostuvan prosessin.

Tässä tutkielmassa menetelmä tarjosi vaatimusmäärittelystä, suunnittelusta ja toteutuksesta sekä arvioinnista koostuvan iteratiivisen prosessin, joka sisälsi paljon kehitystä tukevia toimintoja. Lähtien vaatimusmäärittelystä, prosessi sisälsi fokusryhmäkeskustelun ja vaatimusten muotoilemisen sekä ohjelmistokehityksen alkaessa suunnittelupalaverin, jossa vaatimuksia tarkennettiin ja niistä muodostettiin tehtäviä Kanban-tauluun. Vaatimusmäärittelyä seurannut suunnittelu ja toteutus etenivät iteratiivisesti kurssipohjan rakenteen ja käyttöliittymien kehittämiseksi. Suunnittelu tehtiin ensin paperille luonnostelemalla ja jatkettiin edelleen toteutusvaiheessa katselmoimalla ja kommentoimalla kurssipohjan kehitystyön edistymistä. Toteutuksessa oppimisalustaan ohjelmoitiin uudet vaaditut ominaisuudet ja kurssipoh-

ja luotiin alustalle vaatimusten perusteella.

Lopuksi laadittiin vielä kriteeriperusteinen arviointi kurssipohjan vaatimuksiin perustuen. Vaatimuksiin pohjalta tehty arviointi antoi viitteitä siitä, kuinka riittävällä tasolla kurssipohja on yksittäisten ominaisuuksien ja laajemman kokonaisuuden osalta. Yksittäisten ominaisuuksien osalta arvioinnin tulos auttaa jatkokehityksen priorisoinnissa. Laajemman kokonaisuuden näkökulmasta arviointi auttaa hahmottamaan kurssipohjan tilaa suhteessa pienimpään toimivaan tuotteeseen.

Kurssipohjaa ei tutkielman aikana saatu tuotantokäyttöön, mutta sitä testattiin jatkuvasti manuaalisesti ominaisuuksien kehittämisen yhteydessä. Tulevaisuudessa uusista ominaisuuksista saadaan toivottavasti nopeammin käyttäjäpalautetta, kun kurssipohja kehitetään valmiiksi ja otetaan käyttöön koodausleireillä. Jatkokehitystä ajatellen suosittelemme vaatimusten muotoilemista tarkemmiksi, sillä liian korkean abstraktiotason toiminnalliset vaatimukset voivat aiheuttaa väärinkäsityksiä ominaisuuksia toteutettaessa. Tutkielman kontekstissa vaatimukset olivat kuitenkin riittävän hyvin muotoiltuja ja ne toimivat kattavasti myös artefaktin kriteeriperusteisessa arvioinnissa.

5.5 Jatkotutkimus

Vaikka kurssipohja kehitettiin nimenomaan koodausleiritoiminnan tueksi, sille voi tulevaisuudessa olla käyttöä myös muilla kursseilla. TIM-alustalla järjestetään jo nyt useita erilaisia kursseja eikä näkyvissä ole esteitä sille, että kehitettyä kurssipohjaa ei voisi hyödyntää esimerkiksi tavanomaisilla ohjelmointikursseilla; koodausleirien tavoin myös ohjelmointikurssit yleisemmin olisi mahdollista jakaa kolmeen vaiheeseen ja kurssipohja tukisi vaiheiden läpivientä. Tulevaisuudessa kurssipohja voisi tukea ohjelmointikursseja esim. tehtäväpalautusten takarajojen ym. kalenteroinnissa, osallistujien ja ryhmien hallinnassa sekä edistymisen seurannassa.

Prototyypin kehittäminen auttoi hahmottamaan jatkotutkimuskohteita aiheen ympäriltä. Jo pelkästään tutkielman rajauksen seurauksena kurssipohjan vaatimuksista jäi kolme selkeää jatkokehittävää kohdetta, sillä sidosryhmäintegraatioon liittyviä ominaisuuksia ei toteutettu. Pohdinnan tuloksena saatiin vielä neljä jatkotutkimusta vaativaa kohdetta:

- Koodausleirien aihepiiristä löytyy relevanttia kirjallisuutta, mutta sen linkittyminen koulutukselliseen näkökulmaan on niukkaa (Fronza ja Corral 2024). *Systemaattinen kirjallisuuskatsaus digitaalisista alustoista koodausleirien kontekstissa* vastaisi tähän haasteeseen.
- Tutkielman aikana kehitetty kurssipohja ei ole vielä käytössä koodausleireillä. *Pienimmän mahdollisen toimivan tuotteen kehittäminen kaaviossa 2 esitettyä prosessia soveltaen* laajentaisi kurssipohjan toiminnallisuuksia sekä tarjoaisi paremmat lähtökohdat mm. käytettävyytutkimuksen suorittamisen.
- *Toimivan tuotteen kenttätestauksen suorittaminen koodausleiritapahtumassa* tuottaisi kurssipohjan jatkokehitykselle ja ylläpidolle ensikäden kokemusta siitä, miten kehitetyt ominaisuudet toimivat oikeassa käytössä ja miten niitä voitaisiin parantaa.
- TIM-alustalle tai kurssipohjalle ei ole vielä tehty kattavaa käytettävyytutkimusta eikä tiedossa ole, millä tavalla käyttäjä näkee alustan esim. ensimmäisellä käyttökerralla. *Käytettävyytutkimus TIM-alustalle kurssipohjaa hyödyntävällä kurssilla* voisi tarjota arvokasta tietoa siitä, mihin suuntaan kurssipohjan ja alustan käytettävyyttä tulisi kehittää.

6 Yhteenveto

Jotta luvussa 2 kuvattujen intensiivikurssi-tyyppisten koodausleirien järjestäminen olisi tulevaisuudessa mielekkäämpää ja tehokkaampaa, tarvitaan niiden läpiviennin tueksi tarkoituksenmukainen digitaalinen oppimisolusta. Tämän suunnittelutieteellisen tutkielman aikana suunniteltiin ja toteutettiin digitaalisen oppimisolustan päälle kurssipohja koodausleirien käyttöön. Tutkielmassa käsitellään kurssipohjan ominaisuuksia ja sitä, miten ne tukevat opettajan ja osallistujien toimintaa koodausleirien eri vaiheissa.

Digitaalinen kurssipohja toteutettiin vaatimusmäärittelyn pohjalta aktiivisessa käytössä olevan oppimisolustan päälle. Toteutus tehtiin viiden eri näkymän avulla, jotka sisälsivät opettajalle ja osallistujille suunnattuja koodausleiritoimintaa tukevia ominaisuuksia. Arvioinnin tuloksena havaittiin, että koodausleirien käyttöön kehitetty kurssipohja vastasi ominaisuuksillaan useimpiin sille esitettyihin vaatimuksiin. Se tarjoaa koodausleirien opettajille ja osallistujille perustoimintoja leirin aktiviteettien hallintaan ja seurantaan, tukee hybridiopetusta koodausleireillä sekä tehostaa koodausleirien järjestämistä.

Havaintojemme perusteella voimme todeta, että kurssipohjan kehittäminen on ollut oikeasuuntaista ja kehitetty kurssipohja tukee koodausleirien läpivientä. Kurssipohjan käyttämiseksi oikealla koodausleirillä suosittelemme kurssipohjan jatkokehittämistä vaatimuksia vastaavaksi pienimmäksi toimivaksi tuotteeksi. Tälle tasolle kehitetty toteutus tarjoaisi riittävät perusominaisuudet koodausleirin tueksi alusta loppuun.

Lähteet

Brusilovsky, Peter, Barbara J. Ericson, Cay S. Horstmann, Christian Servin, Frank Vahid ja Craig Zilles. 2023. “The Future of Computing Education Materials”. Teoksessa *CS2023: ACM/IEEE-CS/AAAI Computer Science Curricula*.

Cronholm, Stefan ja Göran Goldkuhl. 2003. “Six Generic Types of Information Systems Evaluation”. *Electronic Journal of Information Systems Evaluation* 6:65–74. https://www.researchgate.net/publication/266576664_Six_Generic_Types_of_Information_Systems_Evaluation.

Fronza, Ilenia ja Luis Corral. 2024. “A Facilitator’s Guide to Create and Consolidate a Teenage Coding Camp”. *ACM Inroads* 15 (2): 31–38. <https://doi.org/10.1145/3643726>.

Fronza, Ilenia, Luis Corral, Claus Pahl ja Gennaro Iaccarino. 2020. “Evaluating the Effectiveness of a Coding Camp through the Analysis of a Follow-up Project”. Teoksessa *Proceedings of the 21st Annual Conference on Information Technology Education*, 248–253. <https://doi.org/10.1145/3368308.3415391>.

Fronza, Ilenia, Luis Corral, Xiaofeng Wang ja Claus Pahl. 2022. “Keeping fun alive: an experience report on running online coding camps”. Teoksessa *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training*, 165–175. <https://doi.org/10.1145/3510456.3514153>.

Fronza, Ilenia, Gennaro Iaccarino ja Luis Corral. (Hyväksytty julkaistavaksi). “Nurturing Hybrid Work Literacy in Upper Secondary Schools: Selecting the Best Hybrid Work Configuration for Coding Camps”. Teoksessa *2024 IEEE Frontiers in Education Conference (FIE 2024)*.

Fronza, Ilenia, Petri Ihantola, Olli-Pekka Riikola, Gennaro Iaccarino, Tommi Mikkonen, Linda García Rytman, Vesa Lappalainen, Cristina Rebollo Santamaría, Inmaculada Remolar Quintana ja Veronica Rossano. 2024. “Towards s’ more connected coding camps”, <https://arxiv.org/abs/2411.05390>.

Gama, Kiev, Breno Alencar Gonçalves ja Pedro Alessio. 2018. "Hackathons in the formal learning process". Teoksessa *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 248–253. <https://doi.org/10.1145/3197091.3197138>.

Happonen, Ari, Matvei Tikka ja Usman Ahmad Usmani. 2021. "A systematic review for organizing hackathons and code camps in Covid-19 like times: Literature in demand to understand online hackathons and event result continuation". Teoksessa *2021 International Conference on Data and Software Engineering (ICoDSE)*, 1–6. <https://doi.org/10.1109/ICoDSE53690.2021.9648459>.

Hevner, Alan R. 2007. "A Three Cycle View of Design Science Research". Copyright - Copyright Association for Information Systems 2007; Last updated - 2023-12-04, *Scandinavian Journal of Information Systems* 19 (2): 4. <https://www.proquest.com/scholarly-journals/three-cycle-view-design-science-research/docview/2632320917/se-2>.

Hevner, Alan R., Salvatore T. March, Jinsoo Park ja Sudha Ram. 2004. "Design Science in Information Systems Research". *Management Information Systems Quarterly* 28:75–. https://www.researchgate.net/publication/201168946_Design_Science_in_Information_Systems_Research.

Hooper, Steffan, Burkhard C. Wünsche, Andrew Luxton-Reilly, Paul Denny ja Tony Haoran Feng. 2024. "Advancing Automated Assessment Tools - Opportunities for Innovations in Upper-level Computing Courses: A Position Paper". Teoksessa *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 519–525. <https://doi.org/10.1145/3626252.3630772>.

Komssi, Marko, Danielle Pichlis, Mikko Raatikainen, Klas Kindström ja Janne Järvinen. 2015. "What are Hackathons for?" *IEEE Software* 32 (5): 60–67. <https://doi.org/10.1109/MS.2014.78>.

March, Salvatore T. ja Gerald F. Smith. 1995. "Design and natural science research on information technology". *Decision Support Systems* 15 (4): 251–266. [https://doi.org/10.1016/0167-9236\(94\)00041-2](https://doi.org/10.1016/0167-9236(94)00041-2).

Meylani, Rusen, Gary Bitter ja Jane Legacy. 2015. “Desirable Characteristics of an Ideal Online Learning Environment”. *Journal of Educational and Social Research* 5. <https://doi.org/10.5901/jesr.2015.v5n1p203>.

Moore, Joi L., Camille Dickson-Deane ja Krista Galyen. 2011. “e-Learning, online learning, and distance learning environments: Are they the same?” Web mining and higher education: Introduction to the special issue, *The Internet and Higher Education* 14 (2): 129–135. <https://doi.org/10.1016/j.iheduc.2010.10.001>.

Neuwirth, Lorenz S, Svetlana Jović ja B Runi Mukherji. 2021. “Reimagining higher education during and post-COVID-19: Challenges and opportunities”. *Journal of Adult and Continuing Education* 27 (2): 141–156. <https://doi.org/10.1177/1477971420947738>.

Page, Finlay ja Sylvester Sweeney. 2016. “The Use of the “Hackathon” in Design Education: An Opportunistic Exploration”. Teoksessa *International Conference on Engineering and Product Design Education*.

Porras, Jari, Jayden Khakurel, Jouni Ikonen, Ari Happonen, Antti Knutas, Antti Herala ja Olaf Drögehorn. 2018. “Hackathons in Software Engineering Education - Lessons Learned from a Decade of Events”. Teoksessa *2018 IEEE/ACM International Workshop on Software Engineering Education for Millennials (SEEM)*, 40–47. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8442127>.

Porras, Jari, Antti Knutas, Jouni Ikonen, Ari Happonen, Jayden Khakurel ja Antti Herala. 2019. “Code camps and hackathons in education - literature review and lessons learned”. Teoksessa *Hawaii International Conference on System Sciences 2019 (HICSS-52)*, 7750–7759. <https://doi.org/10.24251/hicss.2019.933>.

Praseeda, Monjoy Narayan Choudhury, Bhavjyot Singh Chadha ja Srinath Srinivasa. 2023. “A Systematic Review of Online Learning Platforms for Computer Science Courses”. Teoksessa *2023 IEEE World Engineering Education Conference (EDUNINE)*, 1–6. <https://doi.org/10.1109/EDUNINE57531.2023.10102817>.

Robinson, Philip E. ja Johnson Carroll. 2017. “An online learning platform for teaching, learning, and assessment of programming”. Teoksessa *2017 IEEE Global Engineering Education Conference (EDUCON)*, 547–556. <https://doi.org/10.1109/EDUCON.2017.7942900>.

Shakil, Asma ja Paul Denny. 2024. "Enhancing Student Engagement in Large-Scale Capstone Courses: An Experience Report". Teoksessa *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, 722–728. <https://doi.org/10.1145/3649217.3653580>.

Stephen Lane, Paidi O'Raghallaigh ja David Sammon. 2016. "Requirements gathering: the journey". *Journal of Decision Systems* 25 (sup1): 302–312. <https://doi.org/10.1080/12460125.2016.1187390>.

Ulla, Mark Bedoya ja William Franco Perales. 2022. "Hybrid Teaching: Conceptualization Through Practice for the Post COVID19 Pandemic Education". *Frontiers in Education* 7. <https://doi.org/10.3389/feduc.2022.924594>. <https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2022.924594>.

Venable, John, Jan Pries-Heje ja Richard Baskerville. 2012. "A Comprehensive Framework for Evaluation in Design Science Research", 7286:423–438. https://doi.org/10.1007/978-3-642-29863-9_31.

Zapico, J. L., Pargman D., H. Ebner ja E. Eriksson. 2013. "Hacking sustainability : Broadening participation through Green Hackathons." <https://urn.kb.se/resolve?urn=urn:nbn:se:lnu:diva-42228>.

Liitteet

A Vaatimuslista

- R1* As a participant or a peer-tutor, I want to be warned when the deadline is approaching so that I can pay attention to the activity.
- R2* As a teacher, I want to send information of coding camp details, topic, technologies, etc., to contacts, so that they could prepare for the coding camp.
- R3* As a teacher or a participant, I want to obtain contact details of peer-tutors, so that I can communicate about projects or team issues.
- R4* As a teacher, I want to inform participants to bring their own device.
- R5* As a teacher, I want to store participants' contact details so that, for example, official documents can be prefilled.
- R6* As a teacher, I want a customizable web-form or tool to handle and store all necessary user details, feedback, and such data in a coding camp.
- R7* As a teacher or participant, I want to set, check, and visualize dates for the coding camp participants' group online/hybrid work configuration so that participants may see when to be present.
- R8* As a teacher, I want to keep track of what devices the groups have at lend from university so that I can manage them.
- R9* As a teacher, or participatn, I want a helpdesk to be online ready for helping, when participants are setting things up remotely so that fluent start is ensured.
- R10* As a teacher, I want to ensure that the participants have done all necessary pre-camp actions needed so that they are ready for the coding camp.
- R11* As a teacher, I want to inform peer-tutors for which teams they should follow and support.
- R12* As a teacher, I want to be able to make a timeline for the coding camp to follow the release of materials and instructions, so that I can also mark necessary steps in timeline.
- R13* As a teacher or peer-tutor, I want to contact a participant or team privately so that it is enabled to help them or transfer information about the project.
- R14* As a participant or a peer-tutor, I want to know the composition of the team and have

a brief description of each team member, so that it will be easier to interact with them later.

- R15* As a teacher or peer-tutor, I want to maintain FAQ so that I can avoid repeatedly answering FAQ in real-time chat.
- R16* As a teacher, I want to receive content up to date FAQ from peer-tutors so that I can save my own worktime and make sure that FAQ is up to date.
- R17* As a teacher, I want to have a teacher involved in coding camps also, so that they can continue teaching such topics as taught in coding camp.
- R18* As a teacher, I want to involve teachers or employers by reporting to them the results of what participants did so that coding camps could have continuity.
- R19* As a stakeholder or an OSCAR team member, I would like to get results as a report, what students have done in the coding camp so that stakeholders could be informed via report.
- R20* As a teacher, I want to make notes for facilitators on how the teams are doing so that there is common knowledge on the situation.
- R21* As a teacher, I want to get peer-tutors from the list of early years participants list and contact them or their teachers, so that I can find suitable peer-tutors.
- R22* As a teacher, I mostly want to assign the teams randomly to peer-tutors, but I need to interact with the process so that I can solve specific issues during the process.
- R23* As a teacher or peer-tutor, I want to collect the presence of the participants in lessons so that I can ensure participants are attending all the mandatory hours (if there are such) during the lesson.
- R24* As any role, I want to communicate with other roles both separately and together so that messaging is happening in real time.
- R25* As a teacher, I want to check what participants have done, so that I can evaluate their work and skills.
- R26* As a teacher, I want to see a Gantt chart or a similar roadmap to better organize the coding camp.
- R27* As a participant, I want to see feedback regarding my submission so that I can see if I am doing it in the right direction.
- R28* As a teacher, I want to ensure that GDPR (General Data Protection Regulation) data is

- deleted after a minimum required time so that all users' privacy is respected.
- R29* As a teacher, I want to use tools to publish results, so that all the personal details are anonymized.
- R30* As a participant, I want access to all my material afterwards, so that there is a possibility to download it.
- R31* As a teacher or an OSCAR team member, I want to be able to create pseudonym personas for anonymization so that the data could be used for research purposes.
- R32* As a participant, I want to receive notification when I am assigned new tasks or activities, so that I can be better informed.
- R33* As a participant, I want to see a step-by-step guide to planning, design and developing the project so that I can follow my team's work-in-progress.
- R34* As a teacher, I want to check the state of official documents of each participant.
- R35* As a teacher, I want to provide a note taking template and criteria for peer-tutors so that I can consistently review the job done in groups.
- R36* As an OSCAR team member, I want to be able to visualize all demographics and user data so that I can analyze them.
- R37* As a participant, I want to visualize all the material shared by the facilitator, so that I can read it or download it.
- R38* As a teacher, I want to personalize the virtual environment so that specific educational needs are fulfilled.
- R39* As a teacher, I want to be able to use the gamification strategy so that I can better engage teams or individuals.