

Peltihamsteri-sovellusprojekti

**Mari Kasanen
Leevi Liimatainen
Marina Mustonen
Juhani Sundell
Arttu Ylä-Sahra**

Projektiraportti

Versio 1.0.0

Julkinen

14.6.2019

**Jyväskylän yliopisto
Informaatioteknologian tiedekunta**

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.20__		
Tilaaaja	__.__.20__		
Ohjaaja	__.__.20__		

Tietoja dokumentista

Tekijät:

Marina Mustonen marina.s.mustonen@gmail.com 044-9733582

Dokumentin nimi: Peltihamsteri-projekti, Projektiraportti

Sivumäärä: 67

Tiedosto: Peltihamsteri_Projektiraportti_1.0.0.doc

Tiivistelmä: Peltihamsteri-sovellusprojekti kehitti sovelluksen Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratoriossa kerättävän datan valintaan, synkronointiin ja hallintaan. Projektiraportti kuvaa Peltihamsteri-projektin toteutuneen läpiviennin. Raportissa kuvataan projektin termistöä, tuloksia, resursseja, käytänteitä, ryhmän jäsenten vastuualueita, tehtävien työmääriä, prosessimallia ja aikataulua sekä projektiin vaikuttaneita riskejä ja niiden hallintaa. Raportissa verrataan toteutumaa suunnitelmaan. Erot suunnitelman ja toteuman välillä sekä niiden syyt ja seuraukset läpivientiin ja tuloksiin kuvataan. Lisäksi kuvataan ryhmän jäsenten kokemuksia projektista ja mitä he kokivat oppineensa sen aikana.

Avainsanat: Aikataulu, kokemuksia, käytänteet, oppimaa, projektihallinta, projektiorganisaatio, projektiraportti, prosessimalli, resurssit, riskit, sovellusprojekti, tavoitteet, tehtäväjako, tehtävät, toteuma, työmäärät, vastuualueet.

Muutoshistoria

Versio	Päivä	Muutokset	Tekijä
0.0.1	11.4.2019	Projektiraportin laatiminen aloitettiin.	Marina Mustonen
0.0.2	18.4.2019	Tarkennettiin ja korjailtiin erinäisiä kohtia aikatauluun asti.	Marina Mustonen
0.0.3	23.4.2019	Tarkennettiin ja korjailtiin erinäisiä kohtia aikataulusta loppuun asti.	Marina Mustonen
0.0.4	14.5.2019	Korjailtiin ja täydennettiin raporttia.	Marina Mustonen
0.0.8	23.5.2019	Lisättiin käytetyt tunnit ja hiottiin riskien toteutumista ja muita kohtia.	Marina Mustonen
0.1.0	24.5.2019	Viimeisteltiin raporttia.	Marina Mustonen
0.1.5	31.5.2019	Korjailtiin raporttia ohjaajien palautteen perusteella.	Marina Mustonen
0.2.0	5.6.2019	Lisättiin päivitetyt taulukot ja graafit sekä hiottiin tekstiä.	Marina Mustonen
0.3.0	10.6.2019	Korjailtiin ohjaajan kommenttien perusteella.	Marina Mustonen
1.0.0	14.6.2019	Tehtiin pieniä korjauksia ohjaajan kommenttien perusteella.	Marina Mustonen

Tietoja projektista

Peltihamsteri-sovellusprojekti kehitti sovelluksen Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratoriossa kerättävän datan synkronointiin ja hallintaan. Ajosimulaatiokokeessa saadaan erilaisia mittaustuloksia useammalta laitteelta ja kehitetyllä sovelluksella laitteista kerättävä data saadaan valittua ja synkronoitua samaan tiedostoon. Ohjelma kehitettiin C#-kielellä Microsoft Windows -ympäristöön.

Tekijät:

Mari Kasanen	<code>r.mari.s.kasanen@student.jyu.fi</code>	044-2359811
Leevi Liimatainen	<code>leevi.m.m.liimatainen@student.jyu.fi</code>	050-5052731
Marina Mustonen	<code>marina.s.mustonen@gmail.com</code>	044-9733582
Juhani Sundell	<code>juhani.k.sundell@student.jyu.fi</code>	044-2666773
Arttu Ylä-Sahra	<code>arttu.e.yla-sahra@student.jyu.fi</code>	044-0668949

Tilaaajan edustajat:

Hilkka Grahn	<code>hilkka.grahn@jyu.fi</code>	040-8053342
Tuomo Kujala	<code>tuomo.kujala@jyu.fi</code>	0400-247392

Ohjaajat:

Jonne Itkonen	<code>jonne.itkonen@jyu.fi</code>	050-4432381
Jukka-Pekka Santanen	<code>santanen@jyu.fi</code>	040-8053299

Sisältö

1	Johdanto	1
2	Termit	3
3	Tavoitteiden toteutuminen ja tulokset	7
3.1	Taustaa, tarpeita ja ominaisuuksien toteutuminen.....	7
3.2	Sovelluksen toteutuneet tavoitteet.....	8
3.3	Sovelluksen toteutumattomat tavoitteet	9
3.4	Toteutetun sovelluksen rakenne	10
3.5	Sovelluksen työnkulku ja toteutusratkaisut.....	11
3.6	Tulokset.....	12
3.7	Oppimistavoitteet	13
4	Projektiorganisaatio ja resurssit.....	15
4.1	Projektiorganisaatio.....	15
4.2	Tilat ja laitteet.....	16
4.3	Kehitys- ja dokumentointityökalut.....	17
4.4	Luennot ja perehdytykset.....	17
5	Käytänteet.....	19
5.1	Palaverit.....	19
5.2	Tiedotus.....	20
5.3	Tiedostojen nimeäminen.....	20
5.4	Hakemistorakenne	21
5.5	Lähdekoodi	22
5.6	Testaus.....	22
5.7	Versiohallinta ja -numerointi	23
5.8	Katselmoinnit ja tulosten hyväksyminen	24
5.9	Tulosten koostaminen ja toimittaminen	24
6	Roolit, vastualueet ja tehtävät	25

6.1	Roolit ja vastualueet.....	25
6.2	Tehtävät ja niiden työmäärät	27
6.3	Projektinhallinnan työmäärät.....	28
6.4	Palaverien työmäärät	29
6.5	Esitutkimuksen työmäärät	29
6.6	Vaatimusmäärittelyn työmäärät	30
6.7	Suunnittelun työmäärät.....	30
6.8	Toteutuksen työmäärät.....	31
6.9	Järjestelmätestauksen työmäärät.....	32
6.10	Hyväksymistestauksen työmäärät.....	32
6.11	Viimeistelyn työmäärät	33
6.12	Oheiskurssien työmäärät.....	33
6.13	Ryhmän työtunnit tehtäväkokonaisuuksittain.....	34
6.14	Kasasen työtunnit tehtäväkokonaisuuksittain	35
6.15	Liimataisen työtunnit tehtäväkokonaisuuksittain	36
6.16	Mustosen työtunnit tehtäväkokonaisuuksittain	37
6.17	Sundellin työtunnit tehtäväkokonaisuuksittain	38
6.18	Ylä-Sahran työtunnit tehtäväkokonaisuuksittain	39
7	Prosessi ja aikataulu.....	40
7.1	Prosessi.....	40
7.2	Projektin kehitysvaiheet.....	41
7.3	Ryhmän tilalaverit ja tilakatsaukset.....	42
7.4	Aikataulu.....	43
7.5	Ryhmän työtunnit viikoittain.....	46
7.6	Kasasen työtunnit viikoittain	47
7.7	Liimataisen työtunnit viikoittain	48
7.8	Mustosen työtunnit viikoittain.....	49

7.9	Sundellin työtunnit viikoittain	50
7.10	Ylä-Sahran työtunnit viikoittain	51
8	Riskien hallinta.....	52
8.1	Riskien todennäköisyydet ja haittavaikutukset.....	52
8.2	Laitteiden erilaisuus ja tietojen puute.....	53
8.3	Sovelluksen joustavuuden toteuttamisen vaikeus.....	54
8.4	Ominaisuuksien toteuttamisen vaikeus.....	54
8.5	Ryhmän tietotaitojen puutteet.....	55
8.6	Ryhmän jäsenten muut sitoumukset	56
9	Jäsenten kokemuksia ja oppimaa	57
9.1	Kasasen kokemuksia ja oppimaa	57
9.2	Liimataisen kokemuksia ja oppimaa	58
9.3	Mustosen kokemuksia ja oppimaa	60
9.4	Sundellin kokemuksia ja oppimaa	61
9.5	Ylä-Sahran kokemuksia ja oppimaa	63
10	Yhteenveto.....	65
	Lähteet	66

1 Johdanto

Peltihamsteri-projekti kehitti Sovellusprojekti-kurssilla keväällä 2019 Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratoriolle sovelluksen tutkimuksissa kerättävän datan valintaan, synkronointiin ja hallintaan. Ajolaboratoriossa kehitetään testausmenetelmiä teollisuuden ajoneuvokäyttöliittymiä varten, sekä tehdään perustutkimusta kuljettajien tarkkaavaisuudesta, visuaalisesta havainnoinnista ja liikennekäyttäytymisestä. Ajosimulaatiokokeessa saadaan dataa useasta eri laitteesta. Projektiryhmä kehitti sovelluksen, jolla näistä valituista laitteista kerättävä data saadaan synkronoitua keskenään sen käsittelyn ja analysoinnin helpottamiseksi.

Projektiraportti kuvaa, miten projekti toteutui tulosten, resurssien, käytänteiden, tehtäväjaon, työmäärien, aikataulun, riskien hallinnan ja jäsenten kokemusten osalta. Toteutunutta läpivientiä verrataan projektisuunnitelmassa [1] kuvattuun suunniteltuun läpivientiin ja kuvataan syitä erojen taustalla ja miten erot vaikuttivat projektin läpivientiin ja tuloksiin. Projektiraportin laatimisessa on hyödynnetty Monisiro-projektin projektiraporttia [2] ja sovellusprojektien ohjetta [3].

Projektiraportin lisäksi projektiryhmä laati ohjelmistolle vaatimusmäärittelyn [4], jossa kuvataan sovelluksen toimintoja sekä niiden prioriteetteja ja tilaa. Ryhmä laati myös sovelluksen rakennekuvauksen [5], jossa kuvataan sovelluksen rakennetta ja toimintoja eri näkymien kautta. Ryhmän laatimassa järjestelmätestaussuunnitelmassa [6] kuvataan ohjelmistolle suoritettuja testitapauksia, sekä järjestelmätestausraporteissa [7]–[9] testauskertojen tuloksia. Ryhmä laati myös sovellusraportin [10], jossa kuvataan sovelluksen toteutusratkaisuja sekä mahdollisia puutteita ja jatkokehitysideoita. Kaikista projektin palaverista laadittiin pöytäkirjat ja väliesittelyistä muistiota.

Projektiraportin luvussa 2 esitellään projektissa käytettyjä termejä. Luvussa 3 kuvataan projektin taustoja ja tavoitteita, sekä tavoitteiden ja projektiryhmän jäsenten oppimistavoitteiden toteutumista. Luvussa 4 esitellään projektiin osallistuneet henkilöt ja projektiryhmän resurssit. Luku 5 kuvaa projektin käytänteitä. Luku 6 kuvaa projektin jäsenten roolit, vastuualueet, tehtävät ja toteutuneet työmäärät. Luvussa 7 kuvataan sovelluskehitysprosessia ja aikataulua. Luku 8 kuvaa projektin läpivientiin liittyneiden riskien toteutumista, hallintaa ja vaikutuksia. Luvussa 9 kuvataan projektiryhmän jäsenten oppimiskokemukset.

2 Termit

Luvussa kuvataan Peltihamsteri-projektiin liittyviä termejä. Dokumentissa esiintyvät aihealueen termit ovat seuraavat:

AOI	eli Area of Interest on silmänliikekameran ohjelmistoon määritettävät alueet, joista mitataan kuinka usein ja kuinka kauan koehenkilön katse kohdistuu niihin.
Asetustiedosto	(eng. <i>Study settings</i>) on tiedosto, johon tallennetaan kokeen asetukset, jolloin samoja asetuksia voi käyttää useamman eri koehenkilön kanssa.
CSV	eli Comma-Separated Values on tiedostomuoto, jolla tallennetaan yksinkertaista taulukkomuotoista tietoa tekstitiedostoon.
Datan jäsentäminen	on arvojen erottelua ja tunnistamista raakadatasta.
Katsedata	sisältää tiedon, kuinka usein ja kuinka kauan koehenkilön katse kohdistuu AOI:ksi (kts. yllä) määritettyyn kohteeseen.
Koe	(eng. <i>Study</i>) on yhden koehenkilön kanssa suoritettava 1–2 h kestävä ajosimulaatiokerta, jossa koehenkilö suorittaa tehtäviä ja suorituksesta kerätään dataa.
Lua-skripti	on tilaajan käyttämä kooditiedosto ajosimulaattoridatan mittaustulosten valikointiin.
R	on ohjelmointikieli tilastolliseen laskentaan ja grafiikan muodostamiseen.

Tehtävä	(eng. <i>task</i>) on kokeen aikana koehenkilön suorittama yksi ajosuorite.
Toissijainen tehtävä	on tehtävän aikana koehenkilön suorittama muu toiminto kuin ajaminen, esimerkiksi sovelluksen käyttäminen testilaitteella.
Tulostiedosto	on kehitetyn sovelluksen muodostama CSV-tiedosto, johon on yhdistetty ja synkronoitu valituilta laitteilta saatavat valitut dataosiot yhden koehenkilön yhdestä tehtävästä.

Dokumentissa esiintyvät kohdealueen laitteiden ja ohjelmistojen termit ovat seuraavat:

Ajosimulaattori	on laite, jolla simuloidaan oikean ajoneuvon ajamista virtuaalitodellisuudessa.
Android	on Linux-pohjainen käyttöjärjestelmä, jota käytetään yleisimmin älypuhelimissa ja tableteissa.
D-Lab	on tilaajan käyttämän silmänliikekameran ohjelmisto.
DSI-Streamer	on tilaajan käyttämän EEG-laitteen ohjelmisto.
EEG	eli elektroenkefalografia tarkoittaa aivosähkökäyrää.
EepSoft	on tilaajan käyttämän ajosimulaattorin ohjelmisto.
Ergoneers Professional	Dikablis on tilaajan käyttämä uusi silmänliikekamera.
Noldus Observer XT 12	on videopisteytys- ja datan synkronointiohjelmisto.

QStates	on EEG-laitteen kanssa käytettävä ohjelmisto, jolla laskettavia tunnuslukuja käytetään muun muassa koehenkilön kuormituksen arvioimiseen.
SEER	on tilaajan aiemmin käyttämän silmänliikekameran ja ajosimulaattorin datan synkronointiin kehitetty ohjelmisto.
Silmänliikekamera	on laite silmän aseman ja liikkeiden mittaamiseen.
VisGuard Observer	on taustasovellus, jolla kognitiotieteen tutkimusryhmä on kerännyt syötetietoja koehenkilön käyttämästä laitteesta.
Wearable Sensing DSI-	on tilaajan EEG-laite.

24

Dokumentissa esiintyvät kehitysvälineisiin ja -tekniikoihin, sekä projektin hallintaan liittyvät termit ovat seuraavat:

C#	on Microsoftin .NET-alustalle kehitetty ohjelmointikieli.
Git	on hajautettu versiohallintaohjelmisto lähdekoodien ja dokumenttien hallintaan.
Hyväksymistestaus	on tilaajien suorittama testaus sovellukselle, jolla varmistetaan sovelluksen täyttävän tilaajien asettamat tarpeet.
Katselmointi	on silmäkkäin tapahtuva tulosten tarkastaminen, jossa tarkastetaan toteutukset ja niiden erot alkuperäiseen suunnitelmaan, sekä esitetään parannusehdotuksia.

Käyttöliittymä	on ohjelman ihmiselle näkyvä osa, jonka avulla käyttäjä ja ohjelma ovat vuorovaikutuksessa keskenään.
Lähdekoodi	on tietokoneohjelman tekstimuotoinen ohjelmointikielinen listaus.
Tilakatsaus	on projektipäällikön tekemä viikottainen katsaus, jossa käydään läpi projektin eteneminen.
WPF	eli Windows Presentation Foundation on .NET-kirjasto, joka muodostaa rajapinnan ja toimii käyttöliittymän pohjana.
Xamarin	on natiivien sovellusten kehittämiseen tarkoitettu kehitystyökalu, jolla koodin saa helposti jaettua eri käyttöjärjestelmien välillä.
YouSource	on Gitorius-pohjainen paikallinen järjestelmä lähdekoodien ja materiaalien hallintaan ja julkistamiseen.

3 Tavoitteiden toteutuminen ja tulokset

Luvussa käsitellään projektissa kehitetylle sovellukselle, ryhmän oppimiselle ja muille tuloksille asetettuja tavoitteita ja niiden toteutumista. Kaikki korkeimman prioriteetin vaatimukset saatiin toteutettua sovellukseen. Toteuttamatta jäi vain yksi valinnainen ja neljä ideaksi kirjattua vaatimusta. Muut tulokset saatiin toteutettua suunnitellusti ja ryhmän jäsenten oppimistavoitteet toteutuivat.

3.1 Taustaa, tarpeita ja ominaisuuksien toteutuminen

Projektin tilaajana toimi Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratorio. Ajosimulaatiolaboratoriossa on vuosina 2007–2019 tehty kokeellisia tutkimuksia, joissa on keskitytty mittaamaan kuljettajan tarkkaavaisuutta ja liikennekäyttäytymistä, sekä esimerkiksi teollisuuden ajoneuvokäyttöliittymien ja muiden laitteiden käytön vaikutusta niihin. Kokeissa kerätään dataa tyypillisesti vähintään kolmelta eri laitteelta (yleensä ajosimulaattori, silmänliikekamera ja toissijaisen tehtävän laite). Myös EEG-laitteelta on tarkoitus kerätä aivosähkökäyrädataa tulevissa kokeissa.

Datan käsittelyt ja analyysit ovat aiemmin sisältäneet paljon virhealtista manuaalista työtä. Ajosimulaattorista dataa kerätään EepSoft-ohjelmiston ja silmänliikekamerassa D-Lab-ohjelmiston kautta. EEG-laitteella on käytössä kaksi eri ohjelmistoa: sen oma ohjelmisto DSI-Streamer ja koehenkilön kuormituksen arviointiin tarkoitettu QStates.

Kehitetyn Syncster-sovelluksen päävaatimus oli, että sillä saa datan synkronoitua ajosimulaattorista, silmänliikekamerasta ja EEG-laitteesta. Lisäksi oli tarvetta kehittää Android-laitteille sovellus, jolla saa kerättyä syötetietoja koehenkilön kokeen aikana käyttämästä laitteesta. Myös syötetiedot synkronoidaan muun kerätyn datan kanssa. Muu data voidaan kerätä reaaliajassa, mutta jälkikäteen synkronoidaan silmänliikekameran AOI-data, ts. kuinka usein ja kauan

koehenkilön katse kohdistuu AOI:ksi määriteltyyn kohteeseen. Kehitetty sovellus ei korvaa laitteiden omia ohjelmistoja, mutta vähentää virhealtista manuaalista työtä.

Sovellukselle kehitettiin myös helppokäyttöinen käyttöliittymä, jonka kautta voi valikoida halutut laitteet ja niiden dataosiot, täyttää kokeen ja tallennuksen tiedot, aloittaa ja lopettaa tallennuksen sekä yhdistää datan yhteen tulostiedostoon. Käyttöliittymässä myös visualisoidaan, tuleeko laitteilta dataa ja tallennuuko se oikein.

3.2 Sovelluksen toteutuneet tavoitteet

Kaikki tärkeimmän prioriteetin vaatimukset saatiin toteutettua. Toteutetun sovelluksen olennaisimmat ominaisuudet ovat seuraavat:

- Sovellus toimii käyttöjärjestelmässä Microsoft Windows 10.
- Kokeessa käytettävät laitteet voidaan valita.
- Käytettävien laitteiden tarjoamasta datasta voidaan valita halutut dataosiot.
- Sovelluksen muodostama tulostiedosto on CSV-muotoinen.
- Tulostiedoston tiedostonimi sisältää tiedot kokeesta, koehenkilöstä, tehtävästä ja tehtävän suoritusajasta (aikaleima).
- Tulostiedosto sisältää yhden koehenkilön yhden tehtävän valittujen laitteiden valitut dataosiot synkronoituina.
- Tulostiedosto voidaan muodostaa niin, että se sisältää myös jälkikäteen lisättyä dataa.

Graafisen käyttöliittymän (GUI) olennaisimmat ominaisuudet ovat seuraavat:

- Kokeessa käytettävät laitteet voidaan valita.
- Käytettävien laitteiden tarjoamasta datasta voidaan valita halutut dataosiot.
- Käytettävä kieli on englanti.
- Tallennus saadaan päälle.

- Tallennus saadaan pois päältä.
- Sovellusta voidaan käskeä muodostamaan tulostiedosto.
- GUI ilmaisee visuaalisesti, tuleeko laitteilta dataa ja tallentuuko se.

Eri laitteilla on erilaisia viiveitä ja tiedostomuotoja. Ryhmä kehitti sovelluksen tukemille eri laitteille omat moduulit, jotka lukevat datan laitteilta ja kirjoittavat ne väliaikaisiin omiin tiedostoihinsa jäsenneyssä ja yhtenäisessä muodossa. Dataan kirjataan aikaleimat, jolloin data voidaan yhdistää ja synkronoida yhteen tulostiedostoon. Uusien laitteiden lisääminen sovellukseen tehtiin ohjelmointitaitoiselle käyttäjälle mahdolliseksi valmiskomponentteja ja selkeitä ohjeita hyödyntäen. Sovelluksen käyttöä helpottamaan kehitettiin graafinen käyttöliittymä.

3.3 Sovelluksen toteutumattomat tavoitteet

Toteuttamatta jäi vaatimus siitä, että tallennuksen voi aloittaa vasta, kun valitun laitteen dataosio(ita) on valittuna. Vaatimus oli valinnainen ja jäi toteuttamatta sen monimutkaisen toteutuksen ja ajanpuutteen vuoksi. Tilaajalle sopi ratkaisuksi se, että ohjeissa kerrotaan dataosioiden valinnasta.

Toteuttamatta jätettiin myös vaatimus tunnuslukujen laskemisesta automaattisesti datasta. Tämä vaatimus oli tilaajan alkuperäisessä projektiehdotuksessa, mutta sovittiin jo hyvin varhaisessa vaiheessa projektia tilaajan kanssa tehtäväksi vain, jos projektissa jää aikaa. Vaatimusmäärittelyssä vaatimus oli kirjattu ideaksi.

Toteuttamatta jäivät vaatimukset siitä, että silmänliikekameran laitemoduuli osaa käynnistää ja lopettaa tallennuksen D-Labissa sekä luoda D-Labiin *studyn* ja *subjectin* sovelluksessa määritellyillä kokeen ja koehenkilön tunnisteilla etäyhteyden avulla. Vaatimukset oli kirjattu ideoiksi ja jäivät toteuttamatta laitevalmistajalle esitetyn ohjeistuspyynnön myöhäisen vastauksen ja ajanpuutteen takia. Ominaisuudet sovittiin jatkokehitykseen.

Tilaaajan kanssa sovittiin, että käyttöohje toimitetaan ensisijaisesti englanniksi, mutta jos projektissa jää aikaa niin myös suomeksi. Vaatimus suomenkielisestä käyttöohjeesta oli kirjattu ideaksi ja jäi toteuttamatta ajanpuutteen vuoksi.

Lisäksi toteuttamatta jäi kaksi vaatimusta: käyttöliittymän kirjoittaminen suomeksi ja tehtävän tunnusteen kysyminen tallennusta aloittaessa. Ne sovittiin tilaaajan kanssa jo projektin alussa tarpeettomiksi. Käyttöliittymän kieleksi sovittiin tilaaajan kanssa englanti, joten suomenkielinen käyttöliittymä päätettiin jättää toteuttamatta. Projektin alussa tilaaajan kanssa keskusteltiin, että tallennusta aloittaessa sovellus voisi kysyä tehtävän tunnustetta, mutta pian tilaaja päätyi sen olevan huono ratkaisu.

3.4 Toteutetun sovelluksen rakenne

Kehitetty Syncster-sovellus synkronoi datan eri laitteista. Alun perin tilaaja toivoi, että synkronointi tapahtuisi reaaliajassa. Ryhmän perehdyttyä kokeissa käytettäviin laitteisiin ilmeni, että reaaliaikainen synkronointi ei ole käytännöllinen ratkaisu. Tilaaajalle sopi myös ratkaisu, jolla data saadaan synkronoitua jälkikäteen aikaleimojen avulla. Kehitettyssä sovelluksessa data kerätään, jäsennetään ja kirjoitetaan väliaikaisiin tiedostoihin reaaliaikaisesti niiltä osin kuin se on mahdollista. Data yhdistetään ja synkronoidaan datan keräyksen jälkeen.

Ajosimulaattorihjelmisto on hyvin joustava ja räätälöitävissä erilaisiin tarpeisiin. Myös uusi silmänliikekamera (Ergoneers Dikablis Professional) ja EEG-laite (Wearable Sensing DSI-24) tukevat joiltain osin reaaliaikaisuutta. Toissijaisen tehtävän laitteen syötetietojen reaaliaikaiseen keräämiseen projekti-ryhmä kehitti sovelluksen ottaen mallia kognitiotieteen tutkimusryhmän muissa tutkimuksissa käyttämää VisGuard Observer -taustasovelluksesta. Syötetietojen keräämiseen tarkoitettun sovelluksen nimeksi tuli Touchster. Touchster kerää dataa siitä, milloin koehenkilö koskee toissijaisen tehtävän kosketusnäyttöön ja Syncster synkronoi tämän datan muilta laitteilta kerätyn datan kanssa.

3.5 Sovelluksen työnkulku ja toteutusratkaisut

Kehitetty sovellus synkronoi eri laitteilta tulevan datan jatkokäsittelyyn ja analyysien helpottamiseksi. Data kerätään joko TCP-yhteydellä (EEG, silmänliikekamera ja toissijaisen tehtävän laite) tai UDP-yhteydellä (ajosimulaattori). Laitteista kerätään kaikki data ja suodatetaan sitten pois osiot, joita ei haluta. Toissijaisen tehtävän laitteessa on vain yksi dataosio, jolloin ei ole tarvetta suodattaa pois mitään.

Data muutetaan ohjelmointikielen tukemiksi tietotyypeiksi ja tallennetaan väliaikaisiin tiedostoihin. Niistä data välitetään datan yhdistämisestä vastaavalle komponentille. Jokaisella tietotyypillä on yhtenäinen rajapinta merkkijonoksi muuntamiseen, jolloin CSV-tiedoston muodostaminen onnistuu. Synkronoinnissa hyödynnetään eri laitteiden datalle annettuja yhtenäisiä aikaleimoja, jotka lasketaan laitekohtaisesti hyödyntäen esimerkiksi laitteiden ohjelmistojen omia aikaleimoja ja tallennuksen aloitusajankohtaa.

Kehitetty sovellus tallentaa jokaisen yksittäisen koehenkilön yksittäisestä koetehtävästä eri laitteista kerätyt dataosiot yhteen tulostiedostoon. Tiedoston otsikkorivistä selviää, mitä dataa kukin sarake sisältää, jolloin tiedoston voi sellaisenaan jakaa avoimena datana. Tulostiedoston nimi generoidaan käyttäjän määrittämän kokeen nimen, koehenkilön koodin, tehtävän nimen ja vapaaehtoisin kommentin sekä aikaleiman perusteella.

Sovellus kehitettiin työasemalle C#-kielellä, ja se toimii Microsoft Windows -ympäristössä. Sovelluksen käytön helpottamiseksi kirjoitettiin käyttöohjeet. Sovellus kommentoitiin huolella, jotta käyttäjät voivat kehittää sovellusta eteenpäin. Jatkokehityksen mahdollistamiseksi kehitettiin myös tiedossa olevien laitteiden valmiskomponentteja, joita voi hyödyntää uusien laitteiden lisäämisessä sovellukseen. Laitemoduulien lisäämiseen laadittiin ohjeet, mutta lisääminen vaatii jatkokehittäjältä ohjelmointitaitoja, sillä lisättävien laitteiden ominaisuudet pitää ottaa huomioon laitemoduulia lisättäessä.

3.6 Tulokset

Sovelluksen lisäksi projektiryhmä toteutti seuraavat tulokset:

- **Ajankäyttöraportti** sisältää ryhmän jäsenten työtunnit sekä niiden jakautumisen eri tehtäväkokonaisuuksille ja tehtäville.
- **Esittelymateriaalit** sisältävät väli- ja loppuesittelyjen esitysgrafiikat ja muistiot.
- **Itsearviointit** sisältävät ryhmän jäsenten arviot omasta toiminnastaan, kokemuksistaan ja oppimisestaan projektin aikana.
- **Käyttöohjeet** neuvovat, miten sovellusta käytetään.
- **Lisenssisitoumuksella** projektiryhmän jäsenet sitoutuvat sijoittamaan toteuttamansa lähdekoodin valitun avoimen lähdekoodin lisenssin alaisuuteen.
- **Lähdekoodi** sisältää sovelluksen lähdekoodin kommentteineen.
- **Palaverien dokumentit** sisältävät palaverien esityslistat, tilakatsaukset ja pöytäkirjat.
- **Projektiraportti** kuvaa projektin toteutunutta läpivientiä, ryhmän jäsenten kokemuksia ja oppimaa sekä vertaa toteumaa suunnitelmaan kuvaten mm. syitä ja vaikutuksia.
- **Projektisuunnitelma** on projektin läpivientiä kuvaava suunnitelma, jossa kuvataan muun muassa projektin termistö, tavoitteet, projektiorganisaatio ja resurssit, projektiryhmän jäsenten roolit, vastuualueet, tehtävät ja työmäärät sekä projektissa käytettävä prosessimalli, aikataulu ja riskien hallintaa.
- **Rakennekuvaus** kuvaa sovelluksen rakennetta kaavioiden ja käyttötapusten avulla.
- **Sovellusraportti** kuvaa toteutetun sovelluksen oleellisimmat toteutusratkaisut ja toiminnot, tavoitteiden toteutumista sekä mahdolliset puutteet, heikot toteutusratkaisut ja jatkokehitysideat.

- **Sähköpostiarkistot** sisältävät kaikki projektin kahdelle sähköpostilistalle lähetetyt sähköpostiviestit.
- **Tekninen ohje** sisältää ohjeet ryhmän käyttämän kehitysympäristön muodostamiseen jatkokehityksen helpottamiseksi, Syncsterin ja Touchsterin asentamiseen sekä uusien laitteiden lisäämiseen sovellukseen valmiskomponentteja hyödyntäen.
- **Testausraportit** kuvaavat kullakin testauskerralla suoritettujen testitapausten tulokset sekä havaitut virheet ja puutteet.
- **Testaussuunnitelmat** kuvaavat testauskerroilla suoritettavat testitapaukset ja testausympäristön kokoonpanon.
- **Vaatimukset** sisältävät sovelluksen käyttäjilleen tarjoamat tiedot ja toiminnot.

Tulokset toteutuivat pääosin suunnitellusti. Alun perin sovelluksella oli tarkoitus olla arkkitehtuurikuvaus, mutta sovellukselle tehtiin sen sijaan rakennekuvaus. Sovelluksen toimintaa kuvaamaan tehdyt kaaviot ja käyttötapaukset eivät ole yksiselitteisesti arkkitehtuurikuvausta, joten niitä on parempi kutsua yleisemmin rakennekuvaukseksi. Toinen ero suunniteltuun oli, että kehitysympäristön pystytysohjeesta ja laitemoduulien lisäämisohjeesta ei kirjoitettu omia dokumenttejaan vaan kirjoitettiin tekninen ohje, joka sisältää myös em. ohjeet.

3.7 Oppimistavoitteet

Sovellusprojektissa projektiryhmän jäsenten oppimistavoitteet toteutuivat suunnitellusti. He oppivat ohjelmistokehityksen tietotaitoja ja saivat kokemusta projektityöskentelystä. Peltihamsteri-projektin jäsenet saivat projektissa kattavan kuvan ohjelmistokehitysprojektin läpiviennistä aina aihealueeseen tutustumisesta tulosten toimittamiseen saakka. Projektissa oleellista oli kokonaiskuvan saamisen lisäksi ryhmässä työskentelyn taitojen kehittyminen, oman roolin ja vastuun ymmärtäminen, ajankäytön suunnittelu sekä aiemmillä

kursseilla opittujen taitojen hyödyntäminen käytännössä. Lisäksi erityisesti projektipäällikkö ja varapäällikkö oppivat kurssilla projektin hallintaa ja ryhmän johtamista. Projekti antoi myös selkeyttä siihen, millaisissa työtehtävissä haluaa tulevaisuudessa toimia.

Projektissa oli myös tärkeää viestintätaitojen kehittäminen. Projektiryhmä oppi viestimään niin sisäisesti jäsenten kesken kuin ulkoisesti tilaajan edustajien ja ohjaajien kanssa. Jäsenet myös oppivat laatimaan kirjallisia dokumentteja. Lisäksi jäsenet saivat kokemusta suullisesta viestinnästä projektin esittelyissä.

Edellisten tavoitteiden lisäksi jäsenet asettivat seuraavia henkilökohtaisia tavoitteita:

- Mari Kasasen tavoitteena oli oppia projektityötapoja, jotka soveltuvat useamman kuukauden pituiseen työskentelyyn. Pienemmillä kursseilla ei ole mahdollista oppia pitkäkestoista projektityöskentelyä.
- Leevi Liimataisen tavoitteena oli saada kokemusta projektitoiminnan työtavoista ja parantaa osaamistaan ohjelmistokehittämisessä.
- Marina Mustosen tavoitteena oli oppia projektin suunnittelua ja hallintaa, viestintätaitoja ja ohjelmistokehitystyökalujen käyttöä sekä johtamista.
- Juhani Sundellin tavoitteena oli saada aidonoloista kokemusta ohjelmistokehityksestä. Hän oli aiemmin tehnyt ohjelmointitöitä joko omaksi ilokseen tai opettajan vaatimusten mukaan, mutta ei koskaan tilaajan tarpeiden mukaan. Lisäksi hänen viime vuosien harjoitustyönsä ovat pääasiassa olleet WWW-sovelluksia, joten sovellusprojektin aihe tarjosi mahdollisuuden oppia uusia asioita.
- Arttu Ylä-Sahran tavoitteena oli kehittää projektitaitojaan ml. viestintää ja vuorovaikutusta, jotka hän koki heikommiksi kohdikseen. Hän halusi myös kehittää ohjelmoinnissa tarvittavia teknisiä yhteistyötaitoja, kuten Git-versiohallinnan käyttöä.

Ryhmän jäsenet kokevat oppimistavoitteidensa toteutuneen hyvin.

4 Projektiorganisaatio ja resurssit

Luku esittelee sovellusprojektin jäsenet ja ohjaajat sekä tilaajan edustajat. Luvussa kuvataan myös projektissa käytössä olleita tiloja, laitteita ja kehitystyökaluja. Kuvatut resurssit toteutuivat suunnitelman mukaisesti.

4.1 Projektiorganisaatio

Projektiorganisaatio toteutui suunnitellusti. Projektiryhmään kuuluivat Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra. Kaikki jäsenet ovat tietotekniikan maisteriopiskelijoita. Ylä-Sahra on ohjelmointikielten periaatteiden maisteriohjelmassa ja muut ovat ohjelmistotekniikan maisteriohjelmassa. Marina Mustonen toimi projektipäällikkönä ja Mari Kasanen varapäällikkönä.

Projektiryhmän jäsenillä ei ollut ennestään kokemusta tietotekniikan projekti-
muotoisesta työskentelystä tai asiakkaan kanssa toimimisesta. Sovellusprojek-
tin oheiskursseilla käsiteltiin kuitenkin projektin läpiviennin ja hallinnan taitoja
sekä viestintätaitoja, joita projektissa hyödynnettiin.

Kehitetyn ohjelmiston kieli C# oli kaikille ryhmän jäsenille tuttu ennestään. Kasasella ja Liimataisella oli kokemusta käyttöliittymien tekemisestä ja vahvuuksia visuaalisella puolella, joten he keskittyivät enemmän käyttäjälle näkyvään osaan sovelluksen kehittämisessä. Sundellilla ja Ylä-Sahralla oli enemmän kokemusta sovellusten suunnittelusta ja toteutuksesta, joten he keskittyivät enemmän taustalla toimivaan sovelluksen osaan. Kasanen, Liimatainen, Sundell ja Ylä-Sahra olivat myös vastuussa kukin yhdestä laitemoduulista. Mustosella oli kokemusta projektin johtamisesta ja tieteellisen tekstin tuottamisesta aiemmista opinnoistaan biologian puolella, joten hän keskittyi enemmän projektin hallintaan ja dokumenttien tuottamiseen.

Tilaaajaorganisaatio oli Jyväskylän yliopiston informaatioteknologian tiedekunnan kognitiotieteen ajosimulaatiolaboratorio. Tilaaajan edustajina toimivat tutkijatohtori Tuomo Kujala ja tohtorikoulutettava Hilikka Grahn.

Projektin vastaavana ohjaajana toimi tietotekniikan laitoksen Jonne Itkonen. Itkonen toimi myös projektin teknisenä ohjaajana. Toisena ohjaajana toimi Jukka-Pekka Santanen. Jyväskylän yliopiston Digipalvelut vastasi ryhmän jäsenten tietokoneiden ja ohjelmistojen ylläpidosta.

Sovellusprojekti-kurssin yhteydessä järjestettävän *Projektiviestintä IT-alalla* -kurssin opettajina olivat Hanna Kivimäki ja Kati Rantala-Lehtola. *Sovellusprojektin hallintaa, viestintää ja työkaluja* -kurssin pääasiallisena opettajana ja yhteyshenkilönä toimi Jukka-Pekka Santanen. Käytettävyyispäivän veti Hilikka Grahn.

4.2 Tilat ja laitteet

Projektiryhmälle varatut tilat ja laitteet toteutuivat suunnitellusti. Ryhmän käyttöön oli annettu työhuone Ag C226.4 Jyväskylän yliopiston Agora-rakennuksesta. Lisäksi ryhmän varattavissa oli Agorassa sijaitseva tietotekniikan sovellusprojektin kokoushuone Ag C418.1, jota pääasiassa käytettiin palaveri-tilaajan kanssa. Projektin kaksi ensimmäistä palaveria pidettiin tilaajan tiloissa ajosimulaatiolaboratoriossa VS3. Projektiryhmän käytössä oli myös tietotekniikan projektiopintojen yhteistila Ag C225.2.

Työhuone oli varustettu viidellä pöytämallisella tietokoneella, joista kaikkiin oli liitetty kaksi näyttöä. Kaikkiin työasemiin oli asennettu Microsoft Windows 10 -käyttöjärjestelmä. Työpisteisiin oli asennettu tarvittavat kehitysympäristöt ja ohjelmistot. Projektiryhmän käytössä oli kaksi verkkolevyä, joilla voitiin säilyttää projektiin liittyviä dokumentteja. Projektiryhmän varattavissa oli digitaalisanelin palaverien nauhoittamiseen, mutta sitä ei käytetty.

4.3 Kehitys- ja dokumentointityökalut

Kehitys- ja dokumentointityökaluja käytettiin suunnitellusti. Sovelluskehitykseen valittiin pääasialliseksi ohjelmointikieleksi C# ja ohjelmointiympäristöksi Visual Studio 2017. Graafinen käyttöliittymä kehitettiin WPF-kirjastolla (Windows Presentation Foundation). Sovellus syötetietojen keräämiseen Android-laitteilta kehitettiin käyttäen Xamarin-kehitystyökalua. Versiohallintaan käytettiin YouSource-versiohallintajärjestelmää.

Dokumentointiin käytettiin Microsoftin Office-perheen ohjelmistoja ja LaTeX-ladontajärjestelmää MiKTeX-ohjelmistolla. Osa kaavioista tehtiin FreeMind-ohjelmistolla sekä osa Umllet- ja PlantUML -ohjelmalla. Työajanseurantaan käytettiin Petri Heinosen kehittämää Excel-työkirjaa. Työvaiheiden aikataulu laadittiin GanttProject-sovelluksella.

4.4 Luennot ja perehdytykset

Kaikki projektiryhmän jäsenet suorittivat sovellusprojektin ohessa suositeltavat oheiskurssit suunnitellusti. *Projektiviestintä IT-alalla* -kurssi (3 op) piti sisällään seuraavat kokonaisuudet:

- vuorovaikutusosaaminen, ryhmäilmiöt, yhteistyötaidot ja ryhmän tehokkuus,
- projektiviestinnän tavoitteet ja muodot,
- kirjoittamisprosessi,
- palautetaidot sekä
- viestinnän ymmärrettävyys ja tilanteenmukaisuus.

Lisäksi kurssi *Sovellusprojektin hallintaa, viestintää ja työkaluja* (1 op) sisälsi seuraavat luennot:

- aloitusluento,
- vaatimusmäärittely,
- versiohallinta,

- projektin hallinta ja johtaminen,
- käytettävyyshuolto sekä
- tekijänoikeushuolto.

Oheiskursseihin käytettävä aika kirjattiin sovellusprojektin työajanseurantaan omaan tehtäväkokonaisuuteen.

5 Käytänteet

Luvussa kuvataan projektin käytänteitä. Käytänteiden noudattaminen edesauttoi asetettujen tavoitteiden saavuttamista, tulosten kehittämistä ja projektin hallittua läpivientä aikataulun mukaisesti. Käytänteet toteutuivat suunnitellusti.

5.1 Palaverit

Projektiorganisaatio piti helmikuussa yhteisen palaverin joka viikko ja muina kuukausina joka toinen viikko. Palavereihin kutsuttiin kaikki projektiryhmän jäsenet, ohjaajat ja tilaajan edustajat. Projektiorganisaation palaverien lisäksi projektiryhmä piti sisäisiä palavereja viikoittain.

Jotta projektiorganisaation palaveri oli laillinen ja päätösvaltainen, tuli projektiryhmän toimittaa sen esityslista projektiorganisaatiolle vähintään vuorokausi ennen palaveria. Päätösvaltaisessa palaverissa tuli olla paikalla vähintään yksi projektiryhmän edustaja, yksi tilaajan edustaja ja vastaava ohjaaja.

Palavereissa käytiin läpi edellisen palaverin pöytäkirja, projektipäällikön tilakatsaus projektista (lukuun ottamatta kahta ensimmäistä palaveria), seuraavan palaverin ajankohta, jatkotoimenpiteet ja mahdolliset muut asiat. Edellä mainittujen kohtien lisäksi palavereissa keskusteltiin kehitettävän sovelluksen tavoitteista ja toteutusratkaisuista, esiteltiin projektiryhmän tuloksia sekä tehtiin projektin etenemistä koskevia päätöksiä.

Jokaisesta palaverista toimitettiin projektiorganisaatiolle myös pöytäkirja, jonka palaverissa sihteerinä toiminut projektiryhmän jäsen laati. Lisäksi projektiorganisaatiolle toimitettiin palavereissa käsitellyt materiaalit. Puheenjohtaja johti keskustelua ja huolehti palaverin etenemisestä esityslistan mukaisesti. Jokaisen projektiryhmän jäsen toimi vähintään kahdesti sekä puheenjohtajana että sihteerinä. Palaverit toteutuivat suunnitellusti.

5.2 Tiedotus

Tiedotuksen osalta noudatettiin suunniteltuja käytänteitä. Projektin aikana projektiorganisaation tiedotuksesta vastasi ensisijaisesti projektipäällikkö. Projektiryhmän ulkoisessa viestinnässä jäsenet vastasivat omien vastuualueidensa viestinnästä tarvittaville sidosryhmille, kuten Digipalveluihin ja laitevalmistajille. Projektiryhmän sisäisessä viestinnässä jäsenet vastasivat työmäärien, töiden etenemisen ja mahdollisten ongelmien raportoinnista projektipäällikölle.

Projektiorganisaation tiedottamista varten perustettiin kaksi sähköpostilistaa. Sähköpostilistaa `peltihamsteri@korppi.fi` käytettiin projektiryhmän, tilaajan edustajien ja ohjaajien tiedotukseen. Sen sähköpostiarkisto on osoitteessa `https://korppi.jyu.fi/kotka/servlet/list-archive/peltihamsteri`. Lisäksi projektiryhmän jäsenten ja ohjaajien käytössä oli sähköpostilista `peltihamsteri_opetus@korppi.fi`. Sen sähköpostiarkisto on osoitteessa `https://korppi.jyu.fi/kotka/servlet/list-archive/peltihamsteri_opetus`.

Projektiryhmän sisäinen tiedotus tapahtui suullisen viestinnän lisäksi Slack- ja Whatsapp-pikaviestimillä.

5.3 Tiedostojen nimeäminen

Tiedostojen nimeäminen toteutui suunnitellusti. Lähdekooditiedostojen nimeämisessä käytettiin C#-ohjelmointikielen yleisiä käytänteitä. Tiedostojen nimet ja hakemistojen nimet kirjoitettiin englanniksi.

Dokumenttitiedostot nimettiin sisältöä kuvaavilla nimillä ja dokumentin kielen mukaisesti. Kussakin tiedostonimessä on ensiksi projektin nimi ja sen jälkeen dokumentin nimi. Jos nimi koostuu useammasta kuin yhdestä sanasta, käytettiin välimerkinä alaviivaa.

5.4 Hakemistorakenne

Projektiryhmän laatimat tulokset tallennettiin sekä CD-levylle että julkiseen WWW-hakemistoon seuraavan hakemistorakenteen mukaisesti:

```
sovellus
  lähdekoodi
  asennuspaketti
dokumentit
  ajankaytto
  esittelyt
  itsearviointit
  lisenssisitoumus
  ohjeet
  projektiraportti
  projektisuunnitelma
  rakennekuvaus
  sovellusraportti
  vaatimukset
palaverit
  esityslista
  poytakirjat
  tilakatsaukset
sahkopostiarkistot
  peltihamsteri
  peltihamsteri_opetus
testaus
  testausraportit
  testaussuunnitelmat
```

Hakemistorakenne toteutui pääosin suunniteltuna. Alun perin oli tarkoitus olla suunnitteludokumentit-osio, mutta sen nimeksi vaihdettiin rakennekuvaus, joka kuvaa paremmin osiossa olevia kaavioita.

5.5 Lähdekoodi

Sovelluksen lähdekoodi kirjoitettiin ja kommentoitiin englanniksi. Aliohjelmat, luokat ja muuttujat nimettiin kuvaavilla englanninkielisillä nimillä C# -kielen käytänteiden mukaisesti [11]. Lähdekoodi toteutui suunnitellusti.

5.6 Testaus

Suunnitelman mukaisesti ohjelmoija suoritti tarvittavat yksikkötestaukset kirjoittamalleen lähdekoodille. Ajanpuutteen vuoksi yksikkötestaus ei ollut aivan niin kattavaa kuin olisi hyvä olla, mutta oleellisimmat asiat tuli testattua. Sovelluksen käyttöliittymää testattiin kehityksen aikana koko ajan jäsenten toimesta, jotta varmistuttiin sen toiminnasta ja laadusta. Toteutetun sovelluksen laadunvarmistus tapahtui järjestelmätestauksella, joka suoritettiin kolme kertaa. Projektin loppupuolella järjestettiin hyväksymistestaus tilaajan kanssa, jolla varmistettiin vaatimusmäärittelyssä asetettujen vaatimusten täyttyminen.

Järjestelmätestauksesta vastaava henkilö laati testaussuunnitelman sekä vastasi testauksen läpiviennistä. Vastaavan henkilön lisäksi muut projektiryhmän jäsenet olivat mukana järjestelmätestauksen suorittamisessa. Testaussuunnitelma sisälsi testauskerroilla suoritettavat testitapaukset kuvattuina vaihe vaiheelta. Järjestelmää testattiin eri laiteyhdistelmillä ja keräten eri dataosioiden dataa laitteilta. Testauskerran suorittaneen jäsenen laatimassa testausraportissa kuvattiin testauskerran havainnot sekä mahdolliset havaitut virhetilanteet ja puutteet.

Hyväksymistestaukseen osallistui hyväksymistestauksen vastuuhenkilön ja neljän ryhmän jäsenen lisäksi tilaajan edustajat, jotka ovat sovelluksen pääkäyttäjät. Hyväksymistestaus suoritettiin tilaajan tiloissa ajosimulaatiolaboratoriossa. Hyväksymistestauksen vastuuhenkilö laati testaussuunnitelman yhdessä tilaajan edustajan kanssa ja testauskerran suorittaneet tilaajan edustajat laativat

raportin testauksen tuloksista. Suunnitelma sisälsi tilaajien toimesta suoritettavat toimenpiteet. Testausraportti kuvasi tilaajien huomiot ja arviot vaatimusten täyttymisestä sekä testauksen johtopäätöksen.

Järjestelmätestaus toteutui suunnitellusti muutoin, paitsi että testauskerta päättyi tekemään kolme kertaa suunnitellun kahden sijaan. Kahdessa ensimmäisessä järjestelmätestauksessa sovelluksesta ilmeni korjattavaa sen verran, että kolmas testauskerta oli tarpeen. Myös hyväksymistestauksessa oli pieni ero suunnitellun ja toteuman välillä. Alun perin testauskerralla piti olla paikalla tilaajan edustajien lisäksi vain testauskerran valvojan, mutta lopulta paikalla oli suurin osa ryhmästä. Muut ryhmän jäsenet toimivat teknisenä apuna, sillä sovellukselle ei ollut vielä käyttöohjetta testauskertaa suoritettaessa. Alkuperäisessä suunnitelmassa ei ollut otettu huomioon koehenkilön tarvetta, joten osa ryhmän jäsenistä toimi myös koehenkilöinä.

5.7 Versiohallinta ja -numerointi

Versiohallinnan ja -numeroinnin osalta noudatettiin suunniteltuja käytänteitä. Sovelluksen lähdekoodin versiohallintaan käytettiin Git-versiohallintaohjelmistoa. Lähdekoodi sijoitettiin YouSource-palveluun, jossa se oli projektiryhmän ja ohjaajien sekä tilaajan edustajien nähtävissä projektin ajan.

Julkistettavissa dokumenteissa käytettiin kolmiportaista numerointia (i.j.k). Projektiryhmän sisäisiä versioita numeroitiin pienimmällä merkitsevällä numerolla paikassa k, kuten i.j.1. Projektioorganisaatiolle julkistettavat versiot numeroitiin toisella merkitsevällä numerolla paikassa j, kuten i.1.k. Ensimmäisen koko projektioorganisaation hyväksymän dokumentin versio oli 1.0.0, ja tätä seuraavien hyväksytyjen versioiden numerointia kasvatettiin toisen merkitsevän numeron osalta.

5.8 Katselmoinnit ja tulosten hyväksyminen

Katselmointien ja tulosten hyväksymisen osalta noudatettiin suunniteltuja käytänteitä. Projektin aikana sovelluksen lähdekoodi katselmoitiin teknisen ohjaajan toimesta kaksi kertaa. Katselmoinneissa ohjaaja antoi ehdotuksia lähdekoodin parantamiseen. Katselmointiin osallistui teknisen ohjaajan lisäksi koko projektiryhmä. Katselmoinneissa kiinnitettiin huomiota nimeämiskäytänteisiin ja muihin kosmeettisiin seikkoihin. Koodissa ei ollut niin isoja ongelmia, että olisi ollut tarpeen kirjoittaa muistiota tai raporttia katselmoinneista.

Tekninen ohjaaja hyväksyi lähdekoodin viimeisen katselmoinnin yhteydessä. Vastaava ohjaaja ja tilaajan edustajat hyväksyivät keskeiset laaditut dokumentit, joita olivat projektisuunnitelma, projektiraportti, vaatimukset, käyttöohjeet ja sovellusraportti.

5.9 Tulosten koostaminen ja toimittaminen

Projektiryhmä kokosi luvussa 3.6 kuvatut projektin tulokset sekä tulostettuina projektikansioon että tallennettuina CD-levylle luvussa 5.4 kuvatun hakemistorakenteen mukaisesti. Tulokset toimitettiin tilaajalle CD-levyllä ja usb-tikulla. Informaatioteknologian tiedekunnalle toimitettiin tuloksista koostettu projektikansio ja CD-levy. Tiedekunnan arkistoon toimitettiin lisäksi yksi CD-levy.

6 Roolit, vastualueet ja tehtävät

Luvussa määritellään projektiryhmän jäsenten roolit, vastualueet ja tehtävät. Lisäksi esitellään projektin tehtävien työmääriä ja tehtävien jakautumista projektiryhmän jäsenille verrattuna suunniteltuun. Tehtävien jakautumisessa oli vain pieniä eroja suunniteltuun, mutta koko ryhmän työmäärä ylitti suunnitellut tuntimäärät 592:lla tunnilla. Työmäärien ylittyminen tarkoitti, että ryhmän jäsenet tekivät suunniteltua enemmän työtunteja viikkoa kohti projektin loppuvaiheessa ja projekti venyi n. neljällä viikolla suunnitellusta.

6.1 Roolit ja vastualueet

Projektipäällikkönä toimi Marina Mustonen. Projektipäällikön vastuulla oli projektin suunnittelu, hallinta ja raportointi sekä sisäinen että ulkoinen viestintä. Projektipäällikön vastuulla oli myös tehtävien jakaminen projektiryhmän jäsenille ja projektin etenemisen valvominen. Projektipäällikön ollessa poissa varapäällikkö Kasanen hoiti projektipäällikön tehtäviä. Kasanen mm. laati ja julkisti tilakatsauksen projektiorganisaatiolle yhden kerran maaliskuun lopulla ja kolme kertaa projektin lopulla.

Taulukossa 1 on määriteltynä jäsenten toteutuneet vastualueet projektin tuloksiin liittyen. Vastuuhenkilö vastasi pääosin kyseisen tuloksen saattamisesta valmiiksi, sekä siihen liittyvästä viestinnästä projektiorganisaatioon kuuluvien henkilöiden ja sidosryhmien kanssa. Vastualueen vastuuhenkilö ei välttämättä toteuttanut kyseistä vastuualuetta yksin.

Vastuualue	Vastuuhenkilö
Projektisuunnitelma	Marina Mustonen
Projektiraportti	Marina Mustonen
Vaatimusmäärittely	Mari Kasanen
Ajosimulaattori	Arttu Ylä-Sahra
EEG	Mari Kasanen
Silmänliikekamera	Juhani Sundell
Toissijaisen tehtävän laite (Android)	Leevi Liimatainen
Datan keruukomponentit	Juhani Sundell, Arttu Ylä-Sahra
Datan synkronointikomponentit	Mari Kasanen, Juhani Sundell, Arttu Ylä-Sahra
Käyttöliittymä	Mari Kasanen, Leevi Liimatainen
Järjestelmätestaus	Arttu Ylä-Sahra
Hyväksymistestaus	Marina Mustonen
Sovellusraportti	Juhani Sundell
Ohjeet	Mari Kasanen, Leevi Liimatainen
Yhteenvetosivu	Mari Kasanen

Taulukko 1: Toteutuneet vastualueet ja vastuuhenkilöt.

Tehtävien jakautumisessa oli pieniä eroja suunnitellusta. Liimataisen piti olla päävastuussa käyttöliittymän kehittämisestä, mutta toissijaisen tehtävän laitteen moduulin kehittäminen osoittautui odotettua haasteellisemmaksi ja Liimataisen aika kului hyvin pitkälti kyseisen vastualueen parissa. Kasanen kehitti suurimman osan käyttöliittymästä. Kasanen oli suunnitelman mukaan päävastuussa datan synkronointiin liittyvistä komponenteista, mutta pääosin kyseiset komponentit kehittivät Sundell ja Ylä-Sahra. Liimatainen oli päävastuussa ohjeista ja hän laati sovelluksen käyttöohjeen, mutta Kasanen kirjoitti teknisen ohjeen. Yhteenvetosivun piti olla Mustosen vastuulla, mutta hänen sairastuttuaan projektin lopulla Kasanen otti tehtävän vastuulleen. Uusia vastualueita ei tarvinnut sopia projektin aikana.

6.2 Tehtävät ja niiden työmäärät

Tehtävien työmäärät suunniteltiin aikaisempien projektien ja tämän projektin ensimmäisten seitsemän viikon toteutuneiden tuntien pohjalta. Tehtävät pyrittiin jakamaan tasaisesti ryhmän jäsenten kesken. Työtunnit kirjattiin työajan seuranta-aulukkoon [12].

Kokonaisuudessaan työtunteja käytettiin suunniteltua enemmän noin viiden työviikon työmäärän verran. Koko ryhmän suunniteltiin käyttävän sovellusprojektiin yhteensä 1350 tuntia, mutta toteutunut tuntimäärä oli n. 1942 tuntia. Erityisesti sovelluksen suunnitteluun, mutta myös toteutukseen sekä järjestelmä- ja hyväksymistestauksiin meni työtunteja aiottua enemmän. Erot suunniteltujen ja toteutuneiden tuntien välillä johtuvat pääosin projektin laajuudesta ja sovellukseen liittyvien laitteiden tuntemattomuudesta. Taulukossa 2 esitetään eri tehtäväkokonaisuuksien suunnitellut (S) ja toteutuneet (T) tuntimäärät.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Projektin hallinta	11	33	10	15	117	152	10	10	10	27	158	237
Palaverit	31	34	31	26	39	32	31	33	31	29	163	154
Esitutkimus	45	30	45	17	25	40	55	47	40	31	210	165
Vaatimusmäärittely	15	32	2	1	2	3	2	2	2	2	23	40
Suunnittelu	67	145	75	150	42	95	77	138	68	147	329	675
Toteutus	68	59	69	115	15	12	35	90	55	55	242	331
Järjestelmätestaus	12	18	12	18	5	11	12	18	44	57	85	122
Hyväksymistestaus	1	10	1	8	9	26	1	6	1	12	13	62
Viimeistely	20	21	25	49	16	8	47	60	19	18	127	156
Projektin tunnint yhteensä	270	382	270	399	270	379	270	404	270	378	1350	1942
Oheiskurssit	74	56	74	49	74	55	74	46	74	52	370	258
Yhteensä	344	438	344	448	344	434	344	450	344	430	1720	2200

Taulukko 2: Projektin tehtäväkokonaisuuksien työtunnit.

Projektiryhmän jokaisen jäsenen tavoitteena oli käyttää Sovellusprojektiin 20 työtuntia viikossa, mutta työtunteja käytettiin enemmän. Eniten ja vähiten työtunteja tehneiden ryhmän jäsenten ero tunneissa oli 25 tuntia vastaten reilun viikon työmäärää. Projektin ohella suoritettaviin oheiskursseihin on laskettu

työtunnit erikseen ja niiden tuntimäärät jäivät suunniteltua pienemmiksi. Luvuissa 6.3–6.12 kuvataan eri tehtävien suunniteltuja ja toteutuneita työmääriä tehtäväkokonaisuuksittain ja luvuissa 6.13–6.17 ryhmän jäsenten toteutuneita tunteja tehtäväkokonaisuuksittain.

6.3 Projektinhallinnan työmäärät

Projektin hallintaan oli suunniteltu 158 tuntia ja siihen käytettiin 237 tuntia. Tunteja käytettiin siis suunniteltua enemmän. Kaikkia tulosten viimeistelyyn ja kokoamiseen käytettyjä tunteja ei myös voitu vielä merkitä taulukkoon, sillä kokoamista tehtiin vielä joitain tunteja projektiraportin kirjoittamisen jälkeenkkin. Suunniteltua suuremmat tunnit johtuivat erityisesti projektisuunnitelman ja projektiraportin kirjoittamiseen sekä seurantaan ja hallintaan menneistä tunteista. Dokumenttien kirjoittamisessa meni aikaa erityisesti niiden hiomiseen. Seurantaan ja hallintaan kirjattiin välillä muihin kategorioihin sopimattomia asioita, vaikka ne eivät täysin sopineet myöskään seurantaan ja hallintaan, mikä selittää seurannan ja hallinnan suuren tuntimäärä. Suunniteltua vähemmän käytettiin aikaa tiedotukseen.

Projektin hallinnan työtuntien jakautuminen ryhmän jäsenten kesken kuvastaa hyvin tehtävien selvää jakoa. Selkeästi eniten tunteja projektin hallintaan käytti projektipäällikkö, jolloin muut ryhmän jäsenet pystyivät keskittymään sovelluksen kehittämiseen. Taulukossa 3 esitetään projektin hallinnan suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Projektinhallinta												
Projektisuunnitelma	1	3	1	1	50	68	1	0	1	0	54	72
Projektiraportti	2	3	2	1	32	54	2	3	2	1	40	62
Seuranta ja hallinta	0	10	0	5	15	16	0	1	0	8	15	40
Tiedotus	2	3	2	0	6	5	2	0	2	3	14	11
Tulosten viimeistely ja kokoaminen	6	14	5	8	14	9	5	6	5	15	35	52
Yhteensä	11	33	10	15	117	152	10	10	10	27	158	237

Taulukko 3: Projektin hallinnan työtunnit.

6.4 Palaverien työmäärät

Palavereihin oli suunniteltu 163 tuntia ja niihin käytettiin 154 tuntia. Suunniteltu tuntimäärä toteutui hyvin. Taulukossa 4 esitetään palaverien suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Esityslistat	2	1	2	0	2	2	2	1	2	1	10	5
Palaverit	20	20	20	20	20	16	20	20	20	21	100	97
Pöytäkirjat	9	9	9	6	9	5	9	12	9	7	45	39
Tilakatsaukset	0	4	0	0	8	9	0	0	0	0	8	13
Yhteensä	31	34	31	26	39	32	31	33	31	29	163	154

Taulukko 4: Palaverien työtunnit.

6.5 Esitutkimuksen työmäärät

Esitutkimukseen arvioitiin 210 tuntia, mutta käytettiin 165 tuntia. Tuntimäärä jäi alle suunnitellun kaikissa esitutkimuksen tehtävissä. Monet työkalut olivat tuttuja ryhmän jäsenille jo entuudestaan. Työkaluihin ja laitteisiin tutustumista tehtiin myös kehitettäessä sovellusta, joten ne eivät välttämättä tulleet aina kirjattua erillisinä. Aihealueeseen tutustuttiin myös paljon palavereissa tilaajan edustajien kanssa keskustellessa ja osin myös tiimitapaamisissa. Taulukossa 5 esitetään esitutkimuksen suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Aihealueeseen tutustuminen	15	7	15	9	15	28	15	4	15	10	75	58
Laitteisiin tutustuminen	20	19	20	3	5	5	30	39	15	15	90	81
Työkaluihin tutustuminen	10	4	10	5	5	7	10	4	10	6	45	26
Yhteensä	45	30	45	17	25	40	55	47	40	31	210	165

Taulukko 5: Esitutkimuksen työtunnit.

6.6 Vaatimusmäärittelyn työmäärät

Vaatimusmäärittelyyn arvioitiin kuluvaan 23 tuntia, mutta käytettiin 40 tuntia. Vaatimusmäärittelyn suunnittelun tunnit jäivät alle aiotun, mutta raportoinnin tunnit olivat melkein kolme kertaa suunnitellut tunnit. Ero johtuu vaatimusmäärittelydokumentin kieli- ja ulkoasun sekä rakenteen vaaditusta tarkkuudesta, mikä johti useisiin korjailuihin ja tarkennuksiin. Taulukossa 6 on vaatimusmäärittelyn suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Suunnittelu	5	4	2	1	2	2	2	1	2	1	13	9
Raportointi	10	28	0	0	0	1	0	1	0	1	10	31
Yhteensä	15	32	2	1	2	3	2	2	2	2	23	40

Taulukko 6: Vaatimusmäärittelyn työtunnit.

6.7 Suunnittelun työmäärät

Sovelluksen suunnitteluun arvioitiin kuluvaan 329 tuntia, mutta käytettiin 675 tuntia. Yli kaksinkertainen ero suunniteltujen ja toteutuneiden tuntien välillä selittyy sillä, että sovellus oli uusi ja vaati alussa prototyyppien kehittelyä, mikä kirjattiin suunnittelun alle. Erityisesti testilaitemoduulin ja datan keräämisen suunnittelu veivät huomattavasti suunniteltua enemmän aikaa. Myös tiimitapaamiset päätettiin kirjata suunnittelun alle, sillä yleensä niissä suunniteltiin sovellusta tai muuta projektin läpivientiin liittyvää. Tiimitapaamisia oli huomattavasti arvioitua enemmän. Taulukossa 7 esitetään suunnittelun suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Suunnittelu	25	77	25	56	25	63	25	56	25	72	125	324
Tiimitapaaminen	1	1	1	0	1	0	1	0	10	9	14	10
Ajosimulaattori-moduuli	10	17	1	0	1	0	1	1	1	2	14	20
EEG-moduuli	1	0	1	4	1	0	30	37	1	0	34	41
Silmänliikekamera-moduuli	1	0	15	72	1	0	1	0	1	1	19	73
Testilaitte (Android) -moduuli	6	10	6	0	4	12	12	43	12	45	40	110
Sovellus (datan keruu)	8	0	6	0	6	14	4	0	15	18	39	32
Sovellus (datan synkronointi)	15	40	20	18	3	6	3	1	3	0	44	65
Käyttöliittymä	67	145	75	150	42	95	77	138	68	147	329	675
Yhteensä	67	145	75	150	42	95	77	138	68	147	329	675

Taulukko 7: Suunnittelun työtunnit.

6.8 Toteutuksen työmäärät

Toteutukseen suunniteltiin 242 tuntia, mutta käytettiin 331 tuntia. Huomattavasti suunniteltua suurempi tuntimäärä selittyy sovelluksen kehittämisen vaativuudella ja erityisesti testilaitemoduulin sekä datan keräämiseen tarkoitettujen komponenttien kehitykseen vaaditulla työmäärällä. Testilaitemoduuli on oma erillinen sovelluksensa ja sen kehittäminen vaati yli kymmenen kertaa suunniteltua enemmän tunteja. Suunniteltua vähemmän tunteja käytettiin erityisesti datan synkronointiin tarkoitettujen komponenttien ja käyttöliittymän kehittämiseen. Taulukossa 8 esitetään toteutuksen suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Toteutus	1	0	1	0	1	0	1	0	10	12	14	12
Ajosimulaattori-moduuli	10	8	1	0	1	3	1	0	1	0	14	11
EEG-moduuli	1	0	1	0	1	0	10	16	1	0	14	16
Silmänliikekamera-moduuli	1	1	10	113	1	1	1	1	1	0	14	116
Testilaitte (Android) -moduuli	6	3	6	2	4	1	12	59	20	24	48	89
Sovellus (datan keruu)	9	1	6	0	6	1	9	12	21	19	51	33
Sovellus (datan synkronointi)	40	46	44	0	1	6	1	2	1	0	87	54
Käyttöliittymä	68	59	69	115	15	12	35	90	55	55	242	331
Yhteensä	68	59	69	115	15	12	35	90	55	55	242	331

Taulukko 8: Toteutuksen työtunnit.

6.9 Järjestelmätestauksen työmäärät

Järjestelmätestaukseen suunniteltiin 85 tuntia, mutta käytettiin 122 tuntia. Ero johtuu järjestelmätestauksen kolmesta testauskerrasta suunnitellun kahden kerran sijaan sekä järjestelmätestausdokumenttien kieli- ja ulkoasun sekä rakenteen vaaditusta tarkkuudesta, joka johti useisiin korjailuihin. Taulukossa 9 esitetään järjestelmätestauksen suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Järjestelmätestaus												
Suunnittelu	2	0	2	0	2	0	2	2	24	40	32	42
Testaus	10	17	10	18	2	11	10	16	10	17	42	79
Raportointi	0	1	0	0	1	0	0	0	10	0	10	1
Yhteensä	12	18	12	18	5	11	12	18	44	57	85	122

Taulukko 9: Järjestelmätestauksen työtunnit.

6.10 Hyväksymistestauksen työmäärät

Hyväksymistestaukseen suunniteltiin 13 tuntia, mutta käytettiin 62 tuntia. Huomattava ero johtuu hyväksymistestaussuunnittelun yllättävästä vaativuudesta sekä laiteongelmista johtuvasta hyväksymistestauksen valmistelun ja testauksen venymisestä. Lisäksi melkein koko ryhmä osallistui hyväksymistestaukseen pelkän hyväksymistestauksen vastuuhenkilön sijaan, kuten oli alun perin suunniteltu. Raportointiin ei tullut tunteja, sillä testauksen suorittaneet tilaajan edustajat raportoivat testauksen tulokset. Taulukossa 10 esitetään hyväksymistestauksen suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Hyväksymistestaus												
Suunnittelu	1	1	1	0	2	18	1	0	1	0	6	19
Testaus	0	9	0	8	3	8	0	6	0	12	3	43
Raportointi	0	0	0	0	4	0	0	0	0	0	4	0
Yhteensä	1	10	1	8	9	26	1	6	1	12	13	62

Taulukko 10: Hyväksymistestauksen työtunnit.

6.11 Viimeistelyn työmäärät

Viimeistelyyn suunniteltiin 127 tuntia, mutta käytettiin 156 tuntia. Toteutunut tuntimäärä siis ylitti suunnitellun tuntimäärän hieman. Erityisesti yli suunnitellun aikaa meni sovellusraportin ja käyttöohjeen kirjoittamiseen. Kehitysympäristön pystytysohjeet ja laitemoduulin lisäämisohjeet päätettiin kirjoittaa tekniseen ohjeeseen, joka kirjattiin käyttöohjeen alle, joten niihin ei tullut ollenkaan tunteja. Taulukossa 11 esitetään viimeistelyn suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Viimeistely												
Sovellusraportti	2	2	2	1	4	4	30	44	2	1	40	52
Lähdekoodin viimeistely	4	3	4	1	1	0	4	11	4	9	17	24
Katselmoinnit	4	4	4	4	4	4	4	5	4	6	20	23
Kehitysympäristön pystytysohjeet	1	0	1	0	0	0	1	0	1	0	4	0
Käyttöohjeet	5	12	10	43	4	0	2	0	2	2	23	57
Laitemoduulin lisäämisohjeet	2	0	2	0	1	0	4	0	4	0	13	0
Sovelluksen luovutus	2	0	2	0	2	0	2	0	2	0	10	0
Yhteensä	20	21	25	49	16	8	47	60	19	18	127	156

Taulukko 11: Viimeistelyn työtunnit.

6.12 Oheiskurssien työmäärät

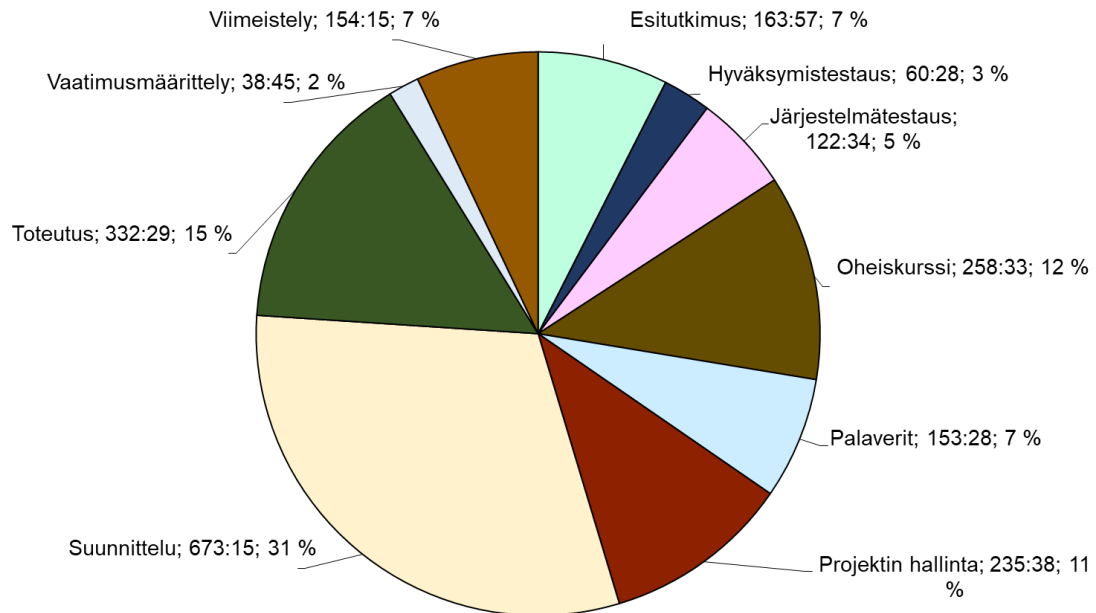
Oheiskursseihin suunniteltiin 370 tuntia, mutta käytettiin n. 258 tuntia. Oheiskursseihin kului odotettua vähemmän tunteja, mutta aiempienkin vuosien projektiryhmät ovat usein varanneet kursseihin tarvetta enemmän tunteja. Erityisesti *Projektiviestintä IT-alalla* -kurssiin meni suunniteltua vähemmän tunteja. Taulukossa 12 esitetään oheiskurssien suunnitellut ja toteutuneet työtunnit tehtävittäin.

Tekijä	MK		LL		MM		JS		AY		Kaikki	
	S	T	S	T	S	T	S	T	S	T	S	T
Oheiskurssit												
Projektiviestintä IT-alalla (XYHI004)	50	31	50	24	50	31	50	28	50	30	250	144
Sovellusprojektin hallinta (TIES412)	24	25	24	25	24	24	24	18	24	22	120	114
Yhteensä	74	56	74	49	74	55	74	46	74	52	370	258

Taulukko 12: Oheiskurssien työtunnit.

6.13 Ryhmän työtunnit tehtäväkokonaisuuksittain

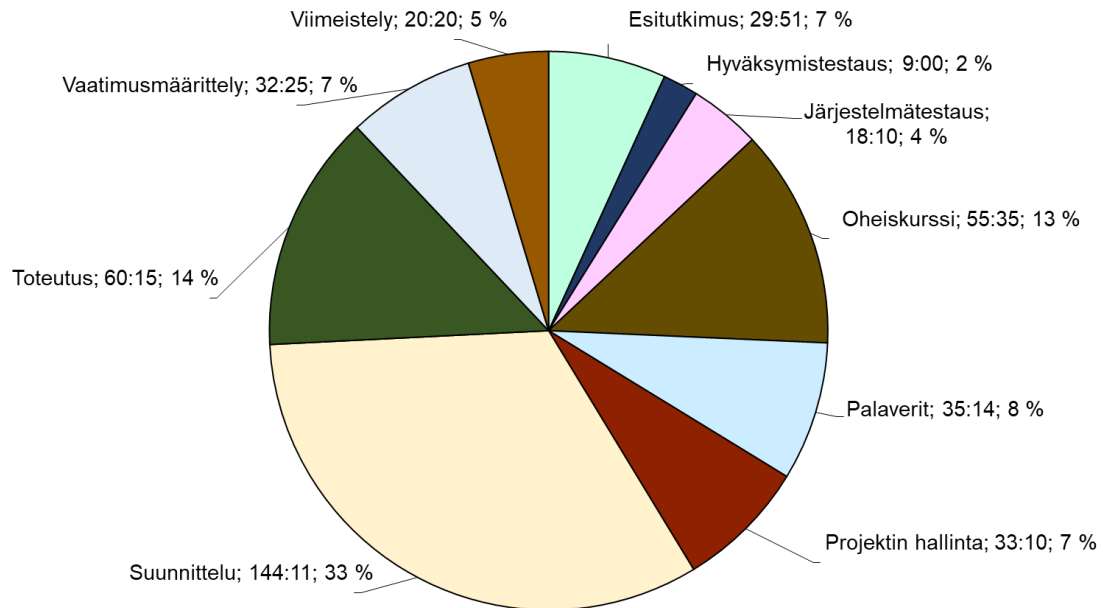
Työtunnit tehtäväkokonaisuuksittain on esitetty kuvassa 1. Toteutuksen osuus työtunneista on 15% ja suunnittelun osuus 31%. Jos varsinaiseen sovelluskehitykseen laskee mukaan vaatimusmäärittelyn, suunnittelun, toteutuksen ja testauksen niin sovelluskehityksen osuus kokonaistuntimäärästä on 56%.



Kuva 1: Ryhmän työtunnit tehtäväkokonaisuuksittain.

6.14 Kasasen työtunnit tehtäväkokonaisuuksittain

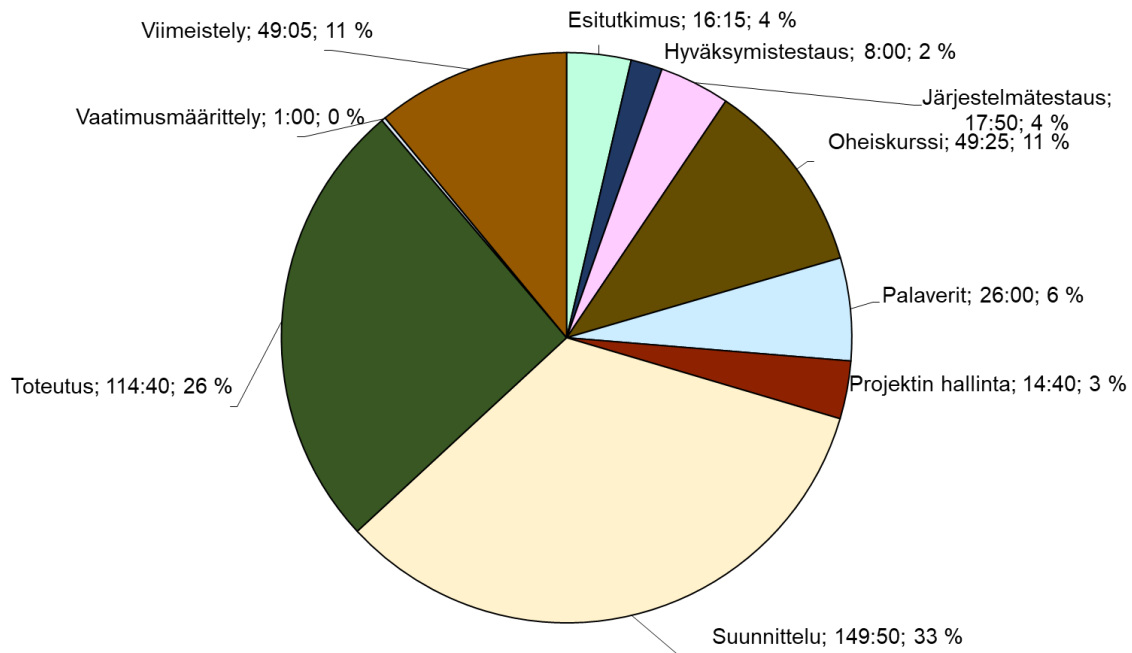
Kasasen työtehtävät painoutuivat käyttöliittymän suunnitteluun ja toteutukseen. Hän teki suurimman osan suunnittelusta ja toteutuksen melkein yksin. Myös EEG-moduuli oli Kasasen vastuulla. Lisäksi Kasanen vastasi vaatimusmäärittelystä ja toimi varapäällikkönä. Kasanen kirjoitti myös teknisen ohjeen.



Kuva 2: Kasasen työtunnit tehtäväkokonaisuuksittain.

6.15 Liimataisen työtunnit tehtäväkokonaisuuksittain

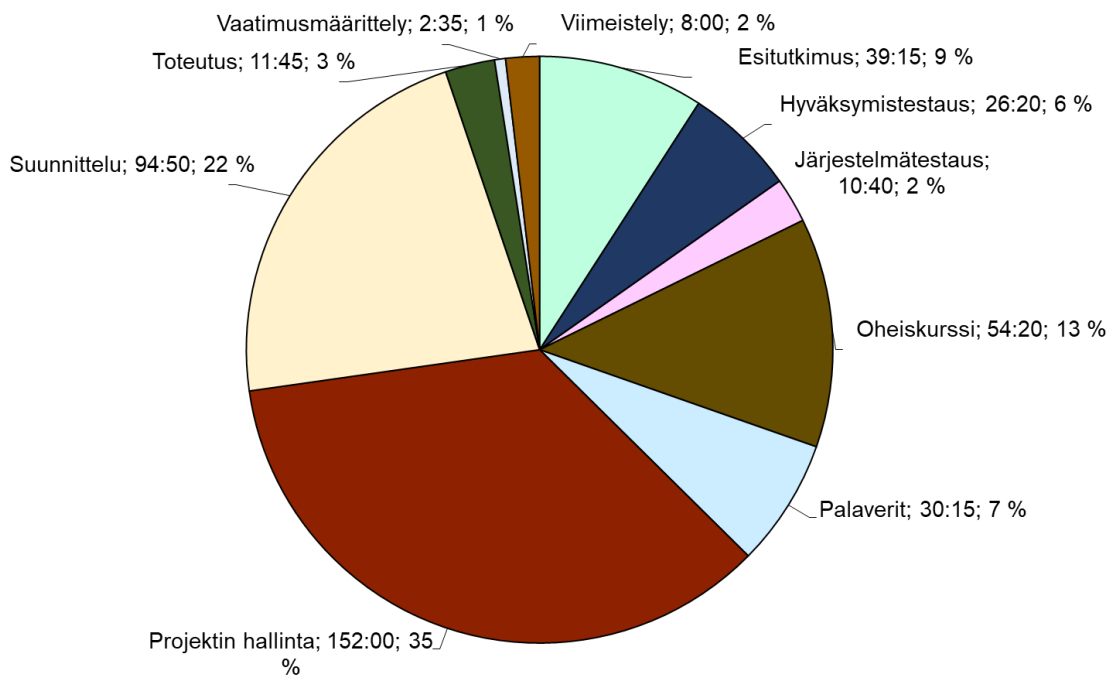
Liimatainen keskittyi melkein yksinomaan testilaitemoduulin kehittämiseen. Kyseessä on erillinen sovellus, joka kerää syötetietoja Android-laitteilta. Liimatainen suunnitteli ja toteutti sovelluksen lähes yksin. Lisäksi Liimataisen vastuulla oli käyttöohjeen kirjoittaminen.



Kuva 3: Liimataisen työtunnit tehtäväkokonaisuuksittain.

6.16 Mustosen työtunnit tehtäväkokonaisuuksittain

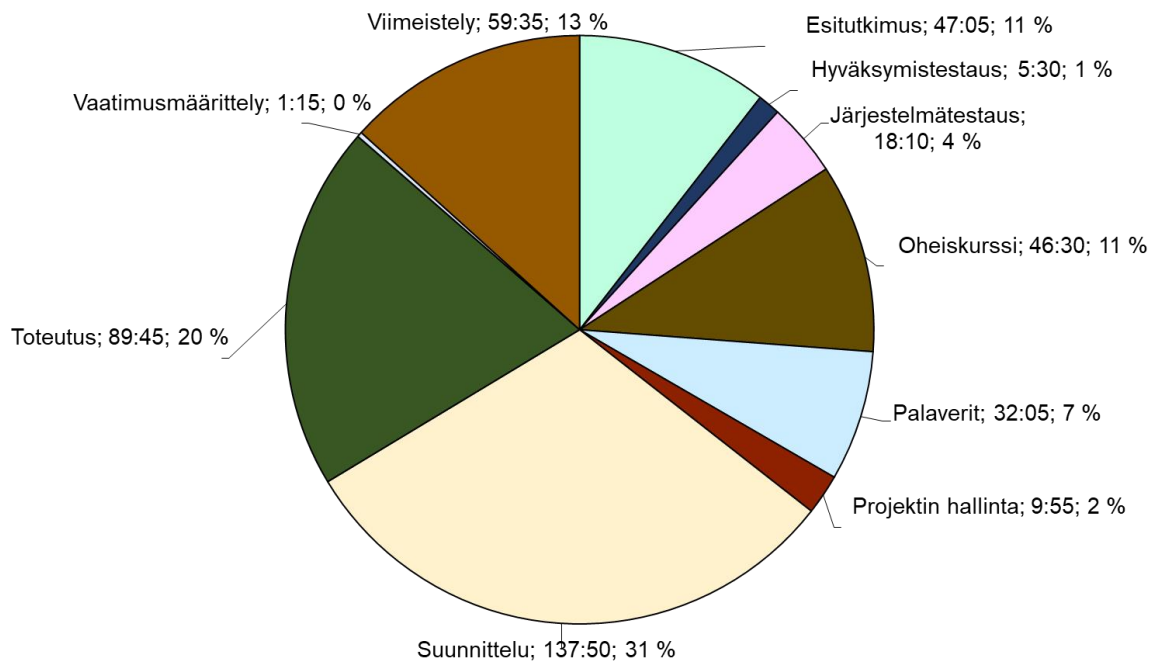
Mustonen toimi projektipäällikkönä, mikä selittää projektin hallinnan suuren osuuden hänen työtunneissaan. Projektin alussa projektisuunnitelman kirjoittaminen sekä loppuvaiheessa projektiraportin kirjoittaminen veivät huomattavan osan työajasta. Ajankäytön seuranta, tiedottaminen sekä tilakatsaukset olivat projektipäällikön vastuulla. Lisäksi Mustonen vastasi hyväksymistestauksesta.



Kuva 4: Mustosen työtunnit tehtäväkokonaisuuksittain.

6.17 Sundellin työtunnit tehtäväkokonaisuuksittain

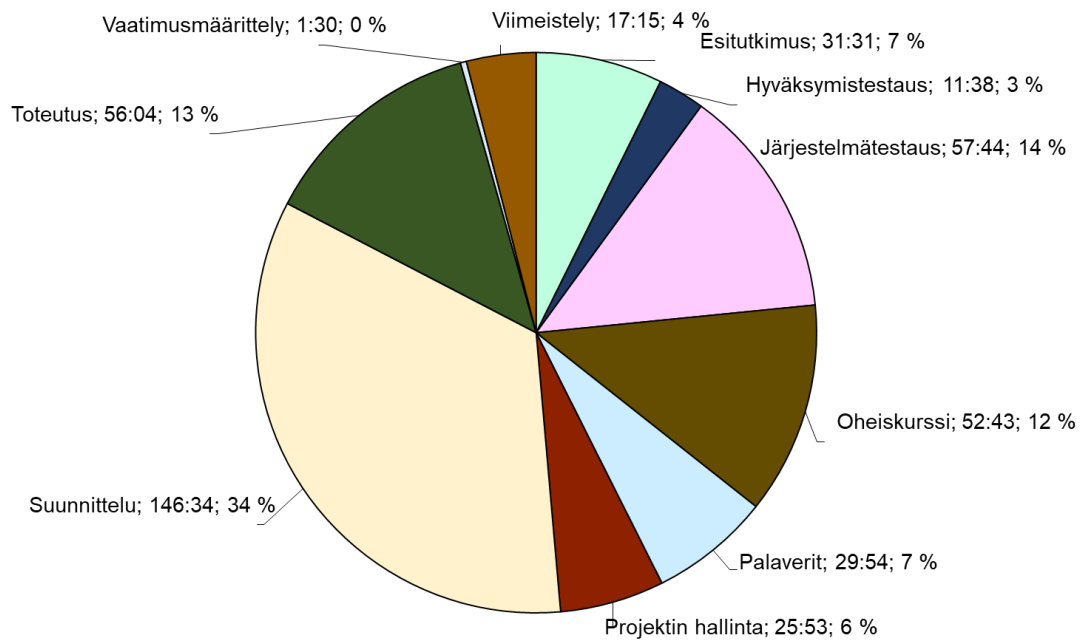
Sundell vastasi sovelluksen toiminnallisuudesta yhdessä Ylä-Sahran kanssa. Hänen työtuntejaan kului datan keräämiseen ja synkronointiin liittyvien komponenttien sekä käyttöliittymän ja laitemoduulien yhteistyöstä vastaavien komponenttien kehittelyyn. Sundellin vastuulla oli myös silmänliikekamera-moduuli. Lisäksi Sundell vastasi sovellusraportin kirjoittamisesta.



Kuva 5: Sundellin työtunnit tehtäväkokonaisuuksittain.

6.18 Ylä-Sahran työtunnit tehtäväkokonaisuuksittain

Ylä-Sahran työtunnit kuluivat sovelluksen toiminnallisuuden kehittelyyn. Hän kehitti datan keräämiseen ja synkronointiin liittyviä komponentteja. Hänen vastuullaan oli myös ajosimulaattorimoduuli. Hän kehitti monet pohjakomponenteista, joita muut ryhmän jäsenet pysyivät hyödyntämään omissa moduuleissaan. Lisäksi järjestelmätestaus oli Ylä-Sahran vastuulla.



Kuva 6: Ylä-Sahran työtunnit tehtäväkokonaisuuksittain.

7 Prosessi ja aikataulu

Luvussa kuvataan sovelluksen kehitystyön tueksi valittua prosessia, joka toteutui muutoin suunnitellusti, mutta yksi sykli venyi viikolla yli suunnitellun. Lisäksi esitellään projektin toteutunutta aikataulua. Projekti venyi suunniteltuun aikatauluun jätetyn kahden viikon pelivaran lisäksi noin kahdella viikolla. Kokonaisuudessaan projekti venyi siis neljällä viikolla. Venyminen johtui projektin vaativuudesta. Laitteet olivat ennestään tuntemattomia ja kehitettävä sovellus uusi, joten tunteja kului erityisesti suunnitteluun ja toteutukseen suunniteltua enemmän. Myös testaukseen meni suunniteltua enemmän tunteja.

7.1 Prosessi

Projekti alkoi suunnitteluvaiheella, jossa kartoitettiin sovelluksen vaatimuksia tilaajan edustajien kanssa keskustellen. Keskustelujen pohjalta laadittiin sovellusta ja projektin läpivientiä kuvaavia määrittely- ja suunnitteludokumentteja.

Sovellusprojektissa noudatettu sovelluksen kehitysprosessi mukaili Boehmin artikkelissa [13] kuvattua spiraalimallia, jota räätälöitiin projektin tarpeisiin sopivalla tavalla. Spiraalimalli on riskien arviointiin pohjautuva iteratiivinen prosessimalli, johon voi ottaa ominaisuuksia muista malleista projektin tarpeista riippuen. Ohjelmiston kehittäminen tapahtuu sykleissä, joissa toistuvat samat askeleet. Alussa syklit ovat nopeita ja pitenevät projektin edetessä. Sykli alkaa ohjelmiston tavoitteiden, vaihtoehtojen ja rajoitusten kartoittamisella. Seuraavaksi tehdään riskianalyysi, jossa kartoitetaan projektiin liittyviä epävarmuuksia ja eri vaihtoehtojen riskejä. Kolmannessa askeleessa voidaan riskianalyysin tulosten ja projektin tarpeiden perusteella tehdä esimerkiksi prototyyppejä, simulaatioita, analyttistä mallinnusta tai näiden yhdistelmiä, sekä projektin

loppupuolella varsinaisen ohjelman kehittelyä. Neljännessä askeleessa arvioidaan aikaansaatuja tuloksia ja suunnitellaan seuraavaa sykliä, joka aloitetaan jälleen tavoitteiden ja vaihtoehtojen kartoittamisella.

Spiraalimalli ei ollut ryhmän jäsenille entuudestaan tuttu, joten sen käyttäminen oli haastavaa. Mallin käytännön noudattaminen ei ollut kovin selkeää, joten mallia päädyttiin käyttämään melko löyhästi. Ryhmä koki kuitenkin hyödylliseksi käydä läpi mallin mukaisia asioita erityisesti syklien alussa, kun kartoitettiin vaihtoehtoja ja niihin liittyviä riskejä.

7.2 Projektin kehitysvaiheet

Peltihamsteri-projektissa aloitettiin vaatimusmäärittelyn laatiminen heti 1.2.2019 järjestetyn ensimmäisen palaverin jälkeen. Vaatimusmäärittely laadittiin tilaajan kanssa käytyjen keskustelujen pohjalta, ja siinä priorisoitiin sovelluksen vaatimukset. Vaatimukset tarkentuivat projektin edetessä, mutta ensimmäinen kaikki toiminnallisuudet kuvaava versio toimitettiin tilaajille ensimmäisten seitsemän viikon aikana.

Ryhmä aloitti ensimmäisen syklin kaksi viikkoa projektin alun jälkeen, jolloin oli saatu alustava käsitys sovelluksen vaatimuksista. Ryhmä aloitti syklin miettimällä vaihtoehtoja, joilla voidaan päästä tavoitteisiin ja täyttää tilaajan asettamat vaatimukset. Ryhmä pohti eri vaihtoehtoja ja niiden rajoitteita. Seuraavaksi ryhmä kartoitti vaihtoehtojen riskit. Kolmanteen askeleeseen sopi kahdessa ensimmäisessä syklissä prototyyppilähtöinen lähestyminen. Kehitettävä sovellus oli uusi (eli kyseessä ei ollut olemassa olevan sovelluksen jatkokehitys), ja kaikkia sovellukseen liitettäviä laitteita ei oltu vielä testattu käytössä, mikä toi epävarmuutta niiden toiminnasta ja ominaisuuksista. Ensimmäisessä syklissä kehitettävillä prototyypeillä kokeiltiin, mikä on toimivaa ja mikä ei. Syklin neljännessä askeleessa arvioitiin kehitettyjä prototyyppejä ja päätettiin,

otetaanko ne jatkokehitykseen seuraavaan sykliin vai hylätäänkö ne. Prototyypit päätettiin ottaa jatkokehitykseen. Ensimmäinen sykli oli luonteeltaan kokeilullinen, joten se pidettiin lyhyenä (vajaa kaksi viikkoa).

Toinen sykli kesti kaksi viikkoa. Se aloitettiin käymällä läpi tarkentuneita vaatimuksia ja pohtimalla vaihtoehtoja. Ryhmä jälleen kartoitti eri vaihtoehtojen riskejä ja kolmannessa askeleessa kehitti prototyyppejä. Neljännessä askeleessa arvioitiin prototyyppien toimivuus ja jatkokehittelypotentiaali. Prototyypit olivat jatkokehityskelpoisia ja otettiin seuraavaan sykliin mukaan.

Ensimmäisessä ja toisessa syklissä keskityttiin kehittämään käyttöliittymän lisäksi datan keräämiseen liittyviä komponentteja. Kolmannessa ja neljännessä kehitettiin myös datan synkronointiin liittyviä komponentteja.

Kolmannessa ja neljännessä syklissä ensimmäisessä askeleessa käytiin läpi ja tarvittaessa tarkennettiin vaatimuksia ja pohdittiin vaihtoehtoja. Riskit kartoitettiin toisessa askeleessa ja kolmannessa askeleessa kehitettiin sovellusta. Neljännessä askeleessa arvioitiin tuloksia, jolloin todettiin toteutetut komponentit pääosin toimiviksi, mutta vielä hiomista kaipaaviksi. Kolmannen syklin päätteeksi oli myös vielä toteuttamattomia komponentteja, jotka toteutettiin neljännessä syklissä. Kolmannen syklin oli tarkoitus kestää kolme viikkoa ja neljännen syklin neljä viikkoa, mutta lopulta sekä kolmas että neljäs sykli kesti neljä viikkoa.

Neljännen syklin päätyttyä koodia vielä hiottiin kaksi viikkoa, jonka jälkeen suoritettiin hyväksymistestaus. Viimeistelyvaiheessa viimeisteltiin sovellus, lähdekoodi ja dokumentit tilaajalle luovutusta varten.

7.3 Ryhmän tilalaverit ja tilakatsaukset

Projektiryhmän sisäisiä palavereja pidettiin vähintään kerran viikossa, mutta yleensä useampi viikossa. Niissä käytiin läpi syklin silloisen askeleen mukaisia asioita ml. tarpeiden ja vaihtoehtojen kartoittaminen, riskianalyysi tai tulosten

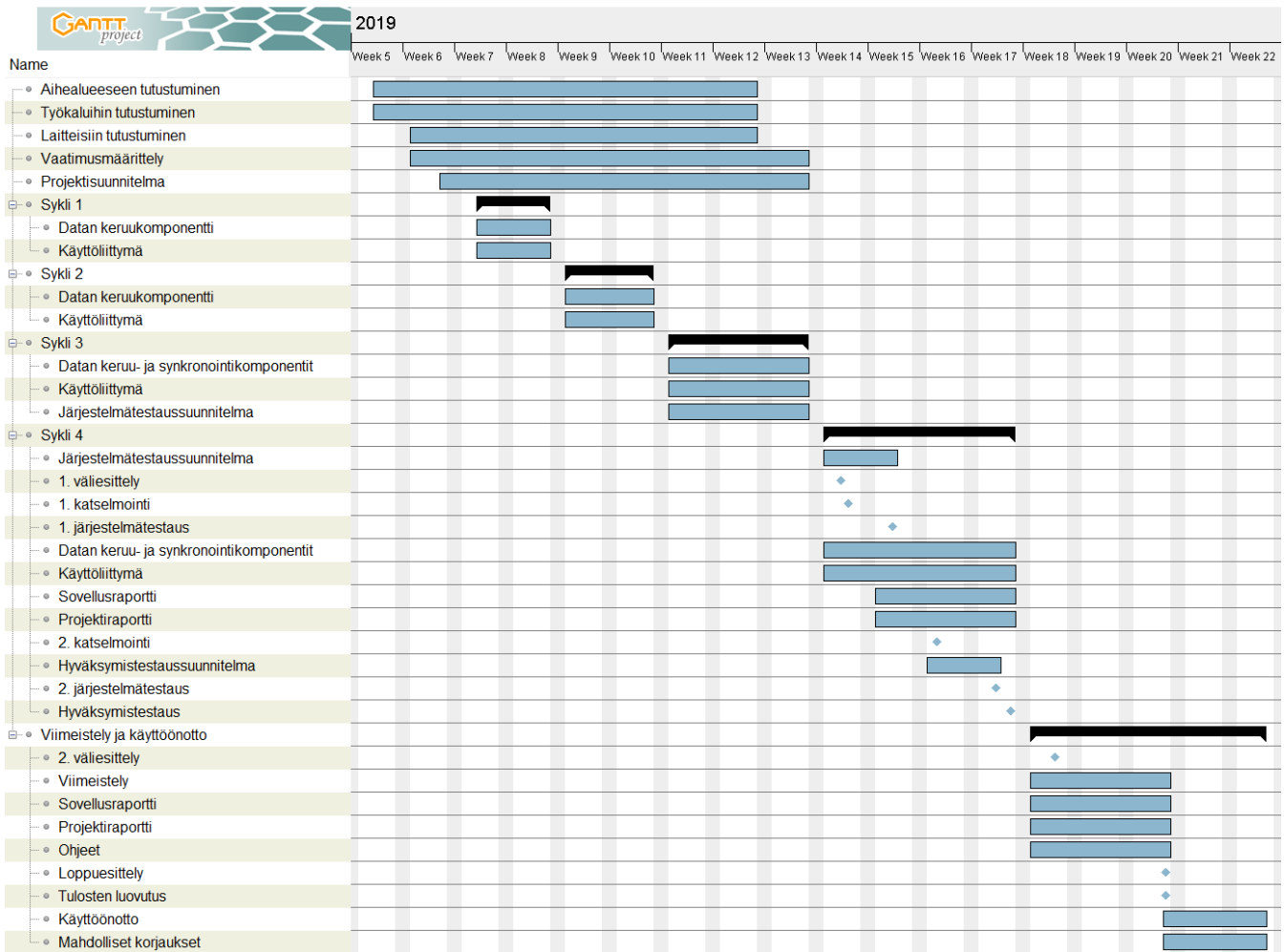
pohdinta ja seuraavan syklin suunnittelu. Lisäksi käsiteltiin muita projektin läpiviennin kannalta oleellisia asioita, kuten tilakatsaukseen tulevat asiat ja sovelluksen arkkitehtuurin suunnittelu.

Projektiorganisaatiolle laadittiin projektin kolmannesta viikosta lähtien tilakatsaus projektin tilasta. Tilakatsauksissa käytiin läpi, mitä oli tehty, mitä tehdään seuraavaksi ja mitä ongelmia oli kohdattu, sekä eri vaiheisiin ja eri jäsenten käyttämät tuntimäärät. Tilakatsaus toimitettiin projektiorganisaatiolle joko palaverissa tai sähköpostitse.

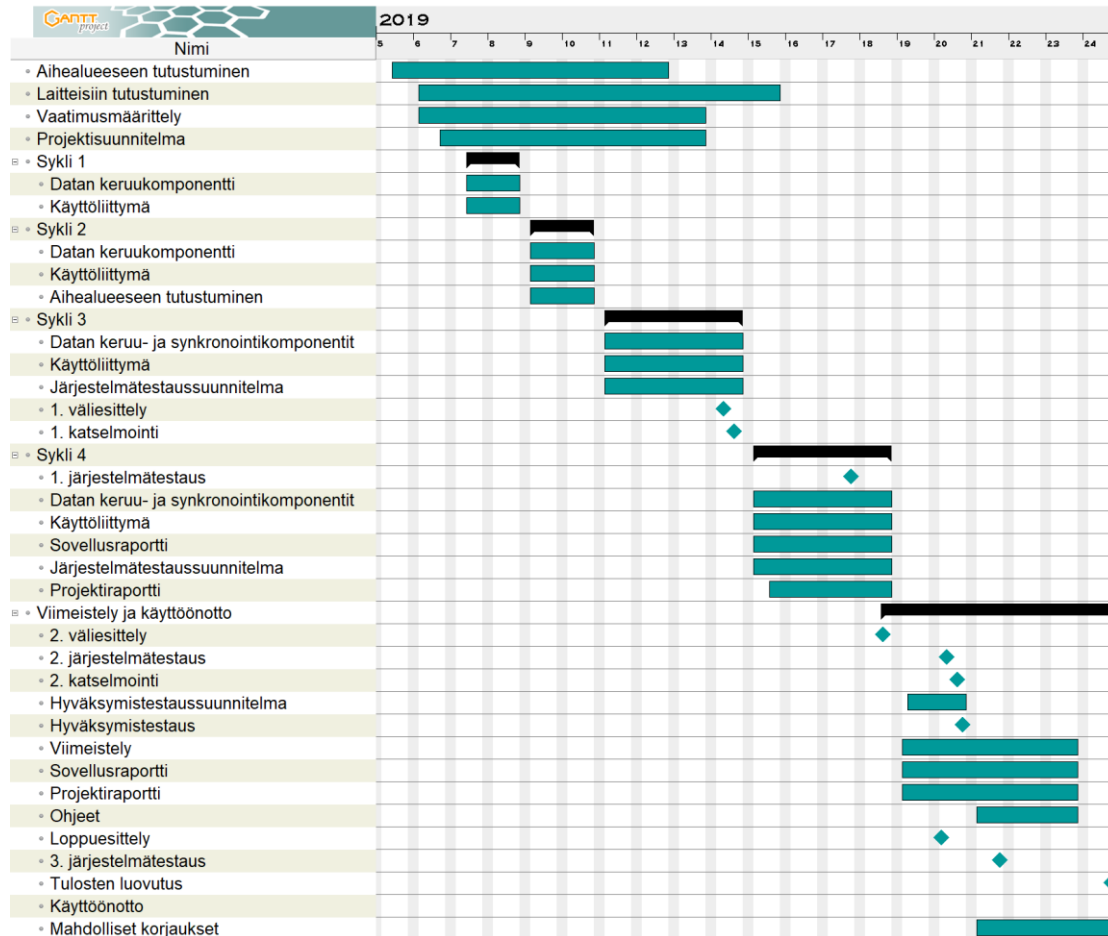
7.4 Aikataulu

Projekti alkoi 29.1.2019 opintojakson aloitusluennolla. Työkaluihin ja aihealueeseen tutustuminen aloitettiin heti ja vaatimusmäärittely 1.2.2019 järjestetyn ensimmäisen palaverin jälkeen.

Sovelluksen kehittäminen tehtiin neljässä syklissä päättyen toukokuun ensimmäisen viikon jälkeen. Toukokuun aikana sovellusta ja projektiin liittyviä dokumentteja viimeisteltiin. Sovellus ja muut tulokset luovutettiin tilaajalle kesäkuun toisella viikolla. Projektin suunniteltu aikataulu esitellään kuvassa 7 ja toteutunut aikataulu esitellään kuvassa 8.



Kuva 7: Projektin suunniteltu aikataulu.

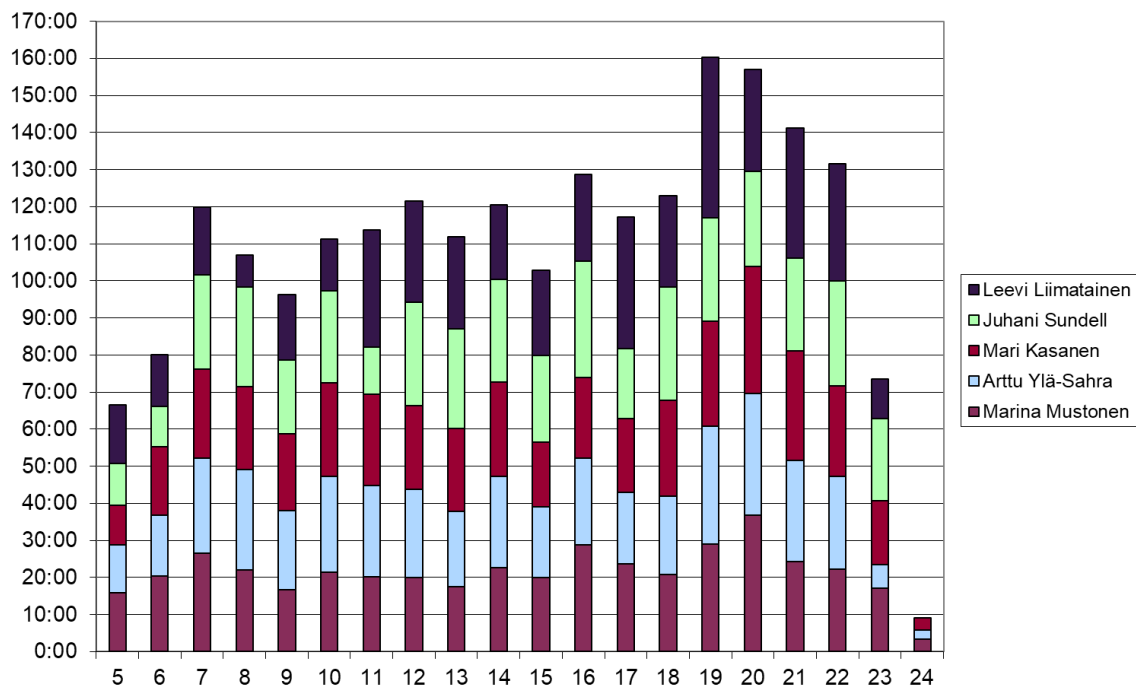


Kuva 8: Projektin toteutunut aikataulu.

Aikataulu ei toteutunut täysin suunnitellusti. Laitteisiin tutustumiseen meni kaksi viikkoa suunniteltua kauemmin. Kolmas sykli oli neljä viikkoa kolmen sijaan, joten neljännen syklin aloittaminen viivästyi viikolla, jolloin myös sen lopettaminen viivästyi viikolla. Sykliä venymisestä johtuen myös viimeistely voitiin aloittaa viikkoa suunniteltua myöhemmin. Sovelluksen viimeistelyyn meni suunniteltua enemmän aikaa, joten molemmat järjestelmätestaukset, hyväksymistestaus ja toinen koodikatselmointi sekä ohjeiden kirjoittamisen aloitus olivat 2–4 viikkoa suunniteltua myöhemmin. Lisäksi tulokset voitiin luovuttaa tilaajalle vasta kesäkuun toisella viikolla toukokuun puolenvälin sijaan.

7.5 Ryhmän työtunnit viikoittain

Ryhmän jäsenet suunnittelivat käyttävänsä projektiin 20 tuntia viikossa, jolloin koko ryhmän viikkotuntimäärä olisi ollut 100 tuntia viikossa. Kaikki ryhmän jäsenet osallistuivat lisäksi oheiskursseille, joihin varattiin n. 5 tuntia per viikko. Oheiskursseille ei kertynyt tunteja joka viikko, mutta joinain viikkoina kertyi enemmän kuin 5 tuntia per viikko. Projektiryhmän viikoittaiset työtunnit on esitelty kuvassa 9.

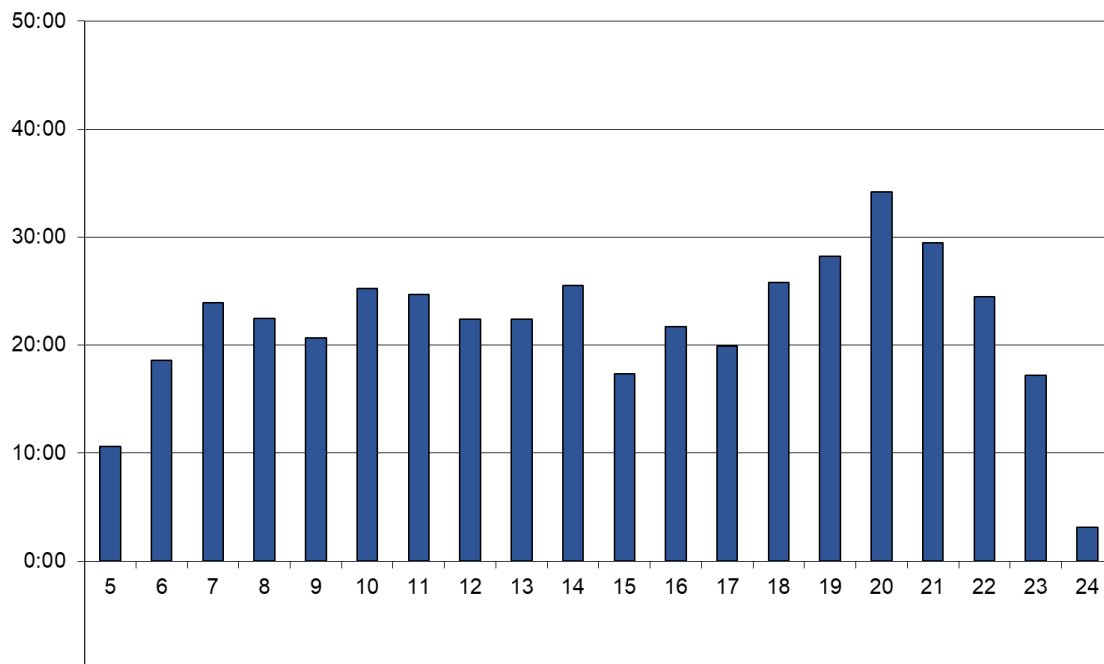


Kuva 9: Projektiryhmän viikoittaiset työtunnit.

Projektissa päästiin tuntitavoitteeseen suurimpana osana viikoista. Kahtena ensimmäisenä viikkona oli perehtymistä ja esitutkimusta sekä kahtena viimeisenä viikkona tehtiin lähinnä tulosten viimeistelyä ja kokoamista, jolloin jäätin tuntitavoitteen alle. Kaikkia viikon 24 tunteja ei vielä raportin kirjoittamisen aikaan ollut myöskään kirjattu. Muutoin tuntitavoitteeseen päästiin viikkoa 9 lukuun ottamatta. Useana viikkona menttiin tuntitavoitteen yli. Työtahti pysyi melko tasaisena läpi projektin, paitsi projektin loppupäässä ryhmän jäsenet tekivät enemmän töitä, jotta sovellus saataisiin valmiiksi ja projekti päätökseen.

7.6 Kasasen työtunnit viikoittain

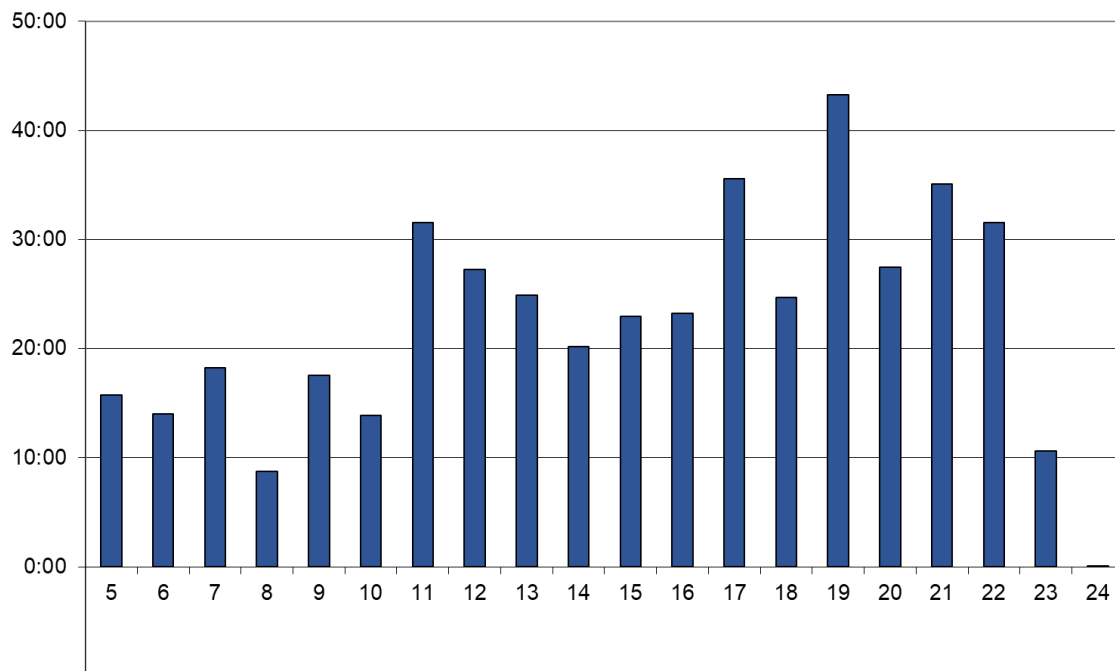
Kasasella pysyi melko tasainen työtahti läpi projektin. Alun kahden viikon pe-rehtymistä ja lopun kahta tulosten kokoamisviikkoa lukuun ottamatta Kasanen jäi tuntitavoitteesta vain viikolla 15, jolloin hän piti yhden vapaapäivän projektipäällikön pyynnöstä tuntimäärien tasaamiseksi. Projektin loppupuolella tun-teja kerrytti erityisesti EEG-moduulin ja käyttöliittymän kehittäminen sekä tek-nisen ohjeen kirjoittaminen.



Kuva 10: Kasasen työtunnit viikoittain.

7.7 Liimataisen työtunnit viikoittain

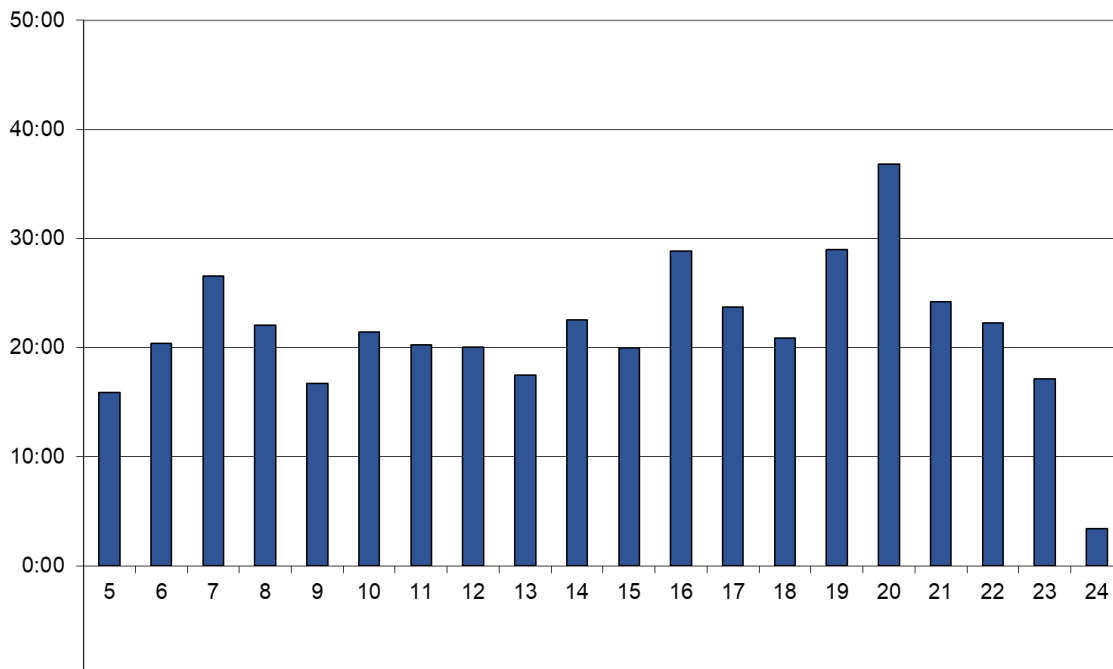
Liimatainen jäi tuntitavoitteesta projektin alkupuolella useampana viikkona johtuen muista samaan aikaan menossa olevista kurseista. Viikosta 11 lähtien hän pääsi joka viikko tavoitteeseen ja useampana viikkona yli tavoitteen, lukuun ottamatta kahta viimeistä tulosten kokoamisviikkoa. Projektin loppupuolella tunteja kertyi erityisesti Touchster-sovelluksen kehittämisestä valmiiksi.



Kuva 11: Liimataisen työtunnit viikoittain.

7.8 Mustosen työtunnit viikoittain

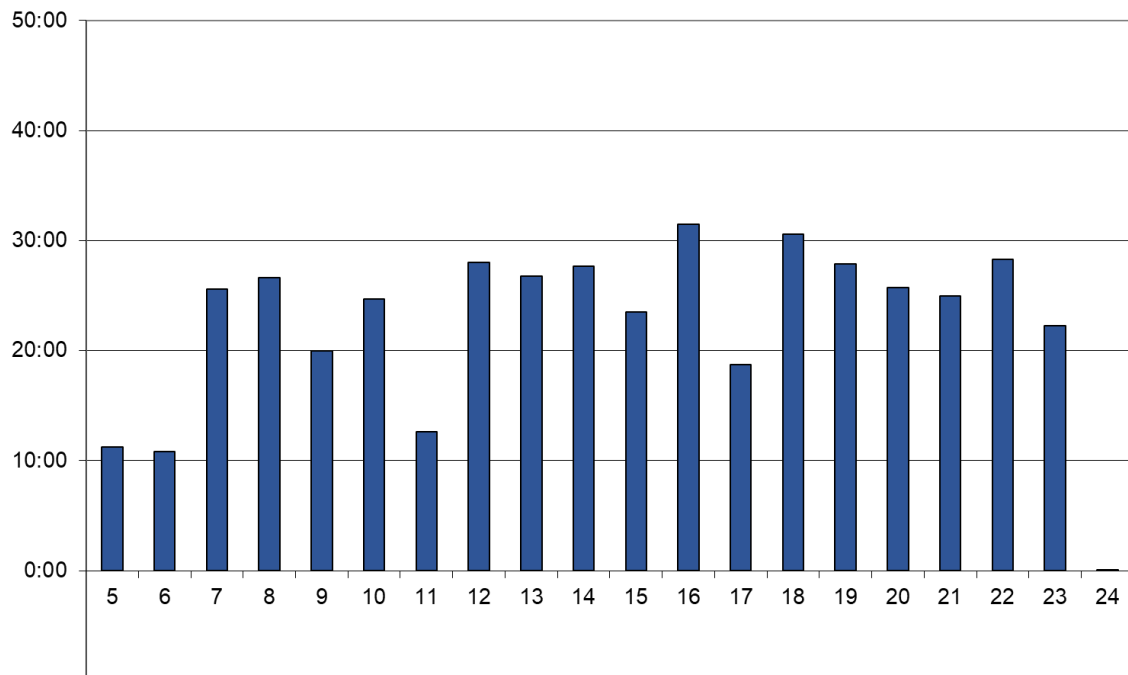
Mustosen työtahti pysyi melko tasaisena. Hän pääsi tuntitavoitteeseen ensimmäistä perehtymisviikkoa, kahta viimeistä tulosten kokoamisviikkoa sekä viikkoja 9 ja 13 lukuun ottamatta. Viikolla 9 oli graduseminaari ja viikolla 13 Mustonen joutui olemaan poissa pari päivää flunssan takia. Projektin loppupuolella tunteja kerrytti erityisesti hyväksymistestauksen suunnittelu ja dokumenttien kirjoittaminen.



Kuva 12: Mustosen tunnit viikoittain.

7.9 Sundellin työtunnit viikoittain

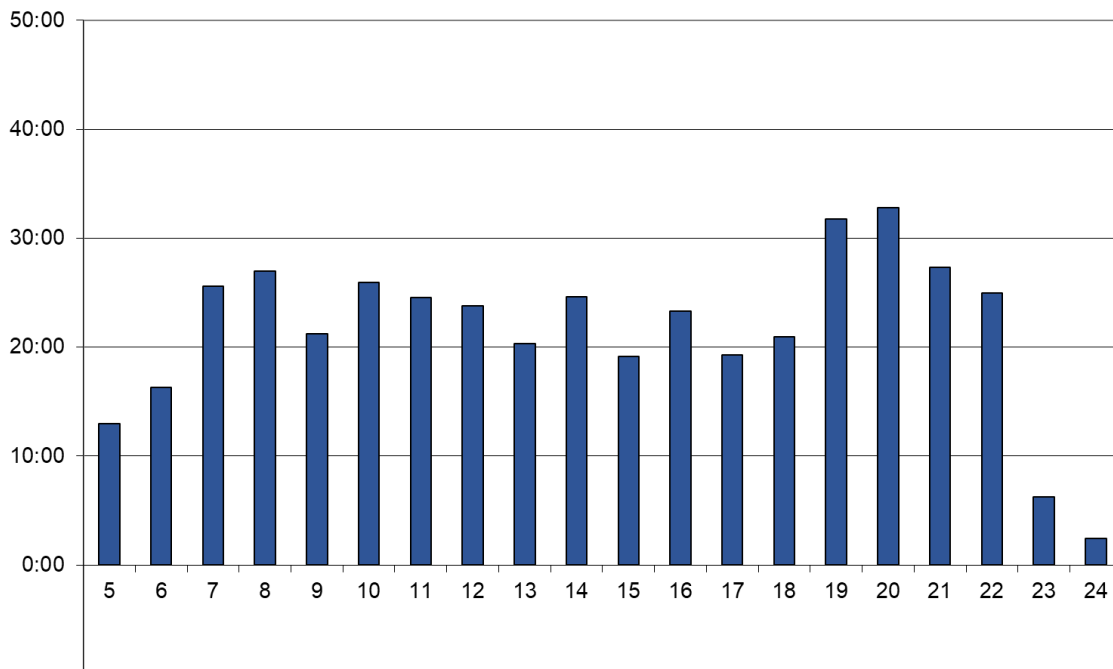
Sundell pääsi tuntitavoitteeseen ja teki useampana viikkona huomattavasti enemmän kuin tavoitteen verran lukuun ottamatta kahta ensimmäistä perehtymisviikkoa, viimeistä tulosten kokoamisviikkoa sekä viikkoja 12 ja 17. Sundellin kokonaistuntimäärä uhkasi kohota reilusti yli muiden ryhmän jäsenten, joten projektipäällikkö pyysi tekemään välillä tuntimäärältään pienempää viikkoa, mikä selittää alle tavoitteen jäävät viikot. Projektin loppupuolella työtunteja kerrytti erityisesti synkronointiin ja datan käsittelyyn liittyvien komponenttien kehittäminen sekä sovellusraportin kirjoittaminen.



Kuva 13: Sundellin työtunnit viikoittain.

7.10 Ylä-Sahran työtunnit viikoittain

Ylä-Sahran työtahti oli melko tasainen läpi projektin. Kahden ensimmäisen perehtymisviikon ja kahden viimeisen tulosten kokoamisviikon lisäksi hän jäi hieman alle tuntitavoitteen viikoilla 15 ja 17. Tämä johtui projektipäällikön pyynnöstä vähentää tunteja, jotta kokonaistuntimäärä ei kasvaisi kovin paljon suuremmaksi kuin muilla ryhmän jäsenillä. Projektin loppupuolella tunteja kertyi erityisesti järjestelmätestauksen suunnittelusta ja koodin viimeistelystä.



Kuva 14: Ylä-Sahran työtunnit viikoittain.

8 Riskien hallinta

Luvussa kuvataan projektin ennakoituja riskejä sekä niiden toteutumista ja vaikutuksia projektin läpivientiin ja tuloksiin. Lisäksi esitellään, miten riskejä ehkäistiin ja miten niistä toivuttiin. Riskien toteutunut vakavuus oli ennakoitua matalampaa. Projekti viivästyi neljällä viikolla pääosin laitteiden tuntemattomuudesta ja tietojen puutteesta johtuen.

8.1 Riskien todennäköisyydet ja haittavaikutukset

Taulukossa 13 on kuvattu riskien ennakoitua todennäköisyyttä ja vakavuutta sekä toteutunutta vakavuutta. Asteikkona todennäköisyyden ja haittavaikutusten arvioinnissa on käytetty kolmiportaista jaottelua: korkea, keskitaso ja matala.

Riski	Todennäköisyys	Ennakoitu vakavuus	Toteutunut vakavuus
Laitteiden erilaisuus ja tietojen puute	Korkea	Korkea	Keskitaso
Sovelluksen joustavuuden toteuttamisen vaikeus	Korkea	Matala	Matala
Ominaisuuksien toteuttamisen vaikeus	Korkea	Matala	Matala
Ryhmän tietotaitojen puutteet	Keskitaso	Keskitaso	Matala
Ryhmän jäsenten muut sitoumukset	Matala	Keskitaso	Matala

Taulukko 13: Projektin riskit.

Riskit eivät oleellisesti vaikuttaneet sovelluksen kehittämiseen. Kaikki tärkeät vaatimukset ja suurin osa muista vaatimuksista saatiin toteutettua. Toteuttamatta jäi vain neljä ideoiksi kirjattua vaatimusta ja yksi valinnainen vaatimus.

Lisäksi oli kaksi vaatimusta, jotka jo projektin alkupäässä sovittiin tilaajan kanssa toteuttamatta jätettäväksi.

Sovelluksen luovuttaminen tilaajalle viivästy kolmella viikolla ja ryhmän jäsenet joutuivat käyttämään suunniteltua enemmän tunteja projektiin. Näihin vaikutti jonkin verran suurin osa ennakoituista riskeistä, mutta luultavasti eniten laitteiden tuntemattomuus ja tietojen puute. Luvuissa 8.2–8.7 esitellään ennakoituja ja toteutuneita riskejä sekä niiden vaikutuksia projektin läpiviintiin ja tuloksiin. Luvuissa esitellään lisäksi käytetyt ehkäisykeinot riskien välttämiseksi ja toipumiskeinot riskistä palautumiseen sen toteuduttua. Ennakoimattomia riskejä ei tullut vastaan.

8.2 Laitteiden erilaisuus ja tietojen puute

Kehitetyllä sovelluksella käsitellään dataa useasta eri laitteesta ml. ajosimulaattori, silmänliikekamera, EEG-laite ja toissijaisessa tehtävässä käytettävä laite. Laitteissa on eri ohjelmistot, eri viiveet datan saamisessa ja pääosin eri tiedostomuodot. Sovelluksella synkronoidaan ja tallennetaan data laitteista yhteen tulostiedostoon. Laitteista uutta silmänliikekameraa ja EEG-laitetta ei oltu vielä käytetty ajosimulaatiokokeissa ollenkaan, joten niiden kaikki ominaisuudet ja käyttömahdollisuudet eivät olleet täysin tilaajien tiedossa.

Eri ohjelmistojen ja tiedostomuotojen yhteensovittaminen ja viiveen huomioiminen synkronisoinnissa sekä epäselvyys laitteiden ominaisuuksista toivat haasteita sovelluksen onnistumiselle ja projektin läpiviennille. Erityisesti toissijaisen tehtävän laitteelle kehitettävä Android-sovellus syötetietojen keräämiseen osoittautui haastavaksi ja aikaa vieväksi. Siihen varattu aika ylittyi reilusti. Myös EEG-laite tuotti hankaluuksia, sillä ryhmän piti tutustua sen ominaisuuksiin alusta lähtien.

Ryhmä minimoi riskiä tutustumalla laitteisiin hyvin ja kehittämällä projektin alussa prototyyppisiä nähdäkseen, mikä on mahdollista ja toimivaa ja mikä ei.

Käytettävien laitteiden ominaisuudet saatiin selville projektin aikana ja niihin saatiin kehitettyä hyväksyttävät ratkaisut, joten riski ei toteutunut sovelluksen kehityksen osalta. Ryhmä joutui kuitenkin alussa ja myös läpi projektin käyttämään paljon aikaa laitteisiin tutustumiseen, mikä vaikutti luultavasti siihen, että sovellusta ei voitu luovuttaa tilaajalle suunniteltuna ajankohtana. Lisäksi työtunteja tuli huomattavasti suunniteltua enemmän.

8.3 Sovelluksen joustavuuden toteuttamisen vaikeus

Tilaaja toivoi kehitettävältä sovellukselta joustavuutta siinä, että uusia laitteita voidaan lisätä sovellukseen myöhemmin. Mahdollisista uusista laitteista ei ollut tietoja, joten niiden ominaisuuksia ja ohjelmistoja ei voitu ottaa huomioon sovelluksen kehittämisessä. Riski vaikutti sovelluksen jatkokehityskelpoisuuteen, joten siitä toipumiseen ei voitu vaikuttaa projektissa.

Ryhmä minimoi riskiä niin, että mahdollisille uusille laitteille toteutettiin valmiita komponentteja nyt tiedossa olevien laitteiden pohjalta. Jatkokehittäjä voi hyödyntää komponentteja uusien laitemoduulien lisäämisessä. Ryhmä kirjoitti laitemoduulien lisäämiseen myös ohjeen, jonka avulla ohjelmointitaitoinen jatkokehittäjä voi lisätä laitemoduuleja.

Riski ei vaikuttanut sovelluksen kehittämiseen ja sen toimintaan tiedossa olevien laitteiden kanssa, mutta jatkokehittämiseen se voi edelleen vaikuttaa. Riskin toteutuminen tulee ilmi vasta projektin päätyttyä.

8.4 Ominaisuuksien toteuttamisen vaikeus

Eryteisesti tilaajan alkuperäisessä aihe-ehdotuksessa sovellukselle oli lueteltu useita haluttuja ominaisuuksia. Monet alkuperäisen aihe-ehdotuksen ominaisuuksista saatiin tarkennettua jo alkupuolella projektia keskusteluissa tilaajan kanssa, kun osa ominaisuuksista rajattiin projektin ulkopuolelle tai päätettiin hyvin matalan tärkeysasteen ominaisuuksiksi.

Tilaaaja toivoi alun perin datan reaaliaikaista synkronointia. Ryhmän perehdyttyä laitteisiin ja todettua reaaliaikaisen synkronoinnin epäkäytännölliseksi saatiin tilaaajan kanssa sovittua, että data voidaan synkronoida jälkikäteen. Datan kerääminen reaaliaikaisesti onnistuu muun datan kohdalla, mutta ei katsedatan pisteytyksessä. Tämä johtuu silmänliikekameran ohjelmiston ominaisuuksista joihin ryhmä ei voinut vaikuttaa. Ryhmä sai kehitettyä sovelluksen niin, että katsedata saadaan synkronoitua muun datan kanssa jälkikäteen, kuten oli suunniteltu.

Ryhmä ehkäisi em. ongelmia tutustumalla laitteiden viiveisiin ja eri synkronointimahdollisuuksiin sekä kehittämällä prototyyppejä datan suodattamisesta.

Riskin vaikutus jäi vähäiseksi, sillä ryhmä sai toteutettua sovelluksen niin, että eri laitteilta tuleva data saadaan kerättyä joko reaaliaikaisesti tai lisättyä jälkikäteen. Halutut mittaustulokset voidaan suodattaa datasta ja data eri laitteilta synkronoida. Projektin aikana esiin tullut mahdollisuus silmänliikekameran ohjelmiston D-Labin etäkäytöstä jäi jatkokehittäväksi, mutta se ei kuulunut tilaaajan vaatimuksiin.

8.5 Ryhmän tietotaitojen puutteet

Projektin jäsenillä ei ollut työelämäkokemusta IT-alalta, ja tämä projekti oli ensimmäinen näin ison kokoluokan ohjelmointityö kaikille jäsenille. Jäsenillä ei ollut paljon kokemusta projektiluontoisesta työskentelystä, joten uutta opittavaa oli paljon, ja se olisi saattanut vaikuttaa projektin läpivientiin.

Suurimmalla osalla jäsenistä oli kuitenkin hyvät ohjelmointitaidot, joten projektissa tarvittavien taitojen oppiminen oli melko nopeaa. Tilaaaja myös antoi vapaat kädet ohjelmointikielen ja työkalujen valintaan, joten ryhmä valitsi ohjelmointikielen ja työkalut, joista oli jo kokemusta. Lisäksi ohjaaja auttoi tarvittaessa.

Ryhmän tietotaitojen puute ei vaikuttanut oleellisesti projektin läpivientiin.

8.6 Ryhmän jäsenten muut sitoumukset

Mustosella oli opintojen lisäksi osa-aikatyö (15h/viikko) maaliskuun puoliväliin saakka, mutta hän pystyi sen ohella sitoutumaan projektiin vähintään 20h/viikko tai enemmän, kuten muutkin ryhmän jäsenet.

Mustosella alkoi kokopäiväinen harjoittelu 20.5. ja hän sairastui 21.5., mutta hän jatkoi projektiraportin kirjoittamista ja muiden dokumenttien hiomista kotona pystyen edelleen tekemään suunnilleen saman määrän tunteja kuin muutkin jäsenet. Muut jäsenten sairastumiset aiheuttivat joinain viikkoina sen, että viikkotuntitavoitteeseen ei päästy joidenkin jäsenten kohdalla, mutta jäsenet saivat ne korvattua muina viikkoina.

Jäsenten muut sitoumukset eivät viivästyttäneet läpivientiä. Projektipäällikkö Mustosen sairastaessa varaprojektipäällikkö Kasanen hoiti projektipäällikön velvollisuuksia.

9 Jäsenten kokemuksia ja oppimaa

Luvussa kuvataan projektiryhmän jäsenten kokemuksia sovellusprojektista ja mitä he kokevat oppineensa sen aikana. Projekti oli kaikille ryhmän jäsenille ensimmäinen tämän kokoluokan projekti, jossa kehitettiin sovellusta sisältäen kaikki kehitysprosessin vaiheet tarpeiden kartoituksesta ja vaatimusmäärittelystä aina tulosten luovuttamiseen asti. Ryhmän jäsenet saivat kattavan kuvan projektin läpiviennin eri vaiheista, eri projektityötavoista, vuorovaikutuksesta mm. ryhmän jäsenten, ohjaajien ja tilaajien kanssa, sekä virallisen tekstin kirjoittamisesta ja projektin hallinnasta. Luvuissa 9.1–9.5 projektiryhmän jäsenet kuvaavat kokemuksiaan ja oppimiaan asioita.

9.1 Kasasen kokemuksia ja oppimaa

Muilla kursseilla ei ole ollut mahdollisuutta olla osana näin suurta projektia, joten odotin sovellusprojektissa oppivani työskentelemään säännöllisellä rytmillä pitkäköllä aikavälillä. Sanoisin tavoitteen onnistuneen, sillä tunteja kertyi tasaisesti koko kevään ajan. Kurssilla sain ensimmäistä kertaa makua koodaamisesta työnä, ja tämä varmasti auttoi ammatillisen itsetuntoni rakentumisesta. Tiesin odottaa, että kurssi olisi hankala, mutta en odottanut kurssin vaativan näin paljon oman terveyden ja vapaa-ajan uhraamista.

Olin ryhmän varapäällikkö, sekä kirjoitin suurimman osan sovelluksen käyttöliittymän koodista. Tämän lisäksi vastasin EEG-moduulista ja sovelluksen vaatimusmäärittelystä. Tämä oli ensimmäinen kerta, kun olen koodannut monisäikeistä sovellusta, joten opin siitä ainakin perusteet. Lisäksi monet projekteihin kuuluvat kommunikointikäytänteet olivat minulle uusia, ja erityisesti palvelerikäyttäytymisestä opin paljon.

Ryhmän haasteina kevään kuluessa olivat työtuntien tasaus jäsenten kesken ja projektin laajuus. Projektissa ei pelkästään automatisoitu tilaajalle tuttuja toimenpiteitä, vaan lisäksi jouduimme tutustumaan tilaajallekin tuntemattomiin

laitteisiin (EEG-laite ja uusi silmänliikekamera). Loppukäyttäjän mielipidettä käyttöliittymästä ei saatu tarpeeksi aikaisessa kehitysvaiheessa. Toteutusta rajaamalla ja käyttöliittymäkompromisseilla selvisimme, sekä saimme aikaan sovelluksen. Henkilökohtainen haasteeni oli projektin stressin ja oman hyvinvointini tasapainottelu. Yritin löytää omalle panostukselleni sellaisen tilan, jossa työni tulokset olisivat riittäviä, enkä silti palaisi loppuun. Tämä ei onnistunut niin hyvin kuin olisin halunnut.

En tuntenut muita ryhmän jäseniä entuudestaan. Olin aiemmin puhunut vain Arttu Ylä-Sahran kanssa muutamilla yhteisillä kursseilla. Ryhmän kommunikointi onnistui kuitenkin yleisesti ottaen hyvin, ja koin ryhmän jäsenten erilaisien vahvuuksien täydentävän toisiaan myös teknisellä puolella. Kevään aikana sovitut työnjaot olivat monilta osin toimivia, ja ongelmia kohdatessa apua sai muilta jäseniltä.

Kurssilla opin paljon hyödyllisiä asioita, kuten projektiluontoista työskentelyä, koodaustaitoja ja viestintää. Myös vähemmän hyödyllisiä asioita opin paljon. Yleisesti ottaen olen sitä mieltä, että mikään kurssi ei saisi aiheuttaa osallistujilleen tällaista ahdistusmäärää. En ole lainkaan vakuuttunut siitä, että tällaisesta panostuksesta palkinnoksi riittäisi kourallinen opintopisteitä, vaan mielestäni sovellusprojektien palkallisuutta tulee harkita vakavasti.

9.2 Liimataisen kokemuksia ja oppimaa

Tämä oli itselleni ensimmäinen kerta, kun pääsin mukaan isompaan ohjelmointiprojektiin. Minulla ei ollut varsinaisesti minkäänlaisia suurempia ennakkoodotuksia kurssin suhteen. Olin varma heti alusta alkaen, että kurssiin saisi varmasti laittaa paljon työtunteja. Odotin oppivani paljon uutta ohjelmoinnista ja ryhmänä toimimisesta. Ohjelmointitaitoni ennen kurssia eivät olleet hirveän kattavat, joten lähes kaikki kurssin aikana vastaan tullut oli ainakin jossain määrin uutta. Kurssin aikana opin muun muassa käyttämään paljon uusia työvälineitä ja ohjelmoimaan Androidilla.

Työskentelyäni hankaloitti alussa jonkin verran muiden kurssien päällekkäisyys. Tästä syystä jouduin usein tekemään töitä eri aikaan kuin muu ryhmä, mikä hankaloitti ryhmänä työskentelyä jossain määrin. Myöhemmin muista kursseista eroon päästyäni pystyin keskittymään paremmin projektiin. Alun perin oli myös tarkoitus, että olisin osallistunut enemmän käyttöliittymän toteuttamiseen. Minulla meni kuitenkin todella paljon aikaa Android-sovelluksen toteuttamisessa, mistä johtuen en juurikaan pystynyt panostamaan käyttöliittymän toteuttamiseen muuten kuin suunnittelutasolla. Isona yleisenä haasteena projektissa oli se, että ohjelmisto jouduttiin rakentamaan kokonaan itse. Tämä omalla tavallaan antoi meille paljon vapauksia, mutta samaan aikaan hankaloitti tekemistä.

En tuntenut ryhmäni jäseniä entuudestaan. Jo kurssin alussa pystyi nopeasti huomaamaan, että kaikki ryhmän jäsenet olivat hyvin erilaisia persooniltaan, taidoiltaan ja toimintatavoiltaan. Tästä huolimatta ryhmähenki on ollut koko projektin ajan mielestäni todella hyvä. Itse olen erityisesti todella kiitollinen siitä, että kokemattomampana tekijänä olen aina tarvittaessa voinut pyytää muilta apua.

Yleiskokemuksena kurssista on jäänyt itselleni hyvä kuva. Pää tavoitteena itselläni oli oppia mahdollisimman paljon uutta ohjelmoinnin maailmasta. Tämä tavoite todellakin täyttyi. Kurssin suorittaminen antaa myös hyvää harjoitusta oikeaan työelämään. Kurssin suorittamisesta saa myös todistuksen, mikä on todella hyvä lisäys omaan ansioluetteloon. Mielestäni kurssin suorittamisen aikana joutui kuitenkin tekemään liikaa paperityötä, mikä kulutti paljon aikaa projektin eteenpäin viemiseltä. Työmäärä oli myös todella suuri, mutta toisaalta niin on myös opintopisteiden määrä.

9.3 Mustosen kokemuksia ja oppimaa

Ennako-odotukseni sovellusprojektikurssista olivat, että se tulisi olemaan aikaa vievä ja haastava, mutta myös hyvin hyödyllinen. Odotin oppivani projektityöskentelyä, hallintaa, ohjelmointityökalujen käyttöä ja viestintää. Odotukseni täyttyivät pääosin. Tosin ohjelmointityökalujen käyttö jäi vähälle koska projektipäällikön muut vastuut veivät niin paljon aikaa. Projektin alussa minulla oli hieman jaksamisen kanssa ongelmia, sillä tein samaan aikaan osa-aikatyötä ja myös muita kursseja. Onneksi kuitenkin noin puolessa välissä projektia pystyin lopettamaan osa-aikatyöni.

Halusin itse projektipäälliköksi, sillä koin siinä roolissa pääseväni parhaiten käyttämään omia vahvuuksiani, joita ovat mm. aikataulutusta ja virallisen tekstin tuottaminen. Ohjelmointitaitoni taas eivät ole kovin vahvat, joten koin projektia hyödyttävän, jos muut enemmän ohjelmointia tehneet ryhmän jäsenet hoitaisivat siitä puolesta pääosan. Pääsin silti olemaan osa prosessia, jossa kehitellään sovellusta alusta alkaen, mikä oli mukavaa. Hallinnan tehtäviäkin riitti hyvin, joten koin olevani hyödyksi ryhmälleni niissä.

Kehitettävä sovellus oli haastava ja siihen liittyi aika paljon epävarmuutta projektin alussa. Kyseessä oli uuden sovelluksen kehittäminen, ja joitain siihen liittyviä laitteita ei oltu vielä kunnolla käytetty. Tilaaja oli kuitenkin hyvin joustava vaatimusten suhteen, ja asioita saatiin pääosin sovittua palaverissa. Vastuuohjaaja oli Jonne Itkonen, joka neuvoi enemmän sovelluksen suunnitteluun ja toteutukseen liittyvissä asioissa. Projektin hallinnasta ja dokumenttien kirjoittamisesta sai neuvoja toiselta ohjaajalta, Jukka-Pekka Santaselta. Molempien neuvot olivat hyviä ja auttoivat kehittymään projektipäällikkönä ja yleisemmin sovellusprojektissa työskentelyssä.

Ryhmän kanssa työskentely oli todella mukavaa. En tuntenut ketään heistä ennestään, mutta tutustuin hyvin projektin aikana. Työhuoneella oli mukava

ja rento tunnelma. Koin myös, että eri jäsenillä oli erilaisia vahvuusalueita, jotka sopivat hyvin yhteen ja hyödyttivät projektin läpivientiä monipuolisesti.

Projekti oli kaiken kaikkiaan hyvä ja hyödyllinen kokemus. Oppimiani asioita pääsen varmasti käyttämään tulevaisuudessa muilla kursseilla ja työelämässä.

9.4 Sundellin kokemuksia ja oppimaa

Sovellusprojektista kehkeytyi monipuolinen kokemus, johon mahtui niin onnistumisia kuin turhauttaviakin hetkiä. Projektin alussa oli mahdotonta ennakoita tulevan urakan laajuutta, sillä avoimia kysymyksiä riitti. Saadaanko kaikki data kerättyä laitteilta reaaliajassa, kuinka todeta synkronoinnin tarkkuus, tarvitaanko graafista käyttöliittymää ja niin edelleen. Hämmennystä ei helpottanut se, että osa laitteista oli uusia tilaajillekin.

Tyypillisellä ohjelmointikurssilla opettaja asettaa selvät vaatimukset harjoitustyölle, sekä ohjaa käytettyjen ohjelmointikielten ja kirjastojen valinnassa. Sovellusprojektin kulku oli tyystin toinen, sillä vaatimukset johdettiin tilaajan kanssa käydyistä keskusteluista, toteutusratkaisut suunniteltiin vapain käsin ja työvälineet valikoitiin oman harkinnan mukaan. Sovellusta kehitettäessä piti hahmottaa ajolaboratorion tutkimustyön nykyiset kompastuskivet ja pullonkaulat, jotka uudella sovelluksella tulisi ratkaista. Toisaalta kaikkia ongelmia ei projektiryhmän resursseilla voitu selvittää, joten odotukset oli pidettävä maltillisella tasolla.

Omiin vastuualueisiini kuuluivat silmäliikekameramoduulin sekä käyttöliittymän ja laitemoduulien välistä kommunikointia hoitaneen managerin kehittäminen. Silmäliikekameran ja sen ohjelmiston kanssa pakertaminen eteni pitkälti yrityksen ja erehdyksen kautta, kun yritin taivuttaa omaa moduuliani vastaanottamaan kaikkia mahdollisia syötevirtoja, joita kamerasta ulos saa. Sain-

pahan ainakin kokemusta laiteläheisestä I/O-ohjelmoinnista. Managerin puolella eniten opettelua vaati asynkroninen ohjelmointi, jonka avulla käyttöliittymä säilyy virkeänä, vaikka taustalla käsiteltäisiinkin valtavia datamääriä.

Projektin suurin päänsärky koitui hyväksymistestausympäristön valmistelusta. Vaikka Syncster toimikin odotetusti, ei silmäliikekameran ohjelmistoa saatu ajolaboratorion tietokoneilla toimimaan useamman tunnin yrityksestä huolimatta. Tästä opin sen, että kun kehitetään sovellusta, jota tullaan ajamaan tarkoin rajatussa ympäristössä, olisi syytä käydä testaamassa kyseisen ympäristön toimintaa pienimuotoisemmin ja useamman kerran ennen kuin kaikki laitteet kuskataan paikan päälle ja laitetaan sormet ristiin.

Työskentely projektiryhmän kanssa sujui mainiosti. Kaikki hoitivat oman osuutensa tunnollisesti, ja ryhmähenki säilyi positiivisena läpi projektin. Meillä oli selkeä työnjako, mikä aluksi johti siihen, että itse kukin hoiti vastuualueitaan hyvin itsenäisesti. Kuitenkin projektin loppua kohden keskinäinen vuoro-vaikutus onneksi lisääntyi, kun palasia alettiin sovittaa yhteen. Alun perin tähtäsimme siihen, että ohjelmointityö loppuu toukokuun alkuun mennessä. Aikataulu kuitenkin venyi lähes kuukaudella. Myöhästyminen ei kuitenkaan johdunut projektiryhmän laiskottelusta, vaan projekti yksinkertaisesti osoittautui hyvin laajaksi kokonaisuudeksi, ja eri osioiden hiominen vei aikansa.

Projektsuunnitelmassa kerroin toivovani aidonoloista kokemusta ohjelmistokehityksestä. Tarkoitin sillä tilaajan tarpeisiin räätälöitävän sovelluksen suunnittelemista ja toteuttamista. Toive tuli jotakuinkin täytetyksi. Sääntöjen laatiminen ja oheiskurssit söivät paikoitellen energiaa, jonka olisin mieluummin käyttänyt varsinaiseen kehitystyöhön. Toisaalta projektin aikana pidetyt väli- ja loppuesittelyt osoittautuivat hyödyllisiksi, sillä ne pakottivat ajattelemaan, kuinka sovelluksen toimintaa ja rakennetta voisi selittää ymmärrettävästi projektin ulkopuolisillekin tahoille

9.5 Ylä-Sahran kokemuksia ja oppimaa

Kurssille saapuessani ei ollut varsinaisia ennakko-odotuksia. Olin ensimmäinen tuttavapiiristäni, joka oli käymässä kyseistä kurssia, joten ns. "luotettavaa" tietoa ei varsinaisesti ollut (käytävähuhut jätin omaan arvoonsa). Lähin arvaus oli, että kyseessä on huomattavaa työmäärää vaativa kurssi jo pelkästään suuren opintopistemääränsä vuoksi, ja odotus oli ehdottomasti paikkansapitävä. Lisäksi takaraivossa nakutti tietty aavistus, että TIM liittyy mahdollisesti tavalla tai toisella asiaan ottaen huomioon, miten runsaasti TIM:iä on kehitetty viime vuosina. TIM:iä tässä kurssi-instanssissa olikin mutta TIM:in kehitys ei realisoitunut omaksi työtehtäväksi, joten siltä osin arvaus ei pitänyt paikkaansa.

Projekti oli ehdottoman antoisa siinä mielessä, että pääsin ensimmäistä kertaa osallistumaan todelliseen, järjestelmällisesti organisoituun ryhmätyöprojektiin. Aiempi ohjelmointikokemus oli pääsääntöisesti omista yksilöprojekteista, joissa tilaaja, ohjaaja ja kehittäjä olivat yksi ja sama henkilö. Oli opettavaista työskennellä projektissa, jossa on monta kehittäjää, ja myös muita ulkopuolisia sidosryhmiä esim. tilaajan ja ohjaajien muodossa. Väittäisin myös, että omat dokumentointitaidot kehittyivät kurssin aikana, epäilemättä omia tottumuksia tarkempien vaatimusten vuoksi.

Ehkä isoin haaste oli se, että projekti oli todellakin ns. alusta alkaen toteutettu projekti verrattuna esim. TIM-projekteihin. Projektin aikana kehitettiin alusta alkaen tarvittava ympäristö ja työkalut datan kaappaamiselle ja sen päälle vielä laitekohtaiset rajapinnat laitteille, joita tilaajakaan ei ollut kovin vakiintuneesti käyttänyt ennen sovellusprojektin aloittamista. Lisähaastetta henkilökohtaisesti toi oma sairastelu projektin aikana ja nyttemmin menestyksekkäästi haastetut omat yksilökeskeiset työtavat.

Oman kokemukseni mukaan ryhmähenki oli todella hyvä. Tietysti pieniä kipinöitäkin lenteli silloin tällöin, mutta se on lähinnä hyvä omasta mielestä, sillä

täysin riidaton tai läpeensä riidoissa oleva ryhmä ei ole terve. Nettona yhteistyö soljui varsin mutkattomasti ja ilman ylitsepääsemättömiä hankaluuksia. Ryhmän jäsenet eivät tunteneet toisiaan kovinkaan syvällisesti entuudestaan, joka ehkä myös auttoi ryhmän rakentumisessa ns. puhtaalta pöydältä ilman ennakkokäsityksiä.

Näin loppuyhteenvedona ymmärrän nyt paremmin sitä, miksi Sovellusprojekti-kurssi on maisteriohjelmassa pakollisena kurssina. Kurssi antaa valmiuksia työelämään tarjoamalla suhteellisen turvallisen ympäristön harjoitella projektityötä, ja mahdollisuuden saada myös yksityiskohtaista palautetta omasta työstään. Toisaalta, kurssi osoittautui epäilemättä kaikkein vaativimmiksi opintopisteiksi, joita olen yliopisto-urani aikana suorittanut. Mielestäni sen vaativuutta pitäisi korostaa vieläkin enemmän esim. opinto-ohjauksen yhteydessä.

10 Yhteenveto

Peltihamsteri-projekti kehitti kevään 2019 Sovellusprojekti-kurssilla Jyväskylän yliopiston kognitiotieteen ajosimulaatiolaboratoriolle sovelluksen, jonka avulla synkronoidaan ajosimulaatiokokeen eri laitteista saatava data ja tulostetaan ne yhteiseen tulostiedostoon. Tulostiedostossa dataa on helppo käsitellä ja analysoida.

Projektissa tärkeimmät sovellukselle asetetut tavoitteet toteutuivat. Toteutumatta jäi vain viisi matalan prioriteetin vaatimusta. Projektin tulokset luovutettiin tilaajalle kesäkuun toisella viikolla. Projektin haasteina olivat riittävä tutustuminen sovelluksen hyödyntämiin laitteisiin ja sovelluksen vaativuus. Projektin läpiviennin suurimmat riskit liittyivät laitteiden tietojen puutteeseen ja joustavuuden toteuttamisen vaikeuteen.

Projekti oli suunniteltua työläämpi, ja ryhmän jäsenet päätyivät tekemään paljon aiottua enemmän työtunteja. Sovellusta ja tuloksia ei voitu luovuttaa tilaajalle suunnitellusti toukokuun puolivälissä, vaan vasta kesäkuun toisella viikolla. Kokonaisaikataulu venyi noin neljällä viikolla.

Sovellusprojekti-kurssi tarjosi projektiryhmän jäsenille kokemusta projektimuotoisesta työskentelystä, projektin hallinnasta, tilaajan kanssa toimimisesta, johtamisesta ja ohjelmistokehityksestä. Sovelluksen kehittämisessä jäsenet saivat tilaisuuden hyödyntää aiemmissa opinnoissa omaksumiaan taitoja.

Lähteet

- [1] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra "Peltihamsteri-projekti, Projektisuunnitelma", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 4.4.2019.
- [2] Severi Jääskeläinen, Samuel Kaiponen, Heta Rekilä ja Sinikka Siironen, "Monisiro-projekti, Projektiraportti", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2.5.2018.
- [3] Jukka-Pekka Santanen, "Tietotekniikan sovellusprojektien ohje", saatavilla PDF-muodossa, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 30.1.2017.
- [4] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Vaatimusmäärittely", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 22.5.2019.
- [5] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Rakennekuvaus", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 10.6.2019.
- [6] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Järjestelmätestaussuunnitelma", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 14.5.2019.
- [7] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Järjestelmätestausraportti", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 26.4.2019.

- [8] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Järjestelmätestausraportti", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 14.5.2019.
- [9] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Järjestelmätestausraportti", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 26.5.2019.
- [10] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Sovellusraportti", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 6.6.2019.
- [11] Anthony D. Green, "C# Language Specifications" <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/introduction>, Microsoft 7.1.2017.
- [12] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra, "Peltihamsteri-projekti, Työajanseuranta", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 10.6.2019.
- [13] B. W. Boehm, "A Spiral Model of Software Development and Enhancement", Computer 21(5): 61-72. 1988.