

Potku-sovellusprojekti

**Jarkko Aalto
Timo Konu
Samuli Kärkkäinen
Samuli Rahkonen
Miika Raunio**

Projektiraportti

Julkinen
Versio 1.0.0
29.5.2013

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Varaprojektipäällikkö	__.__.2013		
Tilaaja	__.__.2013		
Ohjaaja	__.__.2013		

Tietoa dokumentista

Tekijät:

- Jarkko Aalto (JA) jarkko.t.aalto@student.jyu.fi
- Timo Konu (TK) timo.j.konu@student.jyu.fi
- Samuli Kärkkäinen (SK) samuli.p.p.karkkainen@student.jyu.fi
- Samuli Rahkonen (SR) samuli.p.j.rahkonen@student.jyu.fi
- Miika Raunio (MR) miika.o.raunio@student.jyu.fi

Dokumentin nimi: Potku-projekti, Projektiraportti

Sivumäärä: 64

Tiedosto: potku_projektiraportti_1.1.0.tex

Tiivistelmä: Potku-projekti kehitti Jyväskylän yliopiston fysiikan laitokselle rajoitettuun tuotantokäyttöön soveltuvan analyysisovelluksen rekyylispektrometrin mitausdatan analysointiin. Projektiraportissa kuvataan projektin suunniteltua ja toteutunutta läpivientiä käsitellen muun muassa tavoitteita, resursseja ja käytänteitä. Lisäksi projektiraportissa verrataan projektin toteutumaa suunnitelmaan mm. työnjaon ja työmäärien, prosessimallin, aikataulun sekä arvioitujen riskien hallinnan osalta ottaen kantaa erojen syihin ja vaikutuksiin.

Avainsanat: Aikataulu, kokemuksia, käytänteet, ohjelmistoprojekti, projektin läpivienti, projektin toteuma, prosessimalli, opittua, resurssit, riskien hallinta, tavoitteet, tehtävät, työmäärät, vastualueet.

Muutoshistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.0.1	29.4.2013	Projektiraportin laatiminen on aloitettu.	SK
0.0.2	30.4.2013	Projektiraporttia on muokattu projektisuunnitelman pohjalta.	SK
0.0.3	2.5.2013	Tehtäviä, työmääriä ja työnjakoa kuvaavaa lukua on muokattu.	SK
0.0.4	3.5.2013	Tehtäviä, työmääriä ja työnjakoa kuvaavaa lukua on muokattu.	SK
0.1.0	6.5.2013	Prosessimallia ja aikataulua, riskien hallintaa sekä yhteenvetoa kuvaavia lukuja on muokattu. Lisäksi kieliasua on korjattu.	SK
0.1.1	7.5.2013	Kieliasua ja sanamuotoja on korjattu.	SK
0.1.2	8.5.2013	Organisaatiota ja resursseja, käytänteitä sekä tehtäviä, työmääriä ja työnjakoa kuvaavia lukuja on muokattu.	SK
0.1.3	10.5.2013	Tehtäviä, työmääriä ja työnjakoa kuvaavaa lukua on muokattu.	SK
0.1.4	13.5.2013	Prosessimallia ja aikataulua sekä riskejä kuvaavia lukuja on muutettu. Alustava Gantt-kaavio toteutuneesta ajankäytöstä on lisätty. Työtunteja kuvaavaa taulukkoa on päivitetty.	SK
0.1.5	14.5.2013	Dokumentin kieliasua on siistitty. Tehtäviä, työmääriä ja työnjakoa, prosessimallia ja aikataulua sekä jäsenten kokemuksia ja oppimaa kuvaavia lukuja on muokattu.	SK
0.2.0	15.5.2013	Käytänteitä, tehtäviä, työmääriä ja työnjakoa, aikataulua ja prosessimallia sekä jäsenten kokemuksia ja oppimaa kuvaamia lukuja on muokattu.	SK
0.2.1	16.5.2013	Dokumentin kieliasua ja siinä olevia kirjoitusvirheitä on korjattu.	SK
0.2.2	17.5.2013	Dokumentin kieliasua ja siinä olevia kirjoitusvirheitä on korjattu.	SK

Versio	Päivämäärä	Muutokset	Tekijät
0.2.3	20.5.2013	Dokumentin kieliasua ja siinä olevia kirjoitusvirheitä on korjattu. Tehtäviä, työmääriä ja työnjakoa sekä prosessimallia ja aikataulua kuvaavien lukujen kuvat on päivitetty.	SK
0.3.0	21.5.2013	Tehtäviä, työmääriä ja työnjakoa sekä prosessimallia ja aikataulua kuvaavia lukuja on päivitetty. Dokumentin ulkoasua on viimeistelty.	SK
0.3.1	23.5.2013	Dokumentin kieliasua ja siinä olevia kirjoitusvirheitä on korjattu.	SK
0.4.0	24.5.2013	Dokumentin kieliasua ja siinä olevia kirjoitusvirheitä on korjattu. Tehtäviä, työmääriä ja työnjakoa sekä prosessimallia ja aikataulua kuvaavia lukuja on täydennetty sekä niissä olevia taulukoita ja kaavioita on viimeistelty.	SK
0.5.0	27.5.2013	Dokumentin kieliasua ja siinä olevia kirjoitusvirheitä on korjattu.	SK
0.5.1	27.5.2013	Dokumentin kieliasua ja siinä olevia kirjoitusvirheitä on korjattu	SK
1.0.0	29.5.2013	Kohdan 5.4 hakemistorakennetta on päivitetty.	MR

Tietoa projektista

Potku-projekti kehitti Jyväskylän yliopiston fysiikan laitokselle käyttöliittymän rekyylispektrometrin mittausdatan analyysiohjelmistoon, jolla korvattiin käytössä oleva vanha Finlandia-analyysisovellus. Osaa Finlandian käyttämistä analyysikomponenteista käytetään myös kehitetyssä sovelluksessa.

Tekijät:

- Jarkko Aalto (JA) `jarkko.t.aalto@student.jyu.fi`
- Timo Konu (TK) `timo.j.konu@student.jyu.fi`
- Samuli Kärkkäinen (SK) `samuli.p.p.karkkainen@student.jyu.fi`
- Samuli Rahkonen (SR) `samuli.p.j.rahkonen@student.jyu.fi`
- Miika Raunio (MR) `miika.o.raunio@student.jyu.fi`

Tilaaaja:

- Timo Sajavaara `timo.sajavaara@jyu.fi` 040-8054114
- Mikko Laitinen `mikko.i.laitinen@jyu.fi` 0400-994836
- Jaakko Julin `jaakko.julin@jyu.fi` 040-8054097
- Kai Arstila `kai.arstila@iki.fi` -

Ohjaajat:

- Jonne Itkonen `jonne.itkonen@jyu.fi` 050-4432381
- Jukka-Pekka Santanen `santanen@mit.jyu.fi` 040-8053299
- Tero Tuovinen `tero.tuovinen@jyu.fi` 050-4413685

Yhteystiedot:

- Sähköpostilista `potku@korppi.jyu.fi`
- Sähköpostiarkisto `https://korppi.jyu.fi/kotka/servlet/list-archive/potku/`
- Sähköpostilista (opetus) `potku_opetus@korppi.jyu.fi`
- Sähköpostiarkisto (opetus) `https://korppi.jyu.fi/kotka/servlet/list-archive/potku_opetus/`
- WWW-sivusto `http://sovellusprojektit.it.jyu.fi/potku/`

Sisältö

1	Johdanto	1
2	Termejä	2
2.1	Aihealueen termejä	2
2.2	Kehitysvälineisiin ja -tekniikoihin liittyviä termejä	4
3	Tavoitteiden toteutuminen ja tulokset	5
3.1	Taustaa ja tarpeita	5
3.2	Toteutettu sovellus	6
3.3	Sovelluksen tavoitteiden toteutuminen	8
3.4	Projektin tulokset	9
3.5	Oppimistavoitteiden toteutuminen	10
4	Organisaatio ja resurssit	13
4.1	Projektioorganisaatio	13
4.2	Projektin tilat ja laitteet	14
4.3	Ohjelmointi- ja dokumentointityökalut	15
4.4	Luennot ja perehdytykset	16
5	Käytänteet	17
5.1	Palaverit	17
5.2	Tiedotus	18
5.3	Tiedostojen nimeäminen	19
5.4	Hakemistorakenne	20
5.5	Lähdekoodi	21
5.6	Testaus	23
5.7	Versiohallinta ja -numerointi	24
5.8	Katselmoinnit ja tulosten hyväksyminen	25
5.9	Tulosten koostaminen ja toimittaminen	25
6	Tehtävät, työmäärät ja työnjako	26
6.1	Projektin jäsenten vastuualueita	26
6.2	Tehtäväkohtaiset työmäärät ja työnjako	27
6.3	Ryhmän työtunnit tehtäväkokonaisuuksittain	32
6.4	Jarkko Aallon työtunnit tehtäväkokonaisuuksittain	33
6.5	Timo Konun työtunnit tehtäväkokonaisuuksittain	34

6.6	Samuli Kärkkäisen työtunnit tehtäväkokonaisuuksittain	35
6.7	Samuli Rahkosen työtunnit tehtäväkokonaisuuksittain	37
6.8	Miika Raunion työtunnit tehtäväkokonaisuuksittain	38
7	Prosessi ja aikataulu	39
7.1	Prosessi	39
7.2	Aikataulu	40
7.3	Ryhmän työtunnit viikoittain	44
7.4	Jarkko Aallon työtunnit viikoittain	45
7.5	Timo Konun työtunnit viikoittain	46
7.6	Samuli Kärkkäisen työtunnit viikoittain	47
7.7	Samuli Rahkosen työtunnit viikoittain	48
7.8	Miika Raunion työtunnit viikoittain	49
8	Riskien hallinta	50
8.1	Arvioidut riskit, niiden todennäköisyydet ja haittavaikutukset	50
8.2	Sidosryhmien toiminnan viiveet	51
8.3	Kohdealueen haasteellisuus	52
8.4	Jäsenten puutteet toteutuksen tietotaidoissa	52
8.5	Analyysikomponenttien puutteet	53
8.6	Projektin hallinnan puutteet	54
8.7	Projektiorganisaatioon kuuluvien odottamattomat poissaolot	54
8.8	Tavoitteiden rajaus	55
8.9	Sovelluksen kehittäminen eri käyttöjärjestelmille	55
9	Jäsenten kokemuksia ja oppimaa	57
9.1	Jarkko Aallon kokemuksia ja oppimaa	57
9.2	Timo Konun kokemuksia ja oppimaa	58
9.3	Samuli Kärkkäisen kokemuksia ja oppimaa	59
9.4	Samuli Rahkosen kokemuksia ja oppimaa	60
9.5	Miika Raunion kokemuksia ja oppimaa	61
10	Yhteenveto	62
11	Lähteet	63

1 Johdanto

Potku-projekti kehitti Sovellusprojekti-kurssilla keväällä 2013 Jyväskylän yliopiston fysiikan laitoksen kiihdytinpohjaisen materiaalfysiikan tutkimusryhmälle **rajoitettuun tuotantokäyttöön soveltuvan analyysisovelluksen rekyylispektrometrin mittausdatan analysointiin**. Tutkimusryhmällä oli ennen projektia käytössään vanha Finlandia-niminen analyysisovellus, johon tutkijat eivät kaikin puolin olleet tyytyväisiä. Finlandian jatkokehitys ja ylläpito olivat osoittautuneet työläiksi ja haastaviksi. Tutkimusryhmä oli tullut siihen tulokseen, että on helpompaa ja järkevämpää kehittää kokonaan uusi sovellus kuin yrittää jatkokehittää vanhaa.

Projektiraportissa kuvataan projektin toteutunutta läpivientiä esitellen sen tulokset, resurssit, käytänteet, prosessimalli, aikataulu sekä riskien hallintaa. Lisäksi käsitellään projektiryhmän jäsenten tehtäviä, työmääriä ja tehtäväjakoja projektissa. Projektiraportin laatimisessa on hyödynnetty Potku-projektin projektisuunnitelmaa [10], Paatti-projektin projektiraporttia [5], Kuvatus-projektin projektiraporttia [2] sekä Sovellusprojektien ohjetta [7]. Projektissa laaditussa vaatimusmäärittelyssä [9] on kuvattu kehitetylle sovellukselle asetettuja vaatimuksia ja niiden toteumaa. Sovellusraportissa [11] on kuvattu sovelluksen käyttöliittymää, toteutusratkaisuja, havaittuja ongelmia ja jatkokehitysideoita. Projektin muut tulokset on esitelty luvussa 3.4.

Luvussa 2 kuvataan toteutettavaan sovellukseen ja projektiin liittyviä termejä. Luvussa 3 esitellään projektin taustoja, sekä tavoitteiden, tulosten ja projektiryhmän jäsenten oppimistavoitteiden toteutumista. Luku 4 esittelee projektin organisaation sekä projektiryhmän käytössä olleet resurssit. Luvussa käsitellään myös sovelluksen kehittämisessä käytettyjä työkaluja sekä Sovellusprojektin oheiskursseihin liittyneitä luentoja ja koulutusta. Luku 5 esittelee projektissa noudatettuja käytänteitä liittyen muun muassa kokouksiin, projektin sisäiseen tiedotukseen, dokumentointiin, versiohallintaan ja -numerointiin sekä tulosten koostamiseen ja toimittamiseen. Luvussa 6 käsitellään projektin jäsenten vastuualueet sekä esitellään projektin läpivientiin liittyviä tehtäviä ja niiden arvioituja työmääriä. Luvussa 7 esitellään projektin prosessi ja aikataulu. Luvussa 8 käsitellään projektin ennakoituja riskejä sekä niiden toteutumista, vaikutusta ja hallintaa. Luvussa 9 esitellään ryhmän jäsenten kokemuksia ja oppimaa projektista.

2 Termejä

Luvussa esitellään dokumentissa käytettäviä aihealueen ja kehitysvälineiden termejä.

2.1 Aihealueen termejä

Dokumentissa esiintyvät aihealueeseen liittyvät termit ovat seuraavat:

Alkuaine	on aine, jonka ytimessä on tietty määrä protoneja. Alkuaineet kuvataan yleensä jaksollisessa järjestelmässä , jossa alkuaineet on järjestetty ytimessä olevien protonien mukaan.
Alkuaineiden poistumat	(engl. <i>elemental losses</i>) ovat tapahtumia, joissa mittauksen lopussa on näytteessä vähemmän tiettyjä alkuaineita kuin sen alussa.
Finlandia	on fysiikan laitoksella tällä hetkellä käytössä oleva Belgiassa kehitetty rekyylispektrometrin mittaustatun analyysiohjelma. Finlandia hyödyntää Kai Arstilan kehittämiä analyysikomponentteja.
Ioni	on sähköisesti varattu atomi, jolla on eri määrä elektroneja kuin ytimessä on protoneja.
Isotooppi	on saman alkuaineen esiintymä, jolla on ytimessä eri määrä neutroneita.
Lentoaikakalibrointi	on toimenpide, joka muuttaa lentoajan kanavista (ch) sekunneiksi (s) tai nanosekunneiksi.
Kiihdytinlaboratorio	on fysiikan laitoksen tiloissa oleva laboratorio, jossa toimiva kiihdytinpohjaisen materiaalfysiikan tutkimusryhmä käyttää ionisuihkuja materiaalien tutkimiseen ja muokkaamiseen.
Näyte	on usein piin päälle kasvatettu ohutkalvo koostuen esimerkiksi alumiinioksidista Al_2O_3 .

Ohutkalvo	on rekyylispektrometrillä tutkittava kohde. Saman projektin tutkittavat kohteet voivat olla valmistettu- ja esimerkiksi eri lämpötiloissa. Kalvon paksuus tyyppillisimmissä näytteissä on 10–300 nm.
Projekti	on eri mittausten kokoelma, jossa voi olla samasta näytteestä esimerkiksi lämpötilasarja ja/tai paksuus-sarja.
Rekyloitunut ioni	on itse näytteestä ulos potkiintunut alkuaine, joka mitataan ToF-E -teleskoopilla (ERD).
Rekyylispektrometri	on fysiikan laitoksen käyttämä tutkimuslaitteisto, jolla Pelletron-kiihdyttimestä tulevien ammusionien avulla "potkitaan ulos" näytteen atomeja. Näiden atomien lentoaika ja energia mitataan. Laitteistoa saate-taan jossain yhteydessä kutsua ToF-E -teleskoopiksi.
Sironnut ioni	on alkuperäistä hiukkassuihkua (ammusioni), joka on törmännyt näytteen atomiin ja sironnut siitä ToF-E -teleskoopin suuntaan.
Syvyysprofiili	kuvaa alkuaineiden määrien suhteita syvyyden funk-tiona tutkittavassa kalvossa.
ToF-E -histogrammi	on lyhenne Time of Flight - Energy -histogrammista. Histogrammissa on kuvattu sironneiden ja rekyloitu-neiden atomien tuotot (määrät) lentoajan ja energian funktiona. Kyseisessä histogrammissa eri alkuainei-den (massojen) kertymäspektrejä kutsutaan myös "banaaneiksi".

2.2 Kehitysvälineisiin ja -tekniikoihin liittyviä termejä

Dokumentissa esiintyy seuraavia kehitysvälineisiin ja -tekniikoihin liittyviä termejä:

C	on ohjelmointikieli, jolla on kirjoitettu käyttöliittymästä kutsuttavat Kai Arstilan toteuttamat ohjelmat.
Doxygen	on työkalu, jolla generoidaan automaattisesti luokkadokumentaatio Pythonilla kirjoitetuista luokista.
Eclipse	on sovelluskehitysympäristö eri ohjelmointikielille. Projektiryhmä käyttää sitä lähdekoodin kirjoittamiseen.
Egit	on git-laajennus, joka laajentaa versiohallinnan myös Eclipseen.
Git	on hajautettu versiohallintaohjelmisto lähdekoodien ja dokumenttien hallintaan.
Matplotlib	on Pythonille kehitetty kaksiulotteisten kuvioiden piirtämiseen käytettävä kirjasto.
NumPy	on Python-laajennus, joka mahdollistaa suurien, moniulotteisten taulukoiden ja matriisien käsittelyn.
PyDev	on Eclipsen laajennus, jolla Python-ohjelmointikieli liitetään osaksi ohjelmointiympäristöä.
PyQt	on käyttöliittymäkirjasto Pythonille, jonka avulla toteutetaan sovelluksen graafinen käyttöliittymä.
Python	on ohjelmointikieli, jolla sovellus toteutetaan.
SciPy	on Python-laajennus, joka mahdollistaa erilaisten matemaattisten algoritmien suorittamisen.
YouSource	on WWW-pohjainen lähdekoodien julkistusjärjestelmä. YouSource tukee Git-versiohallintaohjelmistoa.

3 Tavoitteiden toteutuminen ja tulokset

Luvussa käsitellään projektissa toteutetun tietojärjestelmän, muiden projektin tulosten ja ryhmän oppimistavoitteiden toteutumista. Projektin aikana toteutettiin sovelusta rajoitettuun tuotantokäyttöön soveltuva versio (myöhemmin sovellus).

Sovelluksen pakollisista vaatimuksista toteutettiin 42/45, tärkeistä 4/16 ja mahdollisista 4/17. Sovellukseen kehitetty graafinen käyttöliittymä vastaa tilaajan tarpeita. Sen sijaan komentorivi- tai Python-tulkkitoiminnallisuus sovittiin tilaajan kanssa jatkokehitykseen. Projektiryhmän laatimat sovellusraportti [11] ja vaatimusmäärittely [9] antavat hyvät valmiudet sovelluksen jatkokehitykseen.

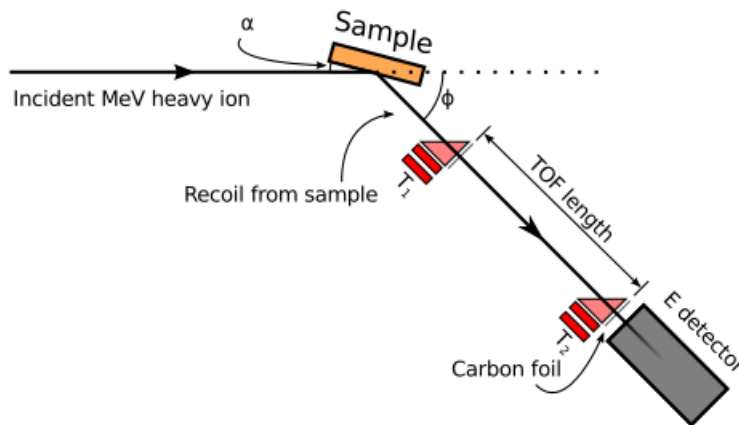
3.1 Taustaa ja tarpeita

Projektin tilaajana oli Jyväskylän yliopiston fysiikan laitoksen **kiihdytinpohjaisen materiaalfysiikan tutkimusryhmä**. Tutkimusryhmän toiminnalla on kolme pääpainopistealuetta. Ne ovat nano- ja mikromittakaavan rakenteiden valmistaminen ja muokkaus ionisuihkujen avulla, ionisuihkujen käyttö ohutkalvojen erilaisten materiaalien syvyysprofiloinnissa ja syklotronin suurien energisten suihkujen käyttö tutkimuksessa.

Tutkimusryhmän tehokkaimpia laitteistoja on **rekyylispektrometri**, jossa Pelletron-kiihdyttimestä tulevien ammusionien avulla "potkitaan ulos" tutkittavan näytteen atomeja. Poispotkittujen atomien lentoaika mitataan kahden hiilikalvoaikailmaisimen avulla ja energia mitataan puolijohde- tai kaasuilmaisimella. Yksittäiset mitaukset voivat kestää tunteja ja tuottaa miljoonia rivejä dataa. Mittauslaitteisto ja mittauksen toimintaperiaate on esitetty kuvassa 3.1.

Kuvassa 3.1 esiintyvä *sample* kuvaa tutkittavaa näytettä piikalvon päällä. Ionin tuloikulma on kuvattu α :lla ja sirontakulmaa ϕ :llä. Lentoaika mittaavat hiilikalvoilmaisimet esitetään kuvassa kohdilla T_1 ja T_2 . Näiden kahden kalvon välinen etäisyys on lentoaika *TOF length*, ja atomin energiaa mittaava ilmaisimien on *E detector*.

Tutkimusten yhteydessä tutkimusryhmä tekee useita mittauksia erilaisille näytteille. Heidän asiakkaansa on voinut esimerkiksi toimittaa näytteitä, jotka on valmistettu eri lämpötiloissa tai paineissa. Eri tavoin valmistettujen näytteiden eroavaisuuksia pyritään selvittämään. Tutkimusryhmä toimittaa näytteen analyysien tulokset tilaajilleen sekä/tai esittelee niitä tieteellisissä julkaisuissa ja tilaisuuksissa.



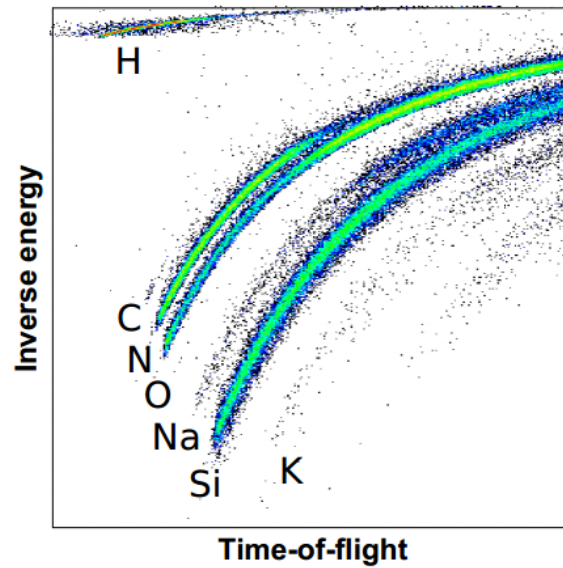
Kuva 3.1: Rekyylispektrometrin toimintaperiaate [4].

Tilajalla oli ennen projektia käytössään **Finlandia-niminen sovellus** mittaustietojen analysointiin. Siihen ei kuitenkaan oltu kaikilta osin tyytyväisiä, sillä siinä oli ohjelmointivirheitä ja sen kehittäminen katsottiin olevan haastavaa. Tilaja olikin tullut siihen tulokseen, että kokonaan uuden sovelluksen kehittäminen oli kannattavampi ratkaisu kuin vanhan ohjelman vaikea ylläpito ja jatkokehittäminen. **Potku-projekti kehitti tilaajan toivoman uuden sovelluksen mittaustietojen analysointiin.**

3.2 Toteutettu sovellus

Projektissa kehitetyllä sovelluksella analysoidaan rekyylispektrometrilla kerättyä mittaustietoa. Kehitetystä käyttöliittymästä kutsutaan Kai Arstilan toteuttamia C-kielisiä analyysikomponentteja.

Tutkimusryhmä tekee useita eri mittauksia, jotka yhdessä muodostavat projektin. Jokaisen mittauksen mittaustietoa sovellus pystyy muodostamaan ToF-E -histogrammin, jossa jokainen datapiste esitetään lentoajan ja energian suhteen. Histogrammissa samassa pisteessä voi olla useita datapisteitä. Niiden määrä esitetään kuvassa eri väreillä siten, että väri muuttuu logaritmisesti määrien mukaan. Kuvassa 3.2 on esitetty ToF-E -histogrammi.

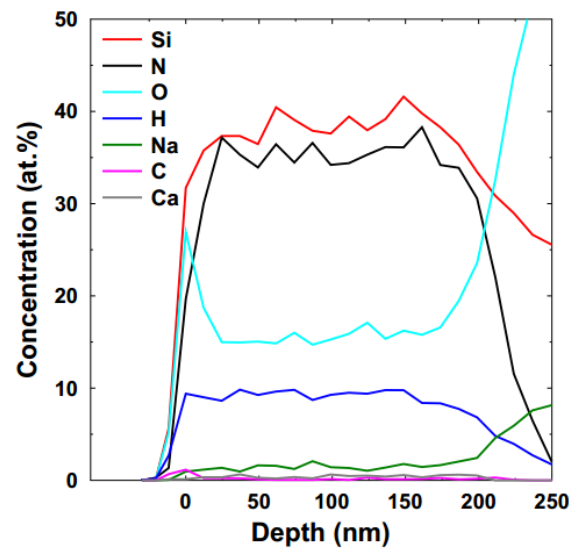


Kuva 3.2: Esimerkki ToF-E -histogrammista [8].

Käyttöliittymän avulla tutkijat **tunnistavat alkuaineita** muodostetusta ToF-E -histogrammista. Kyseiset rajatut alueet kuvaavat eri alkuaineita tai niiden isotooppeja. Kuvan 3.2 kirjainmerkinnät ovat mittauksessa havaitut eri alkuaineet, mutta kuvassa ei ole erikseen esitetty alkuaineille määritettyjä aluerajoja.

Sovelluksella pystytään syöttämään mittauksessa käytetyt **mittausparametrien arvot**, jotka sisältävät muun muassa lentoaikojen mittauksessa käytettävän etummaisena hiilikalvon paksuuden, hiilikalvojen etäisyyden sekä näytteen ja ilmaisimen kulmat. Sovelluksella pystyy myös analysoimaan, miten paljon mittauksen aikana näytteestä **poistuu valittuja alkuaineita**, sekä **muodostamaan energiaspektrejä**.

Luetun mittausdatan ja parametrien arvojen perusteella sovelluksella pystytään muodostamaan **syvyysprofiileja** kuvaten, kuinka paljon valittuja alkuaineita on tietyllä syvyydellä näytepisteessä. Syvyysprofiilista pystytään tämän jälkeen laskemaan eri alkuaineiden suhteet käyttäjän määrittämällä syvyysvälillä. Esimerkki syvyysprofiilista on esitetty kuvassa 3.3.



Kuva 3.3: Esimerkkikuva syvyysprofiilista [8].

3.3 Sovelluksen tavoitteiden toteutuminen

Tilajalla oli selkeä kuva **sovelluksen vaatimuksista**, sillä heillä oli käytössään vanha analyysisovellus. Tilaja osasi määritellä eri ominaisuudet tarkkaan, joten vaatimukset tiedettiin jo varhaisessa vaiheessa. Tilajan edustajilla oli kuitenkin hieman erimielisyyksiä ominaisuuksien prioriteeteista.

Vaatimusmäärittelyssä [9] vaatimukset jaettiin prioriteeteiltaan viisiportaiselle asteikolle. Asteikon portaat olivat pakollinen, tärkeä, mahdollinen, idea ja ei toteuteta. Projektiryhmä kirjasi **yhteensä 80 vaatimusta**. Pakollisista ominaisuuksista projektin aikana toteutettiin 42/45, tärkeistä 4/16 ja mahdollisista 4/17. Suunnitellun mukaisesti ideoita ei toteutettu lainkaan, koska niiden sisällöstä tai tarpeellisuudesta ei oltu varmoja.

Sovelluksen ominaisuuksista toteutettiin kokonaan

- ToF-E -histogrammin piirtäminen ja siihen liittyvät työkalut,
- alkuainevalinta ToF-E -histogrammista,
- alkuaineiden poistumien laskeminen,
- energiaspektri,
- lentoaikakalibraatio ja
- projektin hallinta.

Projektissa sovellukseen kehitetyistä toteutuneiksi on laskettu ominaisuudet, joiden

kaikki ensimmäisen tason prioriteetit on toteutettu.

Osittain toteutettuja ominaisuuksia olivat seuraavat:

- Tehokkuustiedostojen käyttämisen osalta analyysin yhteydessä kehitetty sovellus käyttää vain vedyn tehokkuutta.
- Syvyysprofiilin muodostamiseen liittyvää jarruuntumismallin valintaa ei ehditty toteuttaa.

Toteuttamatta jäivät seuraavat ominaisuudet:

- mahdollisuus käyttää sovellusta myös Python-tulkilta tai komentoriviltä,
- koko mittauksen kattavan raportin muodostaminen ja
- erilaiset plugin-toiminnallisuudet.

Käyttöliittymän käyttäjälleen tarjoamat tiedot ja toiminnot sekä niiden priorisointi ja tilat on kuvattu tarkemmin vaatimusmäärittelyssä [9]. Kehitetyn sovelluksen rakennetta, käyttöliittymää, heikkoja toteutusratkaisuja ja jatkokehitysideoita on kuvattu tarkemmin sovellusraportissa [11]. Testaussuunnitelmassa [12] testitapausten yhteyteen on kirjattu sovelluksen järjestelmätestauksen tulokset sekä mahdollisesti havaitut puutteet ja virheellinen toiminta.

3.4 Projektin tulokset

Kehitetyn käyttöliittymän ohella projektiryhmä toteutti seuraavat dokumentit:

- **Ajankäyttöraportti** sisältää jäsenten kirjaamat työtunnit sekä niiden jakautumisen eri tehtäväkokonaisuuksille ja tehtäville.
- **Esittelymateriaali** sisältää väli- ja loppuesittelyn esitysgrafiikat ja muistiot.
- **Itsearvioinnit** sisältävät ryhmän jäsenten arvioinnit omasta toiminnasta, kokemuksista ja oppimisesta. Lisäksi jäsenet arvioivat projektiorganisaatioon kuuluvien toimintaa ja onnistumista.
- **Lisenssisitoumuksella** ryhmän jäsenet vahvistavat sitoutuvansa sijoittamaan toteuttamansa lähdekoodin valitun avoimen lähdekoodin lisenssin alaisuuteen.
- **Luokkadokumentaatio** sisältää tiedon sovelluksessa käytettävistä luokista sekä niiden attribuuteista ja metodeista.
- **Lähdekoodi** sisältää lähdekoodin kommentteineen.

- **Oheiskurssien materiaalit** sisältävät oheiskurssien suoritukseen kuuluvat materiaalit ja harjoitukset.
- **Palaverien dokumentit** sisältävät palaverien esityslistat, tilakatsaukset ja pöytäkirjat.
- **Projektiraportti** kuvaa projektin toteutunutta läpivientiä sekä vertaa toteutunutta projektisuunnitelmaan pohtien erojen syitä ja vaikutuksia projektin läpivientiin ja tuloksiin.
- **Projektisuunnitelma** on projektin läpivientiä kuvaava suunnitelma, jossa esitellään muun muassa projektin tavoitteet, resurssit, käytänteet, tehtävien arvioidut työtunnit, aikataulu sekä riskien hallintaa.
- **Sovellusraportti** kuvaa toteutetun sovelluksen käyttöliittymää, oleellimmat toteutusratkaisut ja toiminnot, tavoitteiden toteutumista sekä olennaisimpia puutteita, heikkoja toteutusratkaisuja ja jatkokehitysideoita.
- **Sähköpostiarkistot** sisältävät kaikki projektin kahdella sähköpostilistalla käyty keskustelut.
- **Testaussuunnitelma** kuvaa suoritettavat testitapaukset ja testausympäristön kokoonpanon. Lisäksi siihen kirjataan kullakin testauskerralla suoritettujen testitapausten tulokset sekä havaitut virheet ja puutteet.
- **Vaatimusmäärittely** kuvaa kehitettävän sovelluksen tavoitteet, toiminnalliset ja tekniset vaatimukset sekä rajoitteet.

Suunnitelmassa pitäydyttiin lähes kaikkien dokumenttien osalta. Ainoa poikkeama suunnitelmaan oli testaussuunnitelman ja -raportin yhdistäminen samaksi dokumentiksi. Poikkeama johtui siitä, että tavallista suurempi projektiryhmä mahdollisti yhden ryhmän jäsenen keskittymisen testaamiseen. Hän laati kattavat testitapaukset sekä kirjasi suunnitelman ja testauksen tulokset samaan testausraporttiin [12].

3.5 Oppimistavoitteiden toteutuminen

Sovellusprojekti-kurssin oppimistavoitteena on tutustuttaa tietotekniikan opiskelijoita projektimuotoiseen ryhmätyöskentelyyn. Potku-projektin jäsenet saivatkin **kattavan kuvan sovellusprojektin läpiviennistä** aina vaatimusmäärittelystä tulosten toimittamiseen ja käyttöönottoon asti. Projektimuotoisessa työskentelyssä tärkeää oli **oman roolin ja vastuun ymmärtäminen** sekä **ajankäytön hallinta**. Koska kyseessä oli ohjelmiston kehitysprojekti, oppivat ryhmän jäsenet luonnollisesti myös **ohjelmointitaitoja**. Aiemmilla kursseilla oppittuja teoreettisia menetelmiä ja toimin-

tatapoja ryhmän jäsenet oppivat hyödyntämään myös käytännössä.

Sovellusprojekti-kurssilla oheiskurssina oli kirjoitus- ja puheviestinnän kurssi *Projektiviestintä IT-alalla*. Kirjoitusviestintäosiossa opiskelijat **kehittivät kirjallisia taitojaan** ja saivat opastusta muun muassa pöytäkirjojen kieliasun ja sisällön muokkaamiseen. Lisäksi kirjoitusviestinnässä keskityttiin oikeinkirjoitusasioihin, jotta kurssin jälkeen opiskelijat pystyisivät kirjoittamaan virheetöntä, alalle ominaista tekstiä. Puheviestinnän osiossa käsiteltiin **vuorovaikutusta ja esiintymistä** projektin eri tilanteissa. Neuvottelu- ja palaveritaitoja kehitettiin sekä rohkaistiin opiskelijoita toimimaan aktiivisesti palavereissa.

Sovellusprojektin yhteydessä suoritettiin myös kurssi *Sovellusprojektin hallintaa, viestintää ja työkaluja*. Kurssilla järjestettiin puolisen tusinaa luentoa, joiden kautta projektiryhmän jäsenet saivat tietotaitoja projektin hallinnasta, viestinnästä ja työkaluista projektin aikana.

Projektin varsinaisen läpiviennin lisäksi opiskelijat oppivat **yhteistyötä ryhmän sisällä**. Työskenteleminen ryhmässä vaatii vastuunottamista ja -antamista. Projektin aikana opiskelijat oppivat luottamaan itseensä ja ryhmän muihin jäseniin. Pienryhmässä toimiessa myös erilaisiin ongelma- ja ristiriitatilanteisiin oli varauduttava. Näitä ongelmatilanteita ratkoessaan ryhmän jäsenet oppivat myös työelämässä tarvittavia kykyjä. Projektiryhmän sisäisen yhteydenpidon lisäksi projektiryhmä teki aktiivisesti yhteistyötä toisen vuoden 2013 sovellusprojektiryhmän, Hoksottimen, kanssa.

Edellä mainittujen yleisten oppimistavoitteiden lisäksi ryhmän jäsenet asettivat itselleen seuraavat **henkilökohtaiset tavoitteet**:

- Jarkko Aalto halusi oppia työelämässä vaadittavia projektityöskentelyn ja ryhmätyön taitoja.
- Timo Konu halusi kokemusta projektimuotoiseen ohjelmistokehitykseen ja erityisesti ryhmätyöskentelyyn liittyen.
- Samuli Kärkkäinen halusi oppia projektin suunnittelua ja hallintaa sekä projektipäällikkönä toimimista.
- Samuli Rahkonen toivoi saavansa erityisesti työkokemusta sovelluskehityksestä ja siitä, miten laajempia ohjelmistoja kehitetään.
- Miika Raunio halusi kokemusta projektimuotoisesta työskentelystä ja ohjelmointityöstä.

Henkilökohtaiset oppimistavoitteet toteutuivat jokaisen ryhmän jäsenen osalta.

Ryhmän jäsenet ymmärsivät hyvin oman roolin ja vastuun merkityksen sekä ajankäytön hallinnan merkityksen. Myös työajan kirjaus oli osalle ryhmästä uutta, mutta tämä tapahtui projektin aikana tunnollisesti. Kukaan ryhmän jäsen ei analysoinut oppimaansa tarkemmin luvussa 9.

Myös **oheiskurssien oppimistavoitteet toteutuivat**, ja ryhmän jäsenet saivat selkeän kuvan siitä, että sovellusprojektit ovat kaikkea muutakin kuin pelkkää ohjelmointia. Puhe- ja kirjoitusviestintää ryhmän jäsenet oppivat väliesittelyjen, palaverien ja dokumentoinnin yhteydessä. Jäsenet kehittyivät jokaisella osa-alueella projektin aikana, ja oheiskurssit koettiin erittäin tarpeellisiksi.

4 Organisaatio ja resurssit

Luvussa käsitellään projektiorganisaatioon kuuluvien henkilöiden lisäksi muita projektin käytössä olleita resursseja, kuten työtiloja, laitteita ja työkaluja.

Projektisuunnitelmaan kirjattu projektiorganisaatio ja resurssit toteutuivat, eikä merkittäviä muutoksia tullut. Suurin muutos oli osan palavereista pitäminen fysiikan laitoksen tiloissa. Tämä oli kuitenkin tiedossa jo projektin ensimmäisen kuukauden aikana.

4.1 Projektiorganisaatio

Projektiryhmä oli projektissa poikkeuksellisen suuri, koska siihen kuului viisi jäsentä normaalin 3–4 sijasta. Projektiryhmään kuuluivat Jarkko Aalto, Timo Konu, Samuli Kärkkäinen, Samuli Rahkonen ja Miika Raunio. **Projektipäällikkönä** toimi Samuli Kärkkäinen ja **varapäällikkönä** Jarkko Aalto.

Jarkko Aalto oli suorittanut yliopiston lisäksi myös ammattikorkeakoulussa ohjelmisto- ja tietotekniikan opintoja sekä työskennellyt testauspäällikkönä pienessä yrityksessä. Samuli Kärkkäinen oli työskennellyt testaajana ja ohjelmoijana suuressa yrityksessä. Timo Konun, Samuli Rahkosen ja Miika Raunion kokemukset rajoittuivat yliopistossa suoritettuihin kursseihin. Informaatioteknologian opintojen lisäksi Samuli Rahkonen oli suorittanut elektroniikan ja fysiikan perusopintoja yliopistotasolla.

Tilaaajan edustajat olivat Jaakko Julin, Mikko Laitinen, Timo Sajavaara ja Kai Arstila. Julin, Laitinen ja Sajavaara työskentelivät Jyväskylän yliopiston fysiikan laitoksella. Arstila työskenteli sovellusprojektin aikana Belgiassa, mutta siirtyy fysiikan laitokselle kesäkuun 2013 alussa. Tietotekniikan laitokselta projektin **vastaavana ohjaajana** toimi Jukka-Pekka Santanen ja **teknisenä ohjaajana** Jonne Itkonen. Tero Tuovinen toimi projektissa **asiantuntijana**.

Jyväskylän yliopiston IT-palvelut vastasi ryhmän jäsenten tietokoneiden ja ohjelmistojen ylläpidosta ollen näin projektin ainoa **sidosryhmä**. Sovellusprojektin yhteyshenkilö ATK-tuessa oli Santeri Lapinmäki.

Kurssin yhteydessä järjestettävän **viestintäkurssin yhteyshenkilöt** olivat Kaisa Leino kirjoitusviestinnässä ja Maritta Stoor-Lehtonen puheviestinnässä. Meeri Mäntylä

piti **käytettävyyispäivän** projektin jäsenille.

Projektisuunnitelmaan kirjattu projektiorganisaatio toteutui. Arstilan ja Tuovisen osallistuminen projektin läpivientiin oli tosin vähäistä. Itkonen olisi kenties voinut osallistua projektiin tiiviimmin käymällä esimerkiksi palavereissa, jotta kohdealue olisi selvinnyt hänelle. Toisaalta hän oli kiireinen päätyönsä kanssa, joten täyttä sitoutumista ei tietenkään voinut vaatia. Tekninen ohjaaja oli kuitenkin hyvin tavoitettavissa, sillä hänen työhuoneensa oli projektiryhmän työhuoneen kanssa samassa rakennuksessa. Kenties projektiryhmä olisi voinut tukeutua tekniseen ohjaajaan myös hieman enemmän projektin aikana.

4.2 Projektin tilat ja laitteet

Tietotekniikan laitos tarjosi projektiryhmälle **työhuoneen AgC222.2** Agoran C-siiven toisesta kerroksesta. Huoneen läheisyydessä oli monitoimilaitte, jolla ryhmän jäsenet pystyivät tulostamaan, skannaamaan ja kopioimaan projektiin liittyviä dokumentteja. Jokaisen ryhmän jäsenen käyttöön tietotekniikan laitos antoi projektin ajaksi tietokoneen. Yhdessä tietokoneessa oli käyttöjärjestelmänä Fedora 17 ja neljässä Windows 7.

Suunnitelman mukaan ryhmän oli tarkoitus testata sovellusta Mac-luokassa Agoran kolmannessa kerroksessa. Ryhmän ei kuitenkaan tarvinnut käyttää Mac-luokkaa testaukseen, vaan ryhmä sai yhden Macin omaan työhuoneeseensa projektin loppupuolella. Tämä helpotti testaamista Mac-ympäristössä merkittävästi.

Ryhmällä oli käytössään **kaksi verkkolevyä**, joista toinen on tarkoitettu ryhmän sisäiseen tietojen jakamiseen ja toinen projektin WWW-sivuja varten.

Suunnitelman mukaan projektin palaverit pidettäisiin Agoran C-siivessä sijaitsevassa **kokoushuone AgC226.2**. Huoneessa on tietokone ja videoprojektori, joten esitysgrafiikan näyttäminen kokoushuoneessa onnistui tapaamisten yhteydessä.

Suunnitelmasta poiketen palavereita pidettiin myös **fysiikan laitoksen palaverihuoneissa**. Tämä ei vaikuttanut projektin läpivientiin, sillä fysiikan laitos sijaitsee lähellä tietotekniikan laitosta. Lisäksi fysiikan laitoksen kokoushuoneessa oli projektori, jolla pystyttiin näyttämään esitysgrafiikkaa ja kehitettyä sovellusta tapaamisten yhteydessä.

4.3 Ohjelmointi- ja dokumentointityökalut

Projektissa kehitetyn sovelluksen ohjelmointikielenä käytettiin **Python 3.3:a**. Pythonin **laajennuksista** käytössä olivat NumPy 1.7.0, SciPy 0.12.0, PyQt 4.10 ja Matplotlib 1.2.0. Sovelluskehitysympäristönä käytettiin Eclipseä ja sen laajennusta PyDev 2.7.1:ä. Sovelluksen käyttöliittymä toteutettiin PyQt:lla ja Matplotlibillä. Kai Arstilan ohjelmamoduulit oli kirjoitettu C-kielellä.

Versiohallintaan ryhmä käytti Git-versiohallintasovellusta, jonka avulla projektiin liittyvät lähdekoodit ja dokumentit siirrettiin YouSource-nimiseen lähdekoodien julkistusjärjestelmään. **Luokkadokumentaatio** muodostettiin suoraan lähdekoodista käyttäen Doxygen-työkalua. Luokkadokumentaation muodostaminen poikkesi suunnitelmasta, jossa se oli suunniteltu muodostettavaksi PyDocilla.

Projektin **keskeisimmät dokumentit** kirjoitettiin L^AT_EX-ohjelmistolla. Muiden dokumenttien laatimiseen ryhmän jäsenet käyttivät tilanteesta riippuen parhaaksi katsoomaansa ohjelmistoa.

Projektin palaverien yhteydessä esiteltävät tilakatsaukset sekä väli- ja loppuesitelyjen esitysgrafiikat laadittiin käyttäen Microsoft PowerPoint -sovellusta. Tilakatsauksessa hyödynnettiin ryhmän jäsenten kirjaamia työtunteja. Työtunnit kirjattiin käyttäen Petri Heinosen Sovellusprojekteille kehittämää Excel-sovellusta [1].

Ohjelmointikieli **Python ei ollut ryhmän jäsenille entuudestaan kovin tuttu**. Ainoastaan Raunio ja Kärkkäinen olivat tehneet toisella kurssilla pienen harjoitustyön käyttäen Pythonia. Myös PyQT ja Git olivat ryhmän jäsenelle ennestään tuntemattomia. Tämä vaikutti projektin läpiviennissä siihen, että projektiryhmä joutui käyttämään aikaansa uusien asioiden opetteluun. Ryhmä kuitenkin oppi käyttämään kaikkia työkaluja hakemalla tietoa Internetistä sekä teknisen ohjaajan avustuksella. Myös ryhmän sisällä jaettiin tietoa työkaluista. Työkaluihin tutustunut henkilö osasi vastata ryhmän muiden jäsenten kysymyksiin ja opasti myös omatoimisesti muita työkalujen käytössä.

4.4 Luennot ja perehdytykset

Sovellusprojekti kurssin ohella ryhmän jäsenet suorittivat oheiskurssit *Sovellusprojektin hallintaa, viestintää ja työkaluja* sekä *Projektiviestintä IT-alalla*. Näillä kursseilla jäsenet oppivat projektin hallintaan ja projektiviestintään liittyviä taitoja. Luennot ja tapaamiset keskittyvät seuraaviin aiheisiin:

- aloitusluento
- versiohallinta
- projektiluento
- Python-perehdytys
- mahdolliset projektipäälliköiden tapaamiset
- kokous- ja neuvottelukäytänteet
- esiintymisharjoituksia
- kirjoitusviestintä ja kieliopilliset asiat
- kaksi väliesittelyä.

Projektin yhteydessä projektiryhmän jäsenet tutustuivat myös omatoimisesti sekä tietoteknisiin että kirjoitusasullisiin asioihin.

Projektin aikana pidettiin kaikki suunnitelmaan kirjatut luennot ja tapaamiset. Projektipäälliköiden tapaamisia ei varsinaisesti järjestetty, vaan **projektipäälliköt kävivät säännöllisesti läpi projektin läpivientiin liittyviä asioita** ja tarvittaessa opastivat toisiaan.

5 Käytänteet

Luvussa kuvataan projektissa noudatettuja käytänteitä. Nämä käytänteet **edesauttoivat projektin läpivientiä, asetettujen tavoitteiden saavuttamista ja tulosten toteuttamista**. Käytänteiden määrittelyllä varmistettiin, että projekti etenee aikataulussa ja sen aikana syntyvät tulokset ovat korkealaatuisia.

Käytänteet poikkesivat suunnitellusta testauksen ja julkaistavien versioiden osalta. Suunniteltua lähdekoodiin kirjoitettua yksikkötestausta tehtiin projektin lopussa vähän. Paremmalla yksikkötestauksella olisi mahdollisesti nopeutettu projektin etenemistä. Onneksi tilanteita, joissa yksikkötestauksesta olisi ollut merkittävää hyötyä, ei projektin aikana ilmennyt. Jatkokehityksessä niistä olisi kuitenkin ollut hyötyä.

Tilaaajalle ei myöskään toimitettu versioita koekäyttöön. Suunnitelman mukaan projektiryhmä olisi julkistanut jo melko varhaisessa vaiheessa sovelluksen toimivan rungon tilaaajalle. Syvyysprofiilien muodostamiseen meni kuitenkin suunniteltua pidempään, eikä projektiryhmä halunnut toimittaa tilaaajalle versiota, jossa koko analyysiprosessia ei ole mahdollista suorittaa. Prototyyppejä ja niiden toimintaa esiteltiin palavereissa sekä tilaajan edustajista Julin seurasi sovelluksen edistymistä aktiivisesti versiohallinnasta lataamallaan versioilla.

5.1 Palaverit

Projektin yhteydessä pidettiin palaveria aluksi viikon välein. Kun projekti saatiin käyntiin ja tapaamistarve väheni, siirryttiin jo helmikuun aikana pitämään palaveria noin kahden viikon välein.

Palaveriin osallistuivat kaikki projektiryhmän jäsenet, ohjaajat ja tilaajan edustajat. Palaverit olivat **laillisia**, jos palaverin esityslista oli lähetetty vähintään vuorokautta ennen palaverin alkua. **Päätösvaltaiseksi** palaveri katsottiin, jos paikalla oli vähintään yksi projektiryhmän edustaja, yksi tilaajan edustaja ja vastaava ohjaaja.

Kolmea ensimmäistä palaveria lukuunottamatta jokaisessa palaverissa projektipäällikkö piti **tilakatsauksen**, jossa esiteltiin edellisen palaverin jälkeen suoritettujen toimenpiteiden sekä kohdatut ongelmat. Lisäksi käsiteltiin seuraavia toimenpiteitä sekä esiteltiin tehtäväkokonaisuuksittain ja viikoittain projektiin käytetyt työtunnit jokaisen ryhmän jäsenen osalta.

Tilakatsauksen ohella palavereissa käsiteltiin kehitettävän **sovelluksen ominaisuuksia ja toteutusratkaisuja**. Kohdealueen haasteellisuuden takia palavereissa keskusteltiin melko paljon mittaukseen liittyvistä toimenpiteistä sekä itse kohdealueesta. Myös tilaajan toimittamien analyysikomponenttien toimintaa jouduttiin selventämään ja käymään läpi.

Jokainen ryhmän jäsen toimi vuorollaan palavereissa **sihteerinä ja puheenjohtajana**. Toimittuaan ensin sihteerinä siirtyi jäsen seuraavassa palaverissa puheenjohtajaksi. Sihteerin laati palaverista pöytäkirjan, jonka hän toimitti puheenjohtajalle. Puheenjohtaja tarkasti pöytäkirjan ja tarvittaessa pyysi sihteerää tekemään siihen muutoksia. Kun pöytäkirja oli puheenjohtajan osalta hyväksytty, sihteerin toimitti sen koko projektioorganisaatiolle.

Kun joku ryhmän jäsenistä toimi ensimmäistä kertaa sihteerinä, toimitti puheenjohtaja alustavan pöytäkirjan kera siihen kirjaamansa palautteen myös vastaavalle ohjaajalle ja kirjoitusviestinnän opettajalle. Myös he tarkastivat pöytäkirjan ja tarvittaessa ilmoittivat sihteerille tarvittavista muutoksista. Kun kaikki olivat toimineet kerran puheenjohtajana ja sihteerinä, vuorot jaettiin projektiryhmässä sen mukaan, kenellä kullakin viikolla oli eniten aikaa keskittyä kyseisiin rooleihin.

Pöytäkirja tarkastettiin aina seuraavassa palaverissa. Tarvittaessa palaverien yhteydessä pöytäkirjaan esitettiin muutoksia. Jos pöytäkirjassa oli pieniä virheitä tai puutteita, hyväksyttiin se pienin muutoksin. Tällöin sihteerin teki tarvittavat korjaukset ja lähetti muokatun pöytäkirjan sähköpostilla.

Palaverien suunnitellut käytänteet toteutuivat projektissa.

5.2 Tiedotus

Projektin **tiedotuksesta** vastasi ensisijaisesti projektipäällikkö. Jokainen **ryhmän jäsenen ilmoitti** omatoimisesti vähintään projektipäällikölle hänelle määrättyjen toimenpiteiden edistymisestä. Projektipäällikkö piti itsensä aktiivisesti tietoisena projektin etenemisestä. Jos tiedotus tapahtui vain projektipäällikölle, projektipäällikkö tiedotti oleelliset asiat eteenpäin koko organisaatiolle. Projektiin liittyvistä valinnoista, muutoksista ja vaihtoehtoista keskusteltiin pääasiassa projektipalavereissa.

Jos viikolla ei pidetty palaveria, projektipäällikkö toimitti **välitarkastuksen** projektioorganisaatiolle. Välitarkastus sisälsi samat asiat kuin palaverien yhteydessä esitetty

tilakatsaus. Ongelmatilanteista kuvattiin myös se, onko ongelma saatu ratkaistua ja minkälainen ratkaisu ongelmaan on löytynyt.

Projektioorganisaatiolla oli käytössään sähköpostilista `potku@korppi.jyu.fi`, jolla ryhmän jäsenet, ohjaajat ja tilaajan edustajat tiedottivat projektiin liittyviä asioita. Listalle lähetetyt viestit tallentuivat suojattuun postiar্কistoon, joka löytyy osoitteesta <https://korppi.jyu.fi/kotka/servlet/list-archive/potku>.

Projektiryhmän jäsenten ja ohjaajien käytössä oli myös sähköpostilista `potku_opetus@korppi.jyu.fi`. Sen postiar্কisto sijaitsee suojatussa osoitteessa https://korppi.jyu.fi/kotka/servlet/list-archive/potku_opetus/. Sähköpostilistalla käsiteltiin sellaisia asioita, jotka eivät olleet merkityksellisiä tilaajalle. Tällaisia asioita olivat esimerkiksi projektiryhmän sisäiset palaverit sekä erilaiset opetukseen ja koulutukseen liittyvät asiat.

Ryhmän sisäinen tiedotus tapahtui pääosin suullisesti, sillä ryhmän jäsenet työskentelivät samassa tilassa ja olivat yhteydessä toisiinsa lähes päivittäin. Tarvittaessa tiedotus ja keskustelu tapahtuivat sähköpostitse käyttäen jäsenen sähköpostiosoitetta tai edellä mainittuja sähköpostilistoja.

Tiedotuksen toteuma vastasi suunnitelmaa. **Projektiryhmä oli aktiivisesti yhteydessä tilaajaan**, sekä kukin ryhmän jäsen lähetti kysymyksiä ja tarkennuksia sähköpostilistoille. Mikään asia ei ollut niin kiireinen, että sitä olisi täytynyt hoitaa puhelemella.

Projektiryhmä keskusteli projektihuoneessa projektin läpivientiin sekä sovelluksen kehitykseen ja suunnitteluun liittyvistä asioista. Projektiryhmällä oli samanlaiset työajat, joten keskustelu onnistui helposti työpäivän aikana suullisesti.

5.3 Tiedostojen nimeäminen

Lähdekooditiedostojen nimeämisessä käytettiin Pythonin yleisiä käytänteitä [3]. Tiedostojen ja hakemistojen nimet kirjoitettiin **englanniksi**. Tiedostonimissä ei käytetty välilyöntejä, vaan niiden tilalla käytettiin alaviivaa.

Dokumenttitiedostot nimettiin sisältöä kuvaavilla nimillä ja **dokumentin kielen mukaisesti**. Lisäksi dokumenttien nimiin lisättiin **versionumero** luvussa 5.7 esitettyjen käytänteiden mukaisesti, esimerkiksi `potku_palaveri_[palaveri]_poytakirja_[numero].[numero].[numero].pdf`.

Palaverihin liittyvät dokumenttiedostot nimettiin seuraavasti:

- Word-muotoisten pöytäkirjojen nimeämistapa oli
potku_palaveri_[järjestysnumero]_poytakirja_[versio].docx.
- Pdf-muotoisten pöytäkirjojen nimeämistapa oli
potku_palaveri_[järjestysnumero]_poytakirja_[versio].pdf.
- Pdf-muotoiset tilakatsaukset nimettiin
potku_tilakatsaus_[järjestysnumero].pdf.

Projektin aikana tiedostot nimettiin suunnitellusti. Versiohallintaa ja nimeämistä on käsitelty tarkemmin luvussa 5.7.

5.4 Hakemistorakenne

Projektin tulokset tallennettiin CD-levylle ja projektin WWW-hakemistoon päähakemiston alle seuraavan **hakemistorakenteen** mukaisesti:

```
ajankaytto
application_report
class_documentation
esittelyt
itsearviointit
kayttoliittymahahmotelma
muistiot
palaverit
    esityslistat
    poytakirjat
    tilakatsaukset
projektiraportti
projektisuunnitelma
requirements_specification
sahkopostiarkistot
    potku
    potku_opetus
sitoumus_ja_lisenssit
software_and_instructions
source_code
```

system_testing_plan

Hakemistorakenteen osalta suunnitelmasta poikettiin hieman. Projektin WWW-sivulle koottiin **kaksi päähakemistoa**, joista toisen alle sijoitettiin kaikki dokumentit ja toiseen pelkästään englannin kieliset dokumentit vaatimusmäärittely, luokkadokumentaatio sekä sovellus- ja testausraportti. Projektissa **ei kirjoitettu käyttöohjeita**, vaan tilaajan kanssa sovittiin, että tilaaja laatii itse ohjeen jatkokehityksen aikana. Suunniteltua testauskansiota ei ole, vaan **testaussuunnitelma on omana kansionaan**.

5.5 Lähdekoodi

Lähdekoodi kirjoitettiin vastaamaan yleisiä Python-käytänteitä [3] ja kommentit Pythonin kommentoinnin mukaisia käytänteitä [6]. Tilaajan toimittamiin analyysikomponentteihin toteutetut muutokset kirjoitettiin noudattaen standardeja, joita tilaaja oli analyysikomponenteissa itse käyttänyt.

Lähdekoodissa käytetyt **aliohjelmat, luokat ja muuttujat nimettiin** mahdollisimman kuvaavilla **englanninkielisillä nimillä**. Myös lähdekoodin kommentointi tapahtui englanniksi.

Lähdekoodin ja kommentoinnin käytänteet toteutuivat suunnitelman mukaisesti. Ohessa on esimerkki edellisiä käytänteitä noudattaen toteutetusta Python-koodista.

```
'''
```

```
Created 15.3.2013
```

```
Selection.py handles Selector and Selection objects.
```

```
'''
```

```
__author__ = "Jarkko Aalto \n Timo Konu \n  
Samuli Kärkkäinen \n Samuli Rahkonen \n Miika Raunio"
```

```
__versio__ = "1.0"
```

```
def select(self, point, highlight = True):
```

```
    '''Select a selection based on point.
```

```
    Args:
```

```
        point: Point (x, y) which is clicked on the graph  
        to select selection.
```

```
        highlight: Boolean to determine whether to highlight  
        just this selection.
```

```
    Return:
```

```
        1: If point is within selection.
```

```
        0: If point is not within selection.
```

```
    '''
```

```
    for selection in self.selections:
```

```
        path = Path(selection.get_points())
```

```
        if path.contains_point(point):
```

```
            self.selected_id = selection.id # Save selection
```

```
            if highlight:
```

```
                self.grey_out_except(selection.id)
```

```
            return 1
```

```
    return 0
```

5.6 Testaus

Toteutetun sovelluksen toiminnan **verifiointi** tapahtui pääosin **järjestelmätestauksella**. Järjestelmätestauksen tarkoituksena oli löytää lähdekoodista virheitä varmistuen, että toteutettu sovellus toimii suunnitellusti sekä sovellus täyttää sille asetetut toiminnalliset ja laadulliset vaatimukset.

Projektisuunnitelmassa otettiin kantaa myös **yksikkötestaukseen**. Projektissa oli tarkoitus yksikkötestata moduulit, joissa tapahtuu laskentaa. Projektin aikana ei kuitenkaan ohjelmoitu yksikkötestejä, koska suurin osa koodista liittyi käyttöliittymään ja valmiiden komponenttien käyttöön. Kun yksikkötestien kirjoittamisesta ei tullut vakiintunutta käytäntöä, jäivät yksikkötestit laatimatta myös moduuleille, joissa niitä olisi kenties kannattanut käyttää.

Järjestelmätestaukseen osallistuivat kaikki projektiryhmän jäsenet. Järjestelmätestauksesta vastaava henkilö laati **testaussuunnitelman** [12] ja vastasi testauksen läpiviennistä. Testaussuunnitelma sisälsi eri testauskerroilla suoritettavien testitapausten kuvaukset. Testauksesta vastannut henkilö kirjasi kullakin testauskerralla testaussuunnitelmaan käsin suoritettujen testitapausten tulokset sekä havaitut virheet ja mahdolliset puutteet. Projektin loppuvaiheessa suoritettiin testauksia säännöllisesti lähes päivittäin. Sovellusta testattiin suunnitelman pohjalta yhteensä 20 kertaa eri käyttöjärjestelmillä.

Testauksessa käytettävä **mittausdata** saatiin tilaajalta. Testauksen yhteydessä verrattiin kehitettävän sovelluksen tuloksia Finlandia-sovelluksella suoritettuihin varmistuen, että kehitetty sovellus toimi määritellysti antaen oikeat tulokset.

Myös sovelluksen käytettävyyteen kiinnitettiin huomiota. Sovelluksesta pyrittiin kehittämään mahdollisimman käyttäjäystävällinen ja helppokäyttöinen. Varsinaista **käytettävyydestä ei** kuitenkaan erikseen **järjestetty**, vaan käytettävyydsasiantuntija testasi sovelluksen prototyyppejä 10.4.2013 antaen muutos- ja parannusehdotuksia käyttöliittymään ja sovelluksen käytettävyyteen. Myös vastaava ohjaaja koeikäytti sovellusta kaksi kertaa ja antoi projektiryhmälle palautetta käyttöliittymästä.

Projektisuunnitelman mukaan projektiryhmän olisi pitänyt toimittaa tilaajalle projektin aikana kaksi testaustarkoitukseen varattua versiota. Tällaisia versioita ei kuitenkaan toimitettu erikseen, vaan tilaajalla oli oikeudet versiohallintaan. Tilajista Jaakko Julin latsi säännöllisesti sovelluksen uusinta versiota ja antoi palaverissa palautetta sovelluksesta. Lisäksi palaverissa käsiteltiin sovelluksen toimintaa ja

varmistuttiin siitä, että se toimii määritellysti. Samalla tilaaja antoi palautetta sovelluksen ulkoasusta ja toteutetuista ominaisuuksista.

5.7 Versiohallinta ja -numerointi

Tulosten versiohallintaan käytettiin **Git-versiohallintaohjelmistoa**. Sovelluksen lähdekoodi sijoitettiin Git-pohjaiseen **YouSource-julkistusjärjestelmään**, josta se oli koko ajan myös asiakkaan edustajien ja ohjaajien saatavilla. Projektissa laadittujen dokumenttien uusimpia versioita julkistettiin saataville projektin WWW-sivuilla. Dokumentteille ei ollut erillistä versiohallintaa, mutta projektissa laaditut dokumentit aina muokattaessa numeroitiin uudella versionumerolla. Dokumenttien versiohallinta hoidettiin siis lokaalisti muokkaajan koneella, joka ei ollut paras vaihtoehto.

Julkistetuissa dokumenttien ja sovelluksen lähdekoodien versioissa suunniteltiin käytettäväksi **kolmiportaista versionumerointia**. Ryhmän sisäiset versiot aloitettiin versionumerosta 0.0.1, ja kunkin uuden version osalta kasvatettiin vähiten merkitsevää numeroa yhdellä. Tällöin toinen versio oli versionumeroltaan 0.0.2. Projektiorganisaatiolle julkistettavien versioiden numerointi aloitettiin versionumerosta 0.1.0. Seuraavat versiot numeroitiin kasvattamalla toisen tason numeroa yhdellä. Ensimmäisen hyväksytyyn version numero oli 1.0.0, ja sitä seuraavissa hyväksytyissä versioissa kasvatettiin toisen tason numeroa yhdellä (siis toinen hyväksytty versio oli 1.1.0).

Toteuma poikkesi siis suunnitelmasta dokumenttien versiohallinnan osalta. Dokumentteja ei viety suunnitellusti versiohallintaan, vaan ne pidettiin lokaalisti laatijan koneella. Uusin versio oli saatavilla verkkosivulta. Tämä ei aiheuttanut haittaa projektille, sillä dokumenteilla oli aina vain yksi laatija ja muokkaaja. Kuitenkin, jos projektin aikana olisi tullut tilanteita, joissa jonkun jäsenen olisi pitänyt päästä käsiksi toisen jäsenen laatimaan vanhaan versioon, olisi huonosta käytänteestä ollut haittaa projektin läpiviennille.

Myöskään lähdekoodiversioiden hallinta ei mennyt suunnitelman mukaan. Suunnitelmassa määritettyä kolmiportaista versionumerointia ei käytetty, koska sovelluksesta ei julkistettu väliversioita. Lähdekoodit olivat kaikki saman nimisiä tiedostoja, eikä kommentteihin kirjattu kulloista versiota. Tämä olisi mahdollisesti auttanut joissain pienissä päällekkäisyyksissä liittyen versiohallinnan konflikteihin. Vaikutus

oli kuitenkin niin vähäinen, että se ei vaikuttanut lainkaan projektin läpivientiin, eikä tuloksiin.

5.8 Katselmoinnit ja tulosten hyväksyminen

Projektin jäsenten kirjoittama lähdekoodi **katselmoitiin** kaksi kertaa projektin aikana. Katselmoinnissa tekninen ohjaaja kommentoi lähdekoodia antaen vinkkejä ja parannusehdotuksia. Tekninen ohjaaja myös hyväksyi lähdekoodin. Katselmointiin osallistui teknisen ohjaajan lisäksi koko projektiryhmä ja tilaajan edustajista Jaakko Julin. Kummankin katselmoinnin havainnot projektiryhmä kirjasi **muistioksi**.

Projektin lopussa tulokset kokonaisuutena **hyväksytettiin** projektin ohjaajilla sekä tilaajan edustajista Timo Sajavaaralla. Yksittäisistä tuloksista tilaajan edustajan hyväksyntä tarvittiin toteutetulle sovellukselle ja sovellusraportille. Tekninen ohjaaja hyväksyi lähdekoodin yhdessä Jaakko Julinin kanssa. Vastaava ohjaaja hyväksyi projektin keskeisimmistä dokumenteista vaatimusmäärittelyn, projektisuunnitelman, projektiraportin ja sovellusraportin.

Katselmointien ja tulosten hyväksymisen toteuma vastasi suunnitelmaa.

5.9 Tulosten koostaminen ja toimittaminen

Projektiryhmä kokosi luvussa 3.4 mainitut projektin tulokset sekä erilliseen **projektikansioon** että **CD-levylle**. Projektikansioon sijoitettiin kaikki projektissa laaditut dokumentit ja lähdekoodilistaukset. Lisäksi sähköpostiarkistot, tiivistelmä projektista ja jäsenten itsearvioinnit liitettiin projektikansioon ja CD-levylle. CD-levylle tallennettiin edellisten lisäksi myös kehitetty sovellus.

CD-levy koottiin vasta, kun kaikki projektin tulokset oli hyväksytty. Tulokset toimitettiin tilaajalle CD-levyllä ja projektikansiossa. Laitokselle toimitettiin projektikansio kera projekti-CD:n. Toinen CD-levy toimitettiin laitoksen arkistoon. Projektikansio sijoitettiin projektitilan kokoushuoneessa olevaan kirjahyllyyn.

Tulosten koostamisen ja toimittamisen osalta toteuma vastasi suunnitelmaa.

6 Tehtävät, työmäärät ja työnjako

Luvussa esitellään projektipäällikkö ja varapäällikkö sekä jäsenten vastuualueet ja tehtävät. Projektisuunnitelmassa määritetyt vastuualueet pitivät paikkansa, ja jokainen ryhmän jäsen vastasi mallikkaasti omasta vastuualueestaan.

Lisäksi luvussa esitetään ryhmän jäsenten arvioidut ja toteutuneet työtunnit tehtäväkokonaisuuksittain. Projektiin käytettiin suunniteltu määrä työtunteja. Työtunnit eivät kuitenkaan jakautuneet tehtäväkokonaisuuksittain täsmälleen suunnitelman mukaan. Varsinkin jäsenten välinen työtehtävien jako ei vastannut suunnitelmaa ja yksi ryhmän jäsenistä ei päässyt ohjelmoimaan ollenkaan projektin aikana. Vähän tai ei lainkaan ohjelmoineet henkilöt keskittyivät projektin hallintaan sekä analyysikomponentteihin liittyvien ongelmien ratkaisuun ja järjestelmätestaukseen.

6.1 Projektin jäsenten vastuualueita

Projektipäällikkönä toimi Samuli Kärkkäinen ja varapäällikkönä Jarkko Aalto. Projektipäällikön vastuulla olivat projektin suunnittelu, hallinta ja raportointi, tilan määrittäminen ja tiedotus sekä työnjako ja kyseisen työajan pohjalta tehtävien jakaminen.

Ryhmän jäsenten vastuualueet oli määritelty **olennaisten tulosten mukaisesti** taulukossa 6.1. Tuloksesta vastuussa olevat jäsenet vastasivat sen valmistumisesta, asiaan liittyvästä tiedottamisesta ja sen tarkastukseen toimittamisesta. Vastuuhenkilö vastasi myös tarkastuksessa mahdollisesti ilmenneiden puutteiden korjaamisesta. Henkilö ei kuitenkaan yksin vastannut tuloksen toteuttamisesta, vaan jakoi tarvittaessa siihen liittyviä tehtäviä muille ryhmän jäsenille.

Varapäällikkö hoiti projektipäällikön tehtäviä projektin aikana kerran muutaman päivän ajan, kun varsinainen projektipäällikkö oli ulkomailla. Muuten projektin hallinnasta vastasi varsinainen projektipäällikkö.

Tulos	Vastuuhenkilö
Projektisuunnitelma	Samuli Kärkkäinen
Vaativuusmäärittely	Timo Konu
Sovelluksen toimintalogiikka	Timo Konu
Käyttöliittymä	Samuli Rahkonen
Järjestelmätestaus	Jarkko Aalto
Sovellusraportti	Miika Raunio
Projektiraportti	Samuli Kärkkäinen

Taulukko 6.1: Vastuualueet keskeisimpien tulosten osalta.

Projektin vastuualueiden jaon toteuma vastasi suunnitelmaa. Suunniteltu vastuuhenkilö vastasi omasta vastuualueestaan projektin aikana. Projektisuunnitelman sekä projekti-, sovellus- ja testausraporttien kirjoittajien ei tarvinnut jakaa työtehtäviä muille ryhmän jäsenille. Sen sijaan käyttöliittymän ja sovelluksen toimintalogiikan suunnitteluun ja toteutukseen osallistuivat aktiivisesti kaikki ryhmän jäsenet.

6.2 Tehtäväkohtaiset työmäärät ja työnjako

Taulukossa 6.2 on esitetty **suunnitellut (S)** ja **toteutuneet (T) työtunnit** jokaiselle ryhmän jäsenelle **tehtäväkohtaisesti**. Työmäärien suunnittelu ja arviointi perustui **edellisten sovellusprojektien toteutuneisiin työmääriin** sekä projektipäällikön tekemiin **valistuneisiin arvauksiin**. Arvioinnit vastasivat toteumaa projektin hallinnan ja oheiskurssien suhteen. Muissa tehtäväkokonaisuuksissa toteutuneet tunnit poikkesivat suunnitellusta. Projektiryhmän jäsenet sitoutuivat projektiin, ja suunnitelma oli riittävän tarkka projektin läpiviennin kannalta.

Suunnitelman mukaan projektiryhmän jäsenet käyttäisivät itse projektiin 12 opin-
topisteen verran työtä. Tämä tarkoittaa työtunneissa noin 321 tuntia per ryhmän
jäsen vastaten 1605 projektityötuntia koko ryhmällä. Oheiskurssit mukaan luettuna
projektiin oli suunniteltu käytettävän yhteensä 1960 tuntia. Projektin ja oheiskurs-
sien suunniteltu tuntimäärä ylittyi noin 240 työtunnilla, joten toteutunut tuntimäärä
ylittyi noin 12 prosentilla. Tämä johtuu varsinkin esitutkimuksen ja järjestelmät-
tauksen suunniteltua suuremmista työtuntimääristä.

Projektin hallinta muodostui pitkälti projektipäällikön työtunneista. Suunnitelmas-

sa pieniä määriä työtunteja oli jaettu kaikille ryhmän jäsenille, mutta hallinnan tehtävät keskitettiin projektipäällikölle. Muut projektiryhmän jäsenet keskustelivat kylä projektipäällikön kanssa projektin läpiviennistä, mutta he merkkasivat tuntinsa pitkälti sen työtehtävän alle, jota olivat kulloinkin tekemässä.

Palavereiden toteutunut työmäärä oli hieman suurempi kuin suunniteltu. Palavereita oli 11 suunnitellun kymmenen sijaan, sekä pöytäkirjojen kirjoittamiseen ja palavereihin valmistautumiseen meni ryhmän jäseniltä enemmän aikaa kuin projektipäällikkö oli suunnitelmassaan arvioinut.

Tehtäväkokonaisuuden **esitutkimus** nimi ei ole kovin kuvaava työlle, jota merkittiin esitutkimukseen ja kohdealueeseen tutustumiseen. Suuri osa esitutkimukseen merkityistä tunteista oli tilaajan toimittamien analyysikomponenttien kääntämistä ja niihin liittyvien ongelmien selvittelyä. Projektiryhmän kaksi jäsentä joutuivat käyttämään paljon suunniteltua enemmän aikaa komponenttien kääntämiseen eri käyttöjärjestelmillä.

Vaatusmäärittelyyn sen sijaan käytettiin huomattavasti vähemmän aikaa kuin oli suunniteltu. Tämä johtuu siitä, että osa vaatimusmäärittelyyn käytetyistä työtunneista kirjattiin kirjoitusviestintään, koska sen yhdessä tehtävässä laadittiin ajatuskartta sovelluksen vaatimuksista. Näitä vaatimuksia projektiryhmä mietti yhdessä projektihuoneessa, ja täten loi kattavan pohjan viralliselle vaatimusmäärittelylle, joka laadittiin lähes suoraan käsitekarttojen pohjalta.

Suunnittelun ja toteutuksen toteutuneet työtunnit poikkeavat osa-alueittain merkittävästi projektisuunnitelman arvioista. Suunnittelua ja toteutusta kirjattiin ehkä hieman epäloogisesti. Joskus suunnittelu kirjattiin toteutukseen ja toisin päin. Jos vaatimusmäärittelyn, suunnittelun, toteutuksen sekä testauksen suunnitellut ja toteutuneet tunnit lasketaan yhteen, poikkeama on erittäin pieni. Suunnitelmassa tunteja näille neljälle tehtäväkokonaisuudelle oli varattu 754 tuntia ja toteumassa tunteja kertyi yhteensä 790. Tehtäväkokonaisuuksien toteuman arvionnissa onnistuttiin siis hyvin.

Järjestelmätestaukseen työtunteja käytettiin suunniteltua enemmän. Lisäksi järjestelmätestaukseen varatut tunnit ovat toteumassa keskittyneet Aallolle. Tämä johtuu siitä, että Aalto teki huomattavasti projektipäällikön suunnitelmaa kattavamman testaussuunnitelman. Suunnitelman mukaan testaussuunnitelmaan olisi käytetty 25 tuntia, kun toteumassa työmäärä oli yli 100 tuntia. Aalto myös suoritti testauskerrat ja raportoi tulokset muulle projektiryhmälle.

Viimeistelyn työtuntien toteuma on niin ikään suurempi kuin suunniteltu. Tämä johtuu puhtaasti siitä, että projektipäällikkö oli suunnitelmassaan arvioinut tunteja liian vähäisiksi.

Oheiskursseihin yhteensä käytetty tuntimäärä vastaa suunnitelmaa. Puhe- ja kirjoitusviestintään on kirjattu myös vaatimusmäärittelyyn käytettyjä tunteja sekä kaikki esittelyt ja niiden muistioiden laatimiset. Myös Sovellusprojekti-kurssiin liittyvään opetukseen käytettyjen työtuntien toteuma vastasi suunnitelmaa.

Yksittäisten jäsenien kokonaistyötunneissa on noin 30 tunnin ero. Varsinaisesta ohjelmoinnista vastanneet Konu ja Rahkonen tekivät muita enemmän töitä. Tämä johtuu osittain myös siitä, että he tekivät töitä vapaa-ajallaan ja viikonloppuisin. Muiden jäsenten työtunnit keskittyivät lähinnä arkipäiville. Projektin aikana uskottiin työtuntien tasaantuvan loppua kohti, koska ohjelmointi päättyisi ennen dokumenttien valmistumista. Tasaantumista ei kuitenkaan juuri tapahtunut, sillä sovellusta ja lähdekoodia viimeisteltiin lähes projektin päättymiseen saakka.

Ryhmän jäsenet merkkasivat tunnollisesti projektiin käyttämiään työtunteja. Ryhmän olisi kuitenkin kannattanut heti projektin alussa sopia työtuntien **kirjauskäytännöistä** hieman tarkemmin. Jos tunnit olisi kirjattu projektisuunnitelman nimikkeillä, olisi työtuntien vertailu ollut projektipäällikölle helpompaa. Heikot kirjauskäytänteet johtuvat myös osittain siitä, että projektisuunnitelma valmistui vasta projektin keskivaiheilla, 16.4.2013. Tätä ennen tapahtuneet kirjaukset johtivat huonojen kirjauskäytänteiden vakiintumiseen. Projektipäällikkö joutui projektin aikana useaan otteeseen korjaamaan tuntien kirjauksia vastaamaan suunnitelman tehtäviä ja huomauttamaan kirjausten virheellisyydestä. Onneksi työtehtäville oli aina selkeä kuvaus, joten ”kummallisesti” merkitylle työajalle löytyi yleensä helposti kuvaavampi tehtävä. Tulevissa sovellusprojekteissa tätä kannattaa korostaa opiskelijoille, jotta ajankirjaussovelluksen [1] käyttäminen olisi helpompaa. Sama huomio on tehty myös Paatin projektiraportissa [5], mutta ryhmää ei ohjeistettu tarpeeksi asian suhteen.

Kuten taulukosta 6.2 käy ilmi, tehtäväjako ei ollut suunnitelman mukainen, vaan tehtävät keskittyivät eri henkilöille. Konu ja Rahkonen aloittivat projektin alussa varsinaisen suunnittelun ja toteutuksen, kun taas Kärkkäinen kirjoitti projektisuunnitelmaa ja keskittyi projektin hallintaan. Raunio ja Aalto sen sijaan syventyivät tilaajan toimittamiin analyysikomponentteihin. Kun näiden analyysikomponenttien kanssa ilmaantui ongelmia, jatkuivat Raunion ja Aallon työtehtävät analyysikomponenttien parissa pidempään kuin oli suunniteltu. Selvitettyään ongelmat analyys-

sikomponentteihin liittyen, Aalto siirtyi laatimaan testausraporttia ja Raunio ohjelmoimaan syvyysprofiileja. Tämän ajan Rahkonen ja Konu olivat jo toteuttaneet sovellusta. Kun ongelmia analyysikomponenttien kanssa aika ajoin ilmeni, siirrettiin Raunio ja Aalto ratkaisemaan näitä ongelmia.

Rahkonen vastasi käyttöliittymästä tehden suunnitelmat sekä toteuttaen käyttöliittymän päänäkymän ja asetusvalikot. Lisäksi Rahkonen keskittyi vaikeaksi osoittautuneeseen lentoaikakalibraatioon. Konu toimi sovelluksessa pääohjelmoijana. Hän vastasi sovelluksen kokonaisarkkitehtuurista sekä toteutti eri osakokonaisuuksia, kuten ToF–E -histogrammin, alkuainevalinnan ja energiaspektrin. Lisäksi hän suunnitteli ja toteutti mittausprojektin hallintaa.

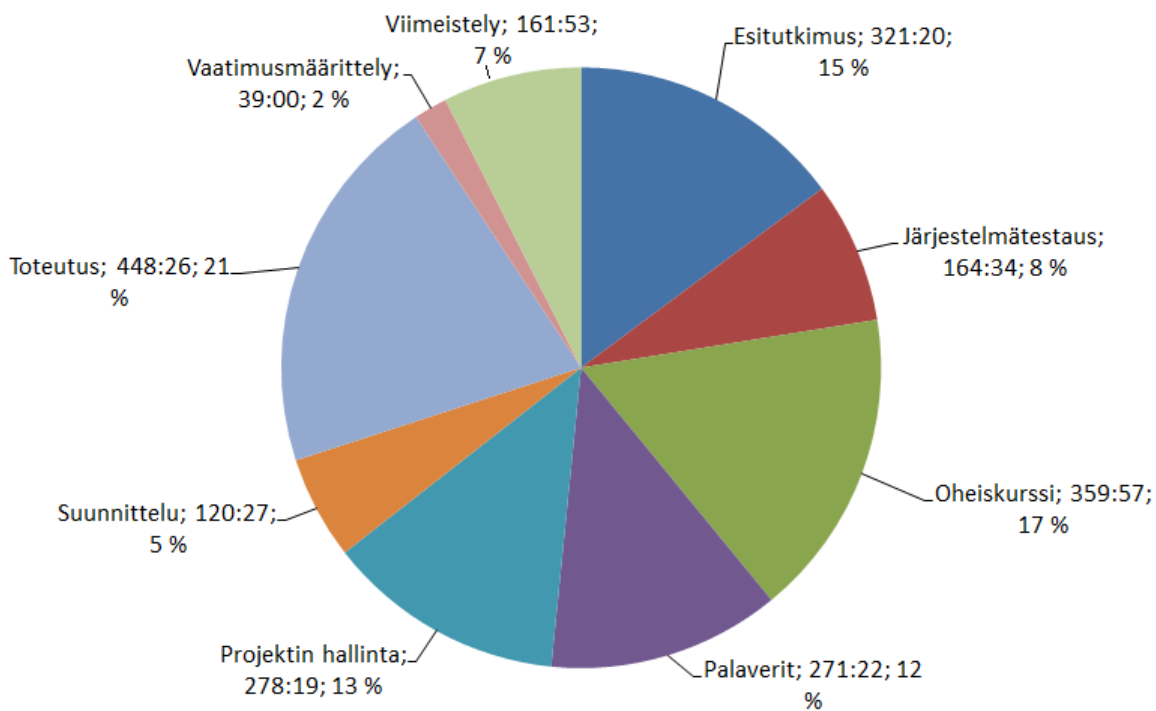
Kärkkäinen toimi projektipäällikkönä, joten projektin hallintaan liittyvät asiat olivat hänen vastuullaan. Hän toteutti sovellukseen lokikomponentin ja suunnitteli Python-tulkin yhdistämistä sovellukseen. Projektipäällikön tehtävät vaativat paljon huomiota, joten osin myös tämän takia Python-tulkin käyttö jouduttiin sopimaan jatkokehitykseen.

Tehtäväkokonaisuus	Tehtävä	JA		TK		SK		SR		MR		Kaikki	
		S	T	S	T	S	T	S	T	S	T	S	T
Projektin hallinta													
	Suunnittelu	5	0	0	0	35	40	0	0	0	0	40	40
	Projektisuunnitelma	2	0	2	0	40	57	2	0	2	0	48	57
	Seuranta	5	1	0	5	15	17	0	1	0	1	20	25
	Tiedotus	2	1	2	6	10	9	2	0	2	5	18	21
	Muut tehtävät	4	1	4	9	4	6	4	12	4	7	20	35
	Projektiaportti	3	3	3	2	40	62	3	2	3	1	52	70
	Tulosten viimeistely	5	3	5	10	10	7	15	3	5	0	40	23
	Tulosten kokoaminen	4	0	8	0	4	3	4	0	4	0	24	3
	Yhteensä	30	9	24	32	158	201	30	18	20	14	262	274
Palaverit													
	Valmistelu	10	4	10	15	15	16	10	13	10	4	55	52
	Palaverit	20	24	20	24	20	24	20	24	20	24	100	120
	Pöytäkirjat	15	35	15	15	15	21	15	23	15	12	75	106
	Yhteensä	45	63	45	54	50	61	45	60	45	40	230	278
Esitutkimus													
	Aihealueeseen tutustuminen	30	73	15	16	15	17	30	16	15	77	105	199
	Koulutus	10	2	10	2	10	2	10	2	10	2	50	8
	Työkaluihin tutustuminen	20	42	40	31	10	2	20	27	20	14	110	116
	Yhteensä	60	117	65	49	35	21	60	45	45	93	265	325
Vaatimusmäärittely													
	Suunnittelu	10	3	10	5	10	7	10	3	10	3	50	21
	Raportointi	0	0	10	19	4	0	0	0	2	0	16	19
	Yhteensä	10	3	20	24	14	7	10	3	12	3	66	40
Suunnittelu													
	Kokonaisarkkitehtuuri	2	0	15	21	2	1	2	5	2	1	23	28
	Käyttöliittymä	5	0	5	3	5	1	25	33	5	0	45	37
	ToF-E -histogrammi	2	0	10	0	0	0	2	0	2	0	16	0
	Alkuainevalinta	2	0	10	0	0	0	2	0	2	8	16	8
	Kalibrointi	5	0	0	0	3	0	5	13	10	2	23	15
	Alkuaineiden poistumat	0	0	0	0	3	0	5	0	10	0	18	0
	Syvyyso profiili	10	0	2	0	2	1	2	0	2	9	18	10
	Energiaspektri	5	0	10	0	0	0	0	0	5	0	20	0
	Raportointi	5	0	0	0	3	0	0	0	0	0	8	0
	Komentorivitoiminnallisuus	5	0	5	0	5	9	5	0	15	0	35	9
	Mittausprojektin hallinta	5	0	5	0	0	13	5	3	10	4	25	20
	Yhteensä	46	0	62	24	23	25	53	54	63	24	247	127
Toteutus													
	Käyttöliittymä	5	0	5	22	5	0	35	99	5	0	55	121
	ToF-E -histogrammi	5	0	10	18	0	0	2	0	0	0	17	18
	Alkuainevalinta	0	0	10	49	0	0	2	1	2	1	14	51
	Kalibrointi	5	0	10	6	0	0	10	66	10	0	35	72
	Alkuaineiden poistumat	0	0	6	13	0	0	20	3	0	0	26	16
	Syvyyso profiili	20	2	15	5	0	0	15	4	4	65	54	76
	Energiaspektri	10	0	10	32	0	0	0	0	6	0	26	32
	Raportointi	5	0	0	0	2	0	2	0	0	0	9	0
	Komentorivitoiminnallisuus	15	0	5	0	5	3	5	0	30	0	60	3
	Mittausprojektin hallinta	5	0	5	25	2	19	5	4	10	16	27	64
	Yhteensä	70	2	76	170	14	22	96	177	67	82	323	453
Järjestelmätestaus													
	Suunnittelu	25	108	2	0	2	4	2	2	2	2	33	116
	Testauskerrat	10	43	10	0	10	3	10	1	10	3	50	50
	Raportointi	15	1	5	0	5	0	5	2	5	1	35	4
	Yhteensä	50	152	17	0	17	7	17	5	17	6	118	170
Viimeistely													
	Katselmoinnit	4	6	4	7	4	8	4	6	8	10	24	37
	Lähdekoodin viimeistely	4	0	4	26	4	2	4	14	4	5	20	47
	Sovellusraportti	2	0	2	0	2	1	2	0	40	80	48	81
	Sovelluksen luovutus	0	0	2	0	0	0	0	0	0	0	2	0
	Yhteensä	10	6	12	33	10	11	10	20	52	95	94	165
	Projektin tunniti yhteensä	321	352	321	386	321	355	321	382	321	357	1605	1832
Oheiskurssit													
	Puhe- ja kirjoitusviestintä	51	56	51	50	51	54	51	57	51	52	255	269
	Sovellusprojektin hallinta	20	20	20	18	20	19	20	20	20	19	100	96
	Yhteensä	71	76	71	68	71	73	71	77	71	71	355	365
	Projektin ja oheiskurssien tunniti yhteensä	392	428	392	454	392	428	392	459	392	428	1960	2197

Taulukko 6.2: Tehtävien suunnitellut ja toteutuneet työtunnit.

6.3 Ryhmän työtunnit tehtäväkokonaisuuksittain

Kuvassa 6.1 on esitetty projektiryhmän kirjatut työtunnit tehtäväkokonaisuuksittain. Esitutkimuksen suuri osuus johtuu siitä, että tutustuminen tilaajan lähettämiin analyysikomponentteihin merkittiin esitutkimukseksi. Kun analyysikomponenttien kanssa oli ongelmia, kirjattiin ongelman ratkaisemiseen käytetyt tunnit esitutkimukseen.



Kuva 6.1: Ryhmän työtunnit tehtäväkokonaisuuksittain.

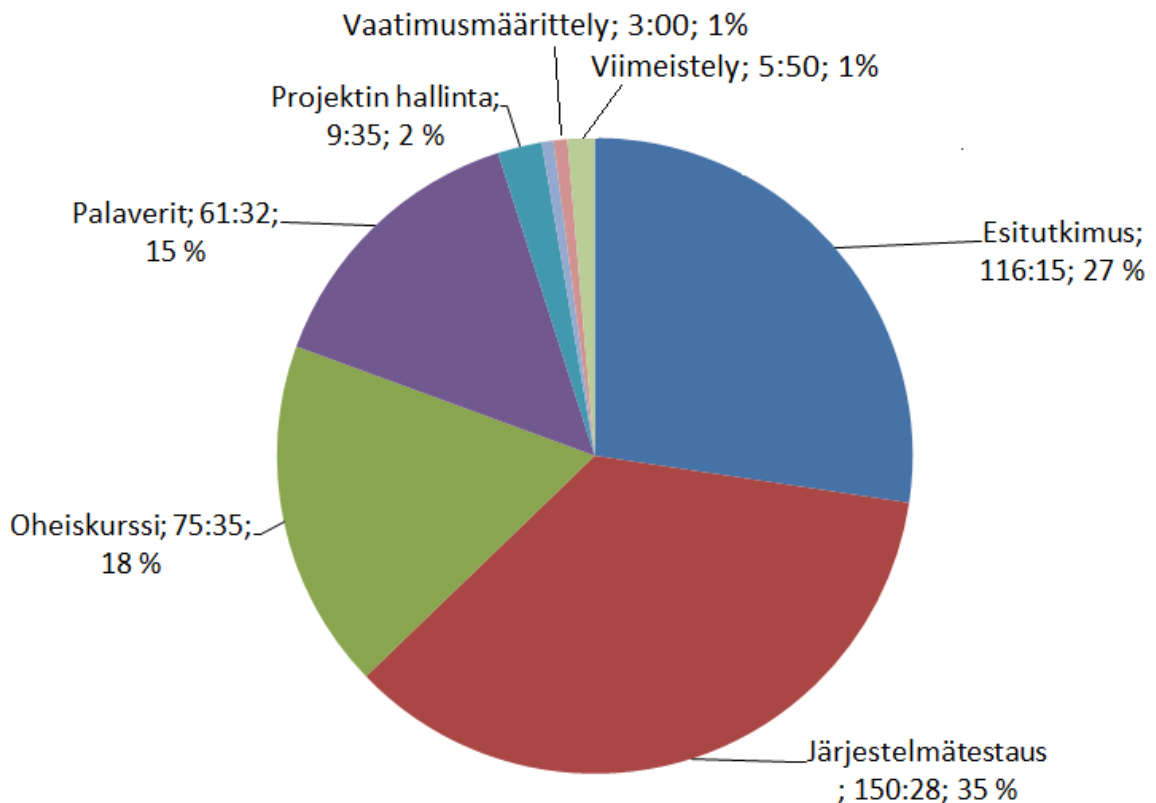
Vaatusmäärittelyn osuus vaikuttaa hyvin pieneltä. Myös suunnittelun osuus suhteessa toteutukseen on melko vähäinen. Suunnittelua ja vaatusmäärittelyä tarkennettiin kuitenkin myös palavereissa, kun projektiryhmä kävi tilaajan kanssa läpi kohdealueeseen liittyviä asioita ja vaatimusten tilaa sekä suunnitteli ja tarkensi sovellukseen toteutettavia ominaisuuksia.

Projektin hallintaa hoiti käytännössä ainoastaan projektipäällikkö. Muun muassa projektisuunnitelman ja -raportin kirjoittaminen kirjattiin projektin hallintaan. Lisäksi se sisältää projektin etenemisen seurannan ja suunnittelun sekä versiohallinnan toteuttamisen.

Järjestelmätestauksen osuus on melko suuri. Tämä johtuu projektin aikana laaditusta testaussuunnitelmasta. Sovellusta myös testattiin eri käyttöjärjestelmillä useaan otteeseen projektin loppuvaiheessa, joten järjestelmätestauksen osuus muodostui suureksi.

6.4 Jarkko Aallon työtunnit tehtäväkokonaisuuksittain

Aalto oli ryhmästä ainoa, jolla oli työkoneessaan käytössä Linux-käyttöjärjestelmä. Tämän takia hän sai projektin alussa tehtäväkseen tutustua tilaajan toimittamien analyysikomponenttien toimintaan. Komponenttien käyttöönottamisen haasteista johtuen tutustumiseen meni suunniteltua huomattavasti enemmän aikaa. Tämä näkyy kuvassa 6.2 suurena määränä esitutkimusta.



Kuva 6.2: Aallon työtunnit tehtäväkokonaisuuksittain.

Esitutkimuksen jälkeen Aalto siirtyi suoraan järjestelmätestauksen suunnittelun pa-

riin. Hän laati kattavan järjestelmätestausdokumentin [12] ja vastasi jäsenistä testauksesta eri käyttöjärjestelmillä raportoiden havaintoja muun projektiryhmän tietoon.

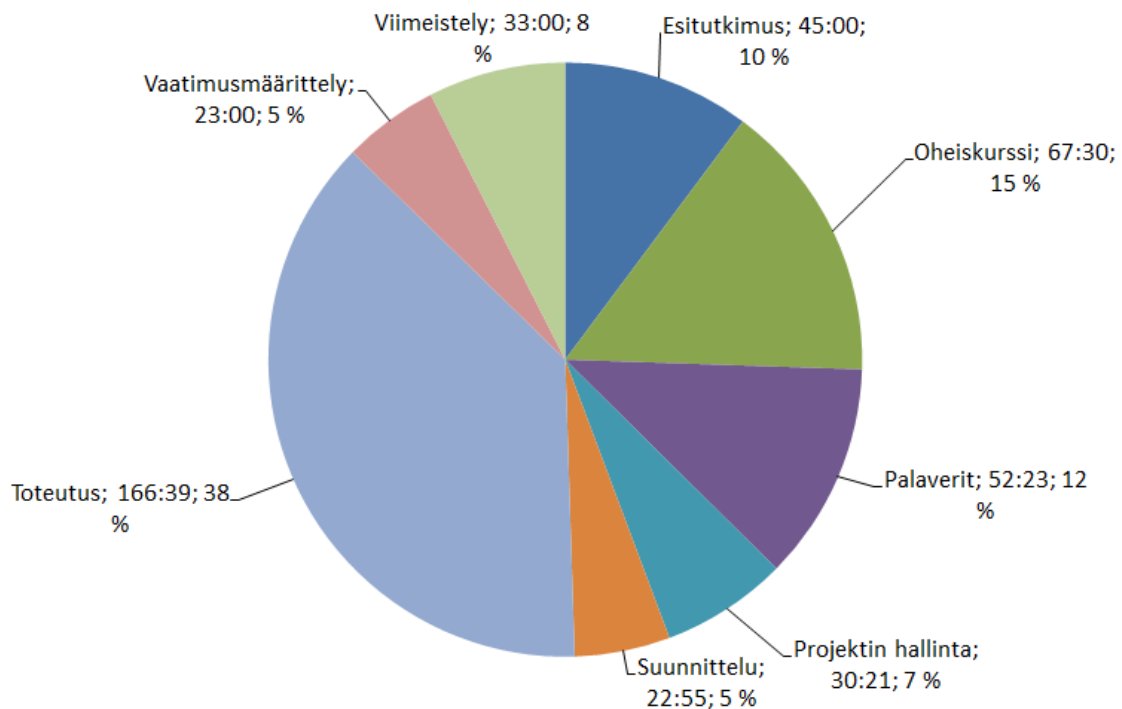
Koska esitutkimukseen ja testaukseen kului suunniteltua enemmän aikaa, ei Aalto ehtinyt kirjoittamaan koodia projektin aikana. Tämä selittää suunnittelun puuttumisen ja toteutuksen pienen osuuden.

Vaikka Aalto oli varapäällikkö, ei hänen tarvinnut tuurata projektipäällikköä kuin muutaman päivän ajan. Tämä näkyy kuviossa projektin hallinnan todella pienenä osuutena.

6.5 Timo Konun työtunnit tehtäväkokonaisuuksittain

Konu aloitti jäsenistä ensimmäisenä varsinaisen ohjelmoinnin. Tästä syystä Konu profiloitui yhdeksi sovelluksen pääohjelmoijista, joka näkyy kuvasta 6.3 suunnittelun ja toteutuksen suurena määränä. Suunnittelun ja toteutuksen epätasapainoinen suhde johtuu osin työtuntien kirjaamisesta. Lisäksi Konulla oli usein selkeä kuva siitä, mitä ja miten toteuttaa seuraavaksi.

Konulla on myös projektiryhmästä suurin osuus vaatimusmäärittelyä, sillä hän vastasi kyseisen dokumentin laatimisesta ja päivittämisestä. Konulla on myös ryhmän toiseksi suurin osuus projektin hallintaa. Tämä johtuu siitä, että hän vastasi myös versiohallinnasta ja siihen liittyvien oikeuksien jaosta. Nämä työtunnit kirjattiin projektin hallintaan.

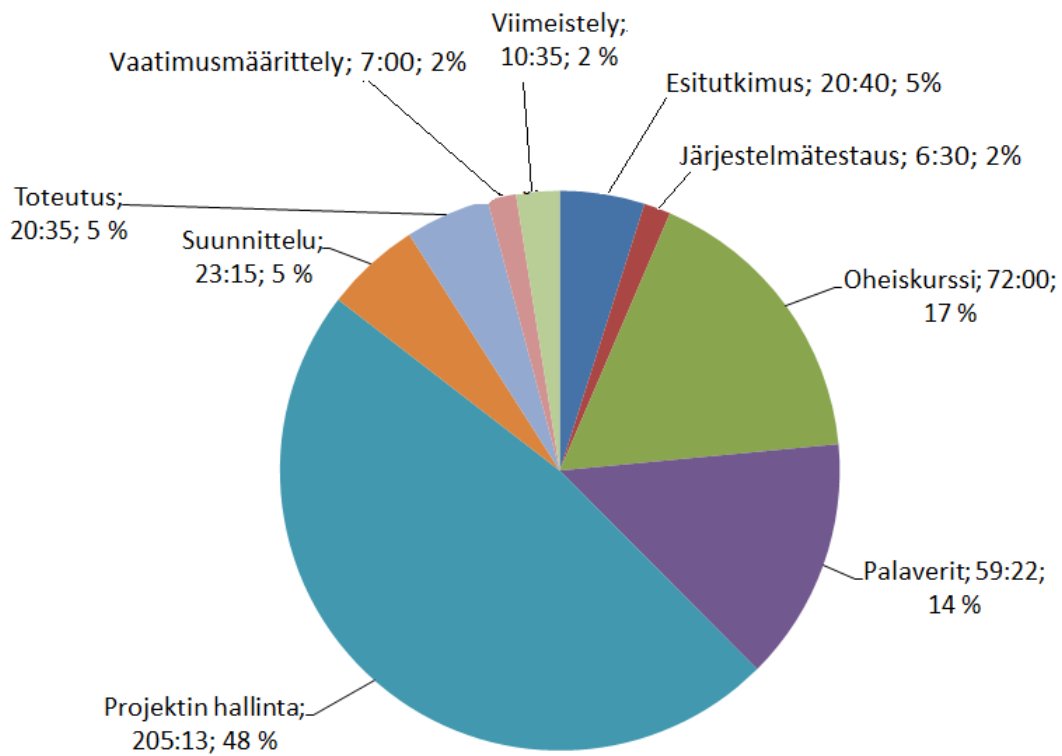


Kuva 6.3: Konun työtunnit tehtäväkokonaisuuksittain.

6.6 Samuli Kärkkäisen työtunnit tehtäväkokonaisuuksittain

Kärkkäinen toimi projektipäällikkönä laatien muun muassa projektisuunnitelman ja projektiraportin sekä seurasi, raportoi ja ohjasi projektin läpivientiä. Kuvassa 6.4 tämä näkyy selvästi muita suurempana osuutena projektin hallinnassa sekä pienenä osuutena esitutkimuksessa, suunnittelussa ja toteutuksessa.

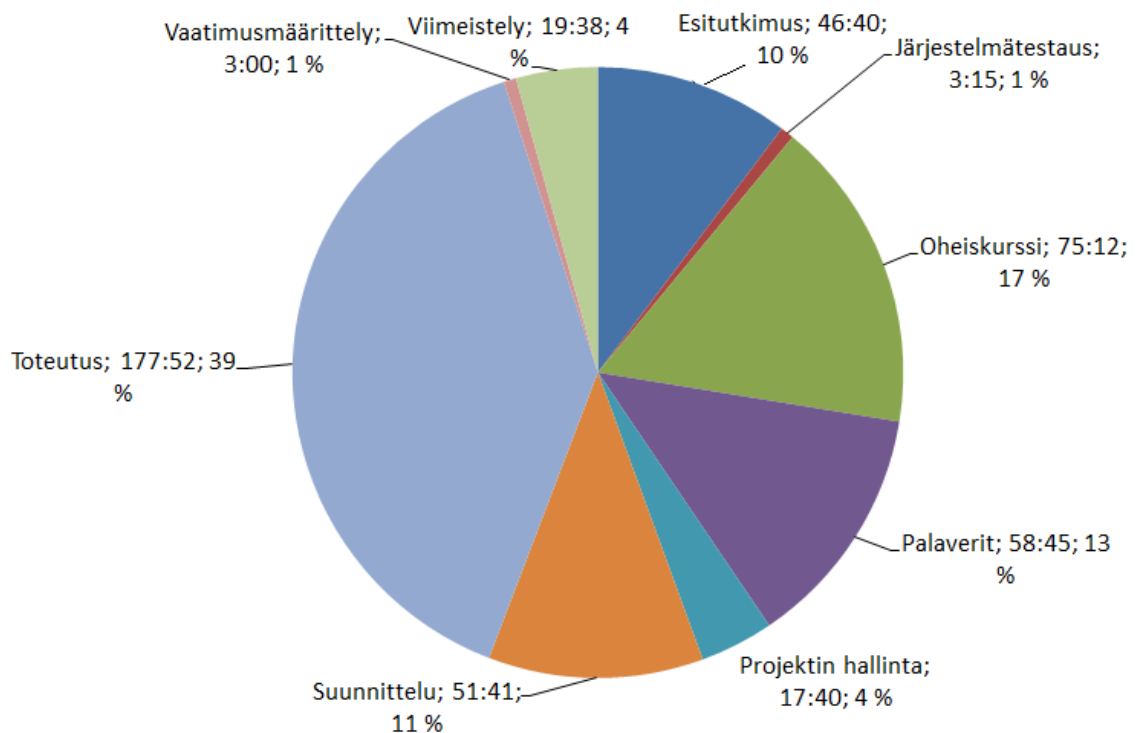
Projektin hallinta sisältää projektin suunnittelun, joten kaaviossa oleva suunnittelu on nimenomaan sovelluksen suunnitteluun käytettyä aikaa. Kärkkäisen toteutuksen osuus on pieni, ja toteutus koostuu lähinnä lokin toteuttamisesta. Koska Kärkkäinen ei ohjelmoinut paljon, ei hänelle kertynyt työkaluihin tutustumiseen liittyvää aikaa. Hän ei myöskään työskennellyt tilaajan toimittamien analyysikomponenttien kanssa, joten hänen esitutkimuksen osuutensa on huomattavasti muita jäseniä pienempi.



Kuva 6.4: Kärkkäisen työtunnit tehtäväkokonaisuuksittain.

6.7 Samuli Rahkosen työtunnit tehtäväkokonaisuuksittain

Rahkonen vastasi projektissa sovelluksen käyttöliittymästä ja tiettyjen sovelluksen osien, kuten lentoaikakalibraation. Tämän takia hänen esitutkimuksen, suunnittelun ja toteutuksen osuus on suuri, kuten kuvasta 6.5 käy ilmi. Rahkosen suunnittelun ja toteutuksen osuus voisi olla tasaisempikin, koska suuri osa työskentelystä oli käyttöliittymän hiomista, jolloin suunnittelutyötä tapahtui toteutuksen ohessa.

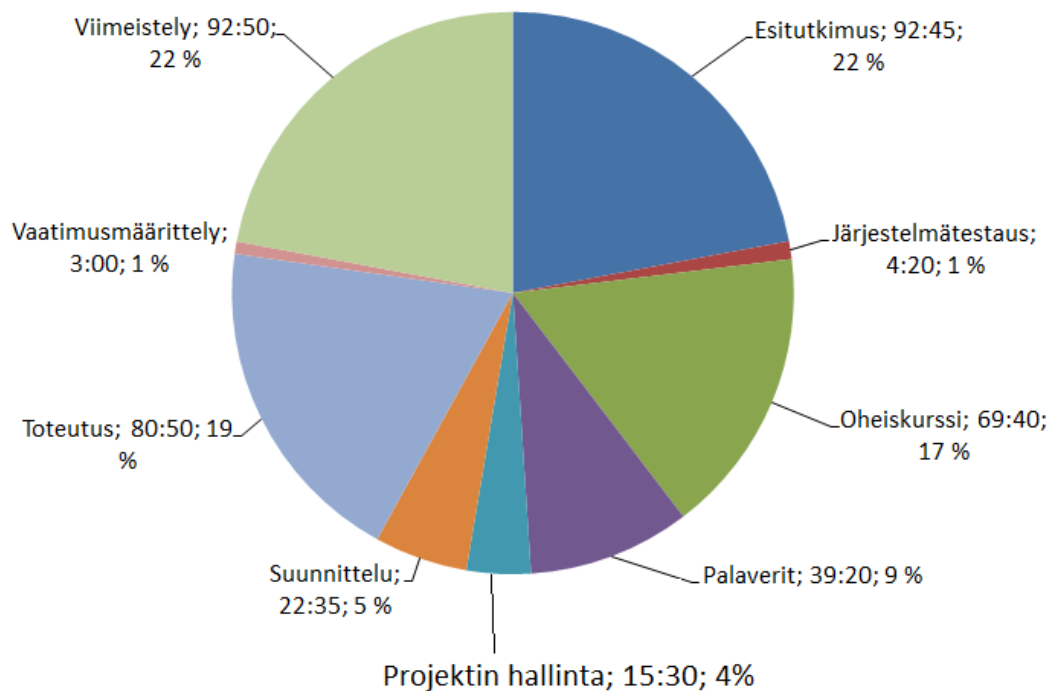


Kuva 6.5: Rahkosen työtunnit tehtäväkokonaisuuksittain.

6.8 Miika Raunion työtunnit tehtäväkokonaisuuksittain

Kuvasta 6.6 käy ilmi, että Raunio käytti työtunteja tasaisesti moneen eri osa-alueeseen. Hän työskenteli tilaajan toimittamien analyysikomponenttien kääntämiseksi Windowsille sekä toteutti käyttöliittymän syvyysprofiileille. Komponenttien kanssa työskentely näkyy kaaviossa suurena osuutena esitutkimusta. Syvyysprofiilien toteutus sen sijaan nostaa toteutuksen osuuden lähes keskimääräiseksi.

Raunion palaverien osuus on pieni. Tämä johtuu todennäköisesti siitä, että muut kirjasivat palaveriin valmistautumista enemmän kuin Raunio. Hän kuitenkin osallistui kaikkiin palavereihin sekä toimi sihteerinä ja puheenjohtajana kahdessa palaverissa. Tästä huolimatta hänen palavereihin käyttämänsä aika oli huomattavasti pienempi kuin muilla.



Kuva 6.6: Raunion työtunnit tehtäväkokonaisuuksittain.

7 Prosessi ja aikataulu

Luvussa kuvataan projektissa noudatettua prosessia ja toteutunutta aikataulua. Projektiin suunniteltua vaihejakoa ei tarkalleen ottaen noudatettu, koska toiseen vaiheeseen suunniteltua komponenttien yhdistämistä ei saatu suoritettua ennen kolmannen vaiheen alkua. Tämä ei kuitenkaan vaikuttanut projektin tuloksiin.

Projektia ei saatu päätettyä suunniteltuun päivämäärään 17.5.2013 mennessä. Kuitenkin projektille suunniteltu kymmenen arkipäivän pelivara riitti, ja täten projekti katsotaan päättyneeksi ajallaan.

7.1 Prosessi

Projektissa suunniteltiin käytettäväksi räätälöityä **prosessia**, jonka mukaan projektia kehitettäisiin ensisijaisesti **inkrementaalisesti** ja toissijaisesti **iteratiivisesti**. Tämä kehityskaari toteutui osittain projektissa, mutta inkrementaalisuus ja iteratiivisuus eivät esiintyneet projektissa niin selkeästi, koska projektin alussa muodostunut kuva sovelluksen toteutusratkaisusta poikkesi todellisuudesta.

Projekti oli tarkoitus läpiviedä neljässä selkeässä vaiheessa, jossa **ensimmäisessä vaiheessa** suunniteltaisiin projektin läpivienti, käyttöliittymä ja sovellusrakenne sekä laadittaisiin vaatimusmäärittely. Lisäksi kehitettäisiin pieniä osakokonaisuuksia itse sovelluksesta. Ensimmäisessä vaiheessa tutustuttiin myös kohdealueeseen. Projektissa poikettiin suunnitellusta siltä osin, että kaikkia suunniteltuja pieniä osakokonaisuuksia ei saatu kehitettyä ensimmäisessä vaiheessa, vaan puuttuvat osakokonaisuudet kehitettiin myöhemmin toisessa ja kolmannessa vaiheessa. Tilaajan toimittamien analyysikomponenttien käyttö osoittautui haastavaksi, mikä vaikutti tiettyjen osakokonaisuuksien, etenkin syvyysprofiilien ja lentoaikakalibraation toteuttamiseen. Nämä osakokonaisuudet kutsuvat käyttöliittymästä ulkoisia komponentteja. Lisäksi käyttöliittymä eli koko projektin ajan, koska tilaajalta tuli siihen parannusehdotuksia ja uusien ominaisuuksien lisääminen kokonaisuuteen vaikutti myös käyttöliittymään. Käyttöliittymä saatiin kuitenkin kuntoon jo maaliskuun lopussa. Kohdealue osoittautui haastavaksi, ja siihen tutustuminen jatkui koko projektin ajan.

Projektin suunnitelman mukaan **toisessa vaiheessa** olisi kasattu inkrementaalisesti sovelluksen runko ensimmäisessä vaiheessa kehitetyistä osakokonaisuuksista ja

kolmannessa vaiheessa olisi kehitetty osakokonaisuuksia iteratiivisesti. Näissäkin vaiheissa suunnitelma ja toteutuma poikkesivat toisistaan. Vaiheet tapahtuivat tosiasiassa limittäin, ja sovellukseen ei lisätty kaikkia osakokonaisuuksia kerralla. Niitä kuitenkin lisättiin inkrementaalisesti, ja kutakin osakokonaisuutta kehitettiin sekä toisen että kolmannen vaiheen ajan iteratiivisesti.

Projektin **viimeinen vaihe** läpiviettiin suunnitellusti. Siinä sovellus ja lähdekoodi viimeisteltiin sekä suoritettiin järjestelmätestaus. Lisäksi laadittiin projekti- ja sovel-lusraportti sekä muodostettiin luokkadokumentaatio. Tässä vaiheessa varmistettiin järjestelmätestauksella sekä vastaavan ohjaajan koekäytön perusteella se, että sovel-lus täytti sille asetetut vaatimukset. Lopuksi hyväksyttiin tulokset ja toimitettiin tulokset tilaajalle.

7.2 Aikataulu

Ennen varsinaisen projektin alkua sovellusprojektikurssilla oli projektiviestinnän luentoja sekä projektien aloitusluento. **Projektin ensimmäinen tapaaminen** tilaa-jan kanssa oli 7.2.2013, jolloin projekti katsotaan alkaneeksi.

Projektin tehtäväkokonaisuuksien ja tehtävien suunnitellut aikajänteet esitellään ku-vassa 7.1. Toteutuneet aikajänteet esitetään kuvan 7.2 Gantt-kaaviossa. Kuvista nä-kyy selvästi suunnitelman ja toteuman aikajänteiden poikkeamat. Varsinkin sovel-luksen suunnittelun toteutuneet aikajänteet olivat paljon suunniteltua lyhyempiä.

Ensimmäisen kolmen viikon aikana projektiryhmä tutustui kehitettävän sovelluk-sen taustoihin ja kohdealueeseen yleensä. Kehitystyökalujen ja ohjelmointikielen valinta viivästyi hieman, sekä sopivan teknisen ohjaajan löytymiseen kului kolme viikkoa aikaa. Tekninen ohjaaja valittiin 25.2.2013.

Suunnitelman mukaan helmikuun loppu ja maaliskuun alku varattiin työkaluihin sekä kohdealueeseen tutustumiselle. Kohdealueeseen tutustuivat kaikki jäsenet. Työ-kaluihin tutustuminen ei kuitenkaan vastannut suunnitelmaa, sillä projektin työn-jako edellytti jäsenten keskittymisen eri asioihin. Konu ja Rahkonen aloittivat ohjel-moinnin, Kärkkäinen projektisuunnitelman sekä Raunio ja Aalto keskittyivät ana-lyysikomponentteihin. Jokainen tutustui työkaluihin, kun se hänen kohdallaan oli tarpeellista.

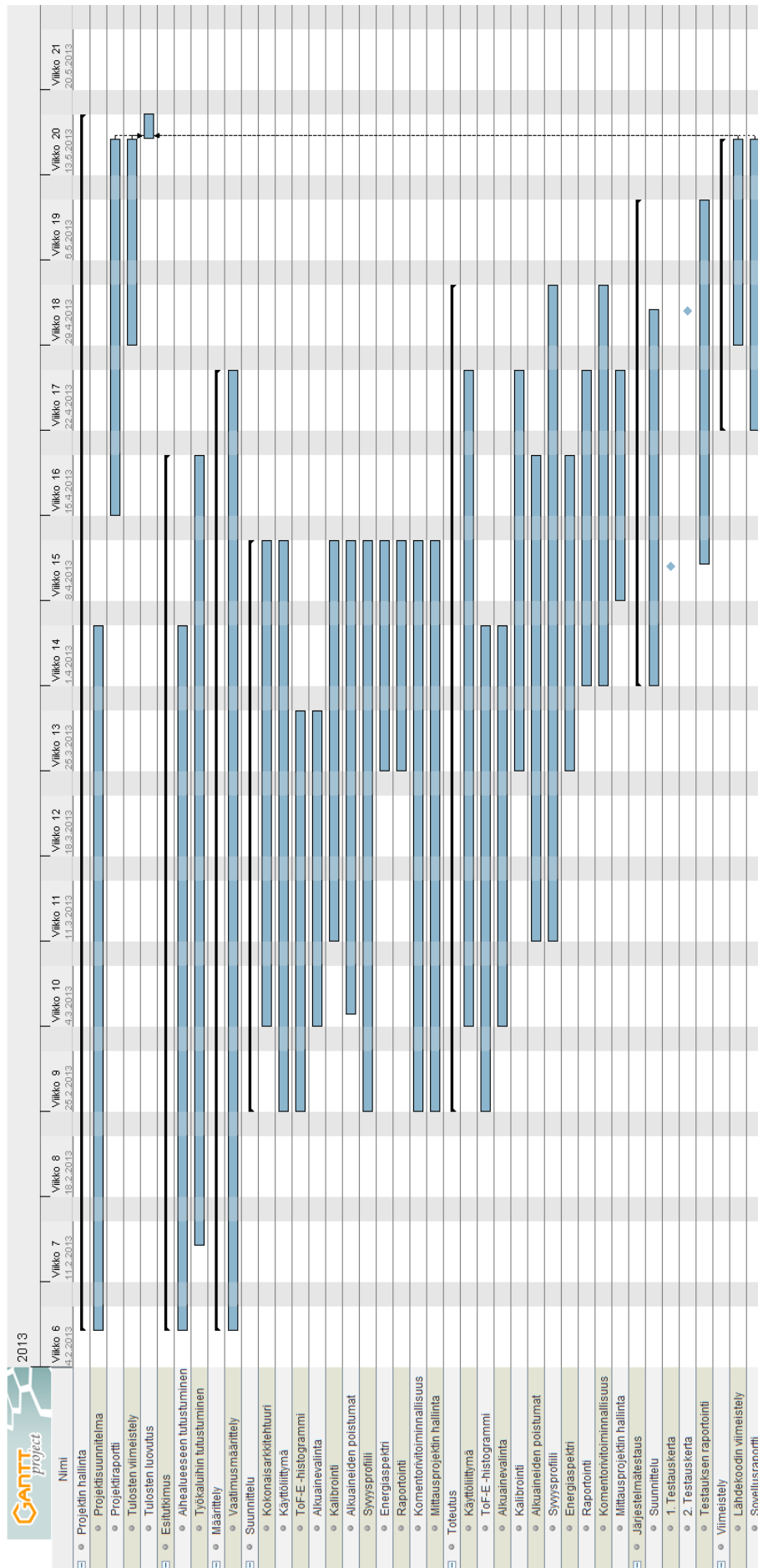
Varsinaisen **sovelluksen kehittäminen** alkoi 19.2.2013, ja uusien ominaisuuksien li-sääminen lopetettiin suunnitellusti 30.4.2013. Toukokuu oli varattu sovelluksen vii-

meistelylle ja raporttien laatimiselle. Toukokuun lopussa oli varattu kymmenen arkipäivän pelivara, josta viisi päivää käytettiin suorittamattomien tehtävien viimeistelyyn.

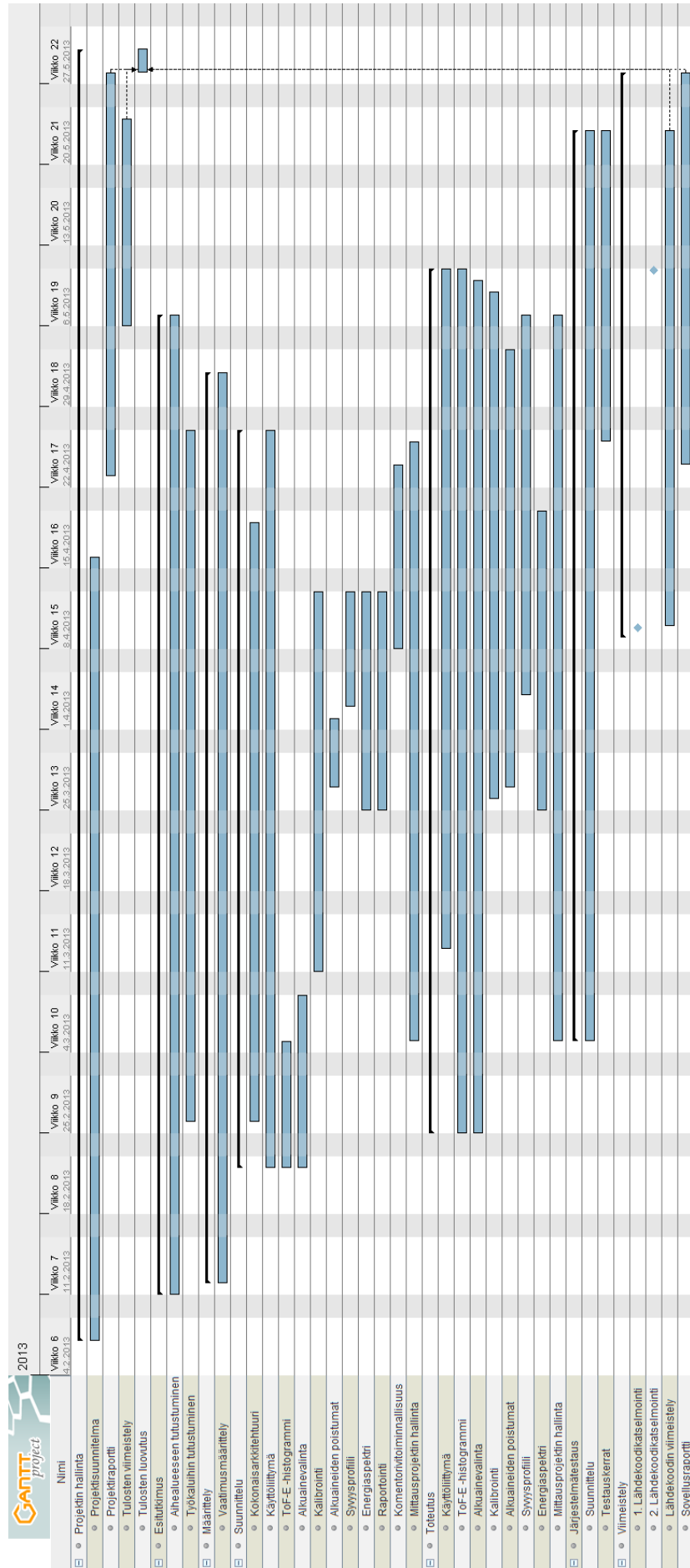
Projektin raportit allekirjoitettiin ja tulokset luovutettiin tilaajalle viikolla 22. Raporttien allekirjoitus ja tulosten luovutus venyi lähes kahdella viikolla suunnitellusta, mutta se säilyi projektille määritetyn pelivaran puitteissa.

Suunnitelmassa ja toteumassa on muutama selkeä poikkeama. Kohdealueeseen tutustuminen jatkui lähes koko projektin ajan. Käyttöliittymää kehitettiin arvioitua pidempään johtuen projektin viimeistelyvaiheessa tapahtuneista pienistä korjauksista. Lisäksi jatkokehitykseen sovittuihin ominaisuuksiin liittyviä käyttöliittymäratkaisuja jouduttiin muuttamaan luovutettavaan versioon. Toteuman ja suunnitelman poikkeamaa on myös ToF–E -histogrammissa, alkuaineiden poistumisissa sekä alkuainevalinnassa, joissa virhekorjauksia tehtiin testauksen perusteella vielä viimeistelyvaiheessa. Lentoaikakalibrointi osoittautui haastavaksi ja myös sitä toteutettiin suunniteltua pidempään. Komentorivitoiminnallisuutta tai raportointia ei toteutettu lainkaan, vaan niitä ehdittiin projektissa vain alustavasti suunnitella.

Järjestelmätestausta suunniteltiin suunniteltua pidempään. Erillisiä päivämääriin kiinnitettyjä järjestelmätestauskertoja ei järjestetty, vaan sovellusta testattiin säännöllisesti järjestelmätestaussuunnitelman pohjalta. Projektioorganisaatiolle julkaistiin kuitenkin vain lopullisen järjestelmätestauksen tulokset.



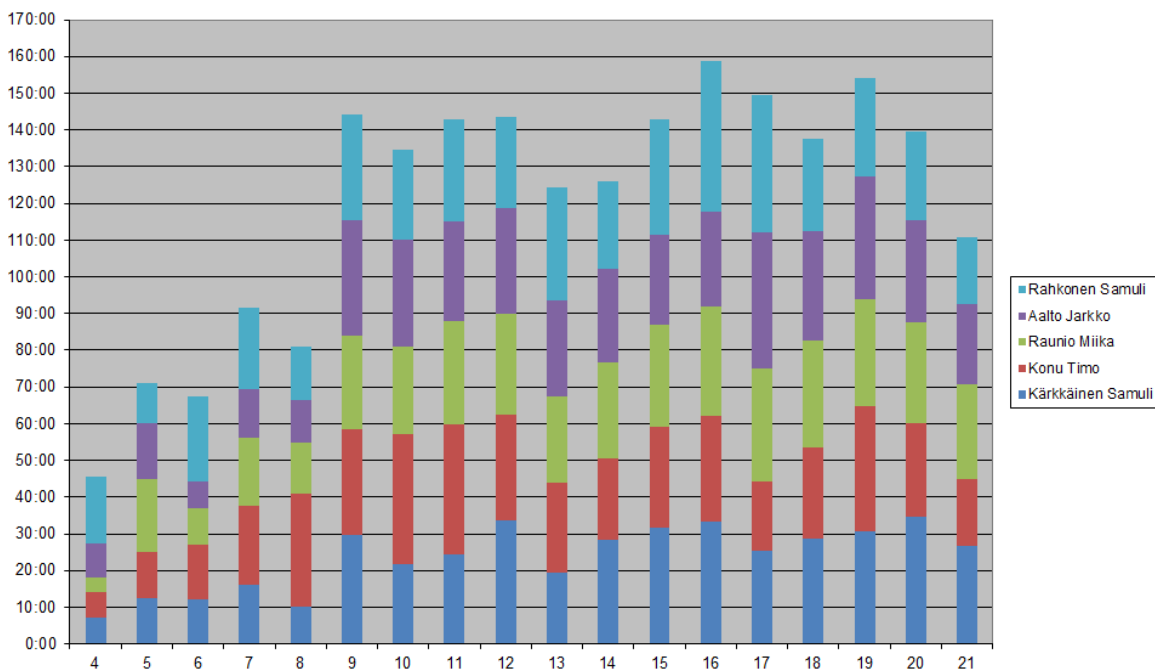
Kuva 7.1: Projektin suunniteltu aikataulu.



Kuva 7.2: Projektin toteutunut aikataulu.

7.3 Ryhmän työtunnit viikoittain

Kuvassa 7.3 on esitetty ryhmän käyttämien työtuntien jakaantuminen eri viikoille. Projektiin suunniteltiin käytettäväksi 25 tuntia viikossa kutakin ryhmän jäsentä kohden, eli **125 tuntia** viikossa koko ryhmältä. Lopulta ryhmän keskimääräiseksi toteumaksi tuli reilu 24 tuntia viikossa kultakin ryhmän jäseneltä.



Kuva 7.3: Ryhmän työtunnit viikoittain.

Kuten kuvasta käy ilmi, työtunnit eivät jakautuneet tasaisesti projektin aikana. Ensimmäinen palaveri oli viikolla 6, joten jo sitä ennen projektissa oli käytetty kaksi viikkoa oheiskursseihin. Lisäksi ohjelmointikielen valinta viivästyi, joten jopa viisi ensimmäistä projektiin käytettävää viikkoa kuluivat lähinnä puhe- ja kirjoitusviestintään, projektiluentoihin, aihealueeseen tutustumiseen sekä vaatimusmäärittelyn laatimiseen.

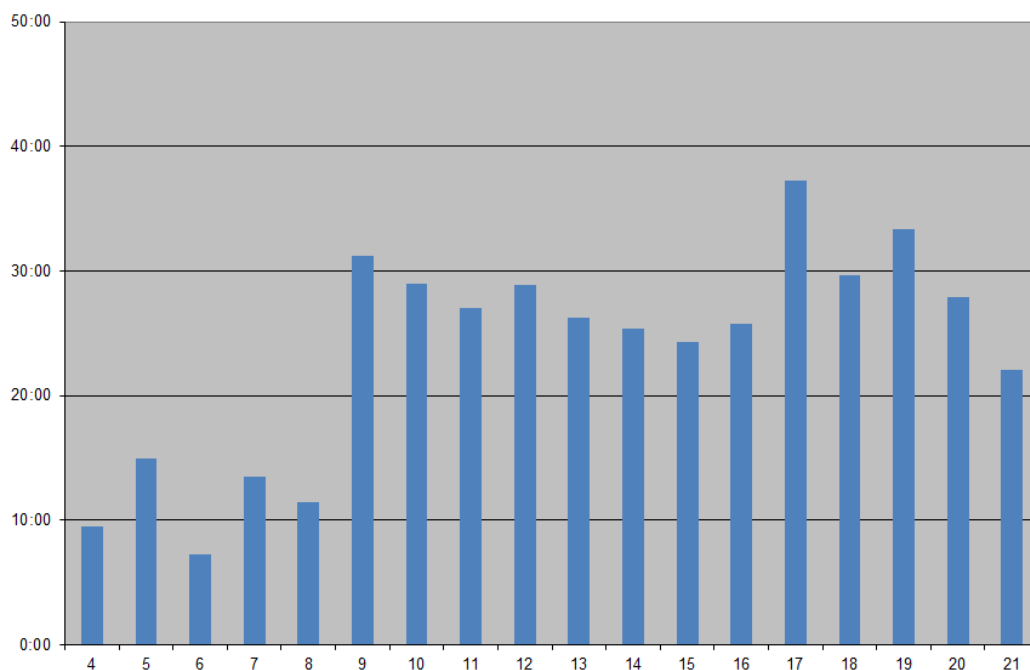
Projekti lähti toden teolla käyntiin vasta viikolla 9, jolloin aihealue oli täsmentynyt jäsenille sekä projektin läpivientiin ja sovelluksen kehittämiseen liittyvät asiat alkoivat vähitellen olla selvillä. Tämän jälkeen projektiin käytettiin keskimäärin 140 tuntia viikossa lukuunottamatta pääsiäisviikkoja 13 ja 14 sekä vappuviikkoa 18.

Suunniteltu 25 tuntia viikossa osoittautui haastavaksi, koska ryhmän jäsenet joutuivat tekemään maaliskuulta toukokuulle suunniteltua enemmän työtunteja. Projekti-päällikön olisi pitänyt huomioida hidas aloitus laatiessaan ajankäyttösuunnitelmaa.

7.4 Jarkko Aallon työtunnit viikoittain

Kuvassa 7.4 on esitetty Aallon projektiin käyttämät työtunnit viikoittain. Kuten muutkin projektin jäsenet, Aalto aloitti työnteon toden teolla vasta viikolla 9. Tämän jälkeen Aalto teki töitä tasaisesti reilun 25 tunnin viikkovauhtia ja otti loppukirin neljän viimeisen viikon aikana, jotta työtunnit tasaantuisivat muiden jäsenten kanssa.

Aallolla oli projektin aikana myös muita opintoja, kuten pro gradu -tutkielman tekeminen, mutta hän pystyi sitoutumaan projektin alussa sovitusti 25 tuntia viikossa.

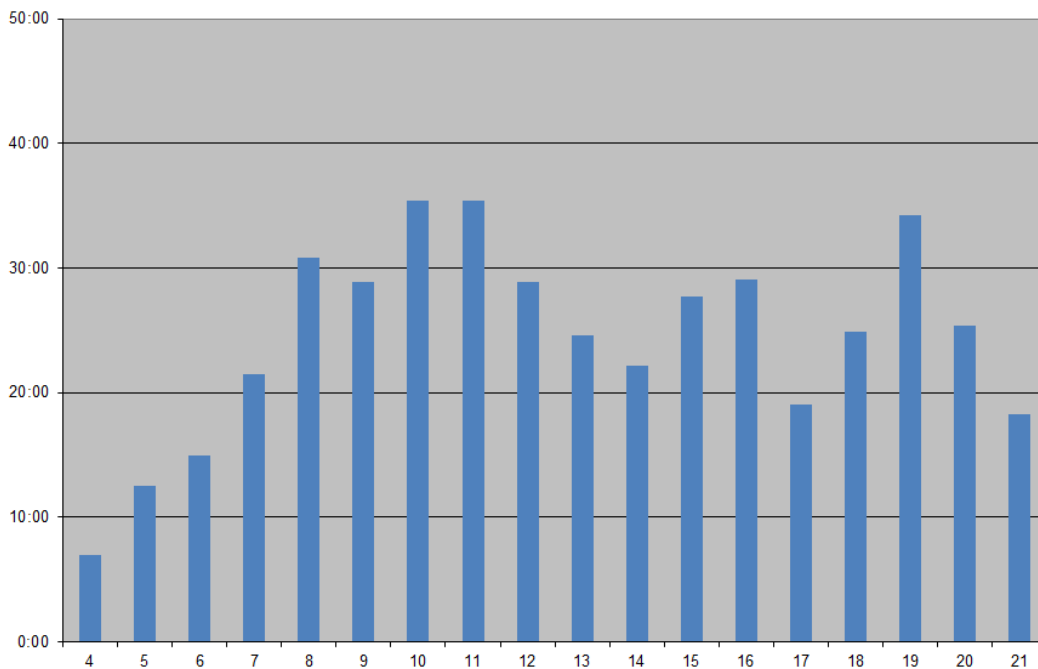


Kuva 7.4: Aallon työtunnit viikoittain.

7.5 Timo Konun työtunnit viikoittain

Konun työtunnit on esitetty kuvassa 7.5. Konun viikottaisia suuria työtuntimääriä selittää se, että hän teki hommia myös viikonloppuisin, jolloin viikolla 10 ja 11 kertyi lähes 40 tuntia työtä.

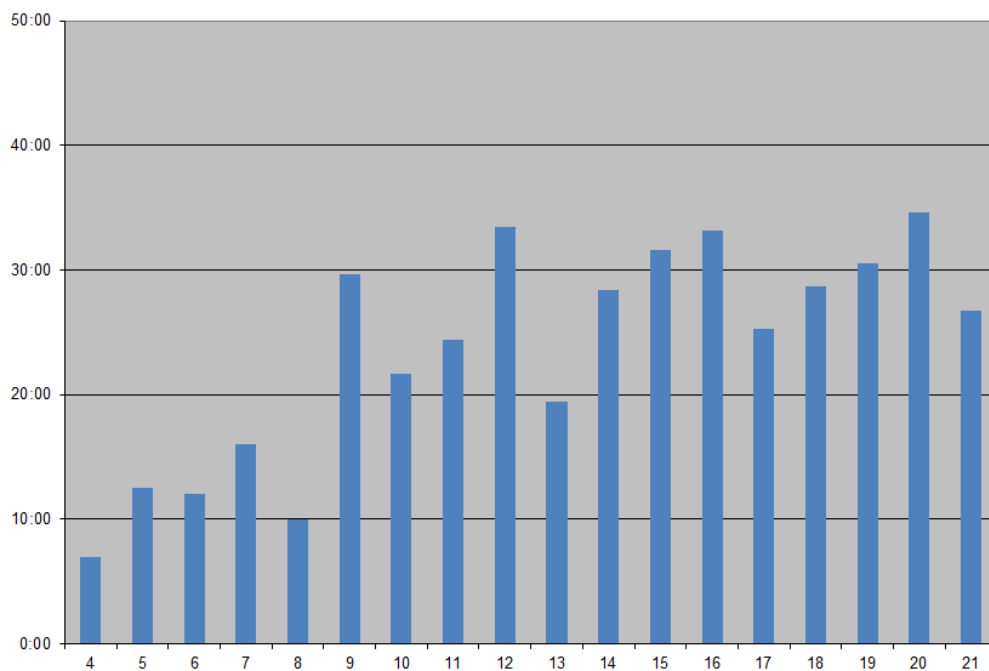
Konun työmäärä projektin alussa on nouseva, sillä hän aloitti ensimmäisenä varsinaisen ohjelmoinnin. Viikon 8 hän tutustui Pythoniin ja sen työkaluihin toteuttaen ensimmäiset osat sovelluksesta. Muun projektiryhmän tapaan Konu vietti pääsiäistä viikoilla 13 ja 14. Lisäksi hän oli sairaana viikolla 17, jolloin hänen työmääränsä on muita pienempi.



Kuva 7.5: Konun työtunnit viikoittain.

7.6 Samuli Kärkkäisen työtunnit viikoittain

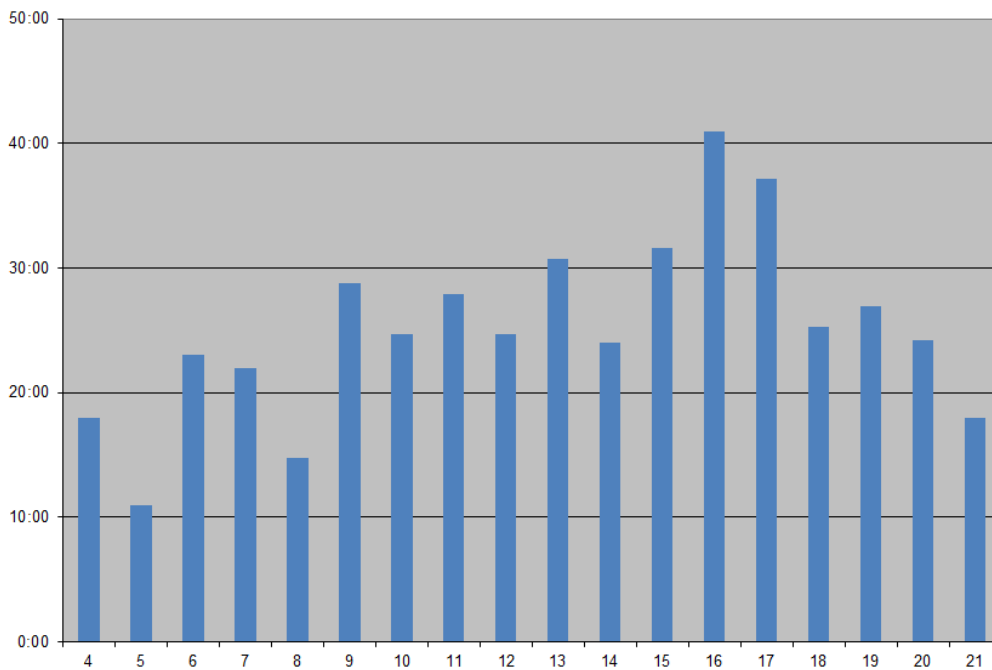
Kuvassa 7.6 on esitetty Kärkkäisen projektiin käyttämät tunnit. Kärkkäinen ei pystynyt tekemään tasaisesti tunteja, sillä hänellä oli Sovellusprojektin aikana myös muita opintoja. Lisäksi Kärkkäisellä oli ulkomaanmatka viikoilla 10–11. Myös pääsiäisenä hän oli pari päivää poissa työn äärestä, joten kyseiset viikot näkyvät selvänä pudotuksena projektin työtunteina viikolla 13.



Kuva 7.6: Kärkkäisen työtunnit viikoittain.

7.7 Samuli Rahkosen työtunnit viikoittain

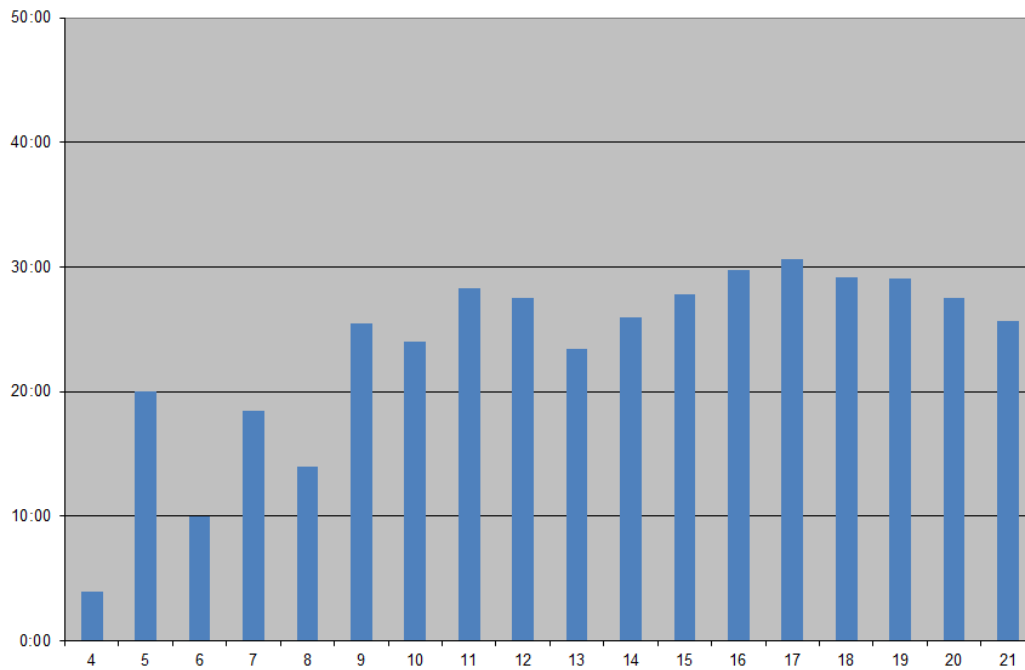
Rahkosen työtunnit jakaantuivat melko tasaisesti. Kuvasta 7.7 näkyy, että hän käytti projektin alussa muita enemmän aikaa. Hän käytti tuolloin muita enemmän aikaa oheiskursseihin ja kohdealueeseen tutustumiseen. Loppua kohti hän kiristi tahtiaan, sekä viikoilla 16 ja 17 hän työskenteli myös viikonloppuisin.



Kuva 7.7: Rahkosen työtunnit viikoittain.

7.8 Miika Raunion työtunnit viikoittain

Raunio teki töitä tasaisesti koko kurssin ajan. Kuvasta 7.8 näkyy työtuntien tasainen kertyminen koko projektin ajan, eikä työtunneissa ole suuria poikkeamia.



Kuva 7.8: Raunion työtunnit viikoittain.

8 Riskien hallinta

Luvussa on esitelty **projektisuunnitelmassa ennakoitujen riskien**, sekä niiden **toteutuminen** projektin aikana. Riskien toteutumista, **vaikutusta** ja **hallintaa** on käsitelty luvuissa 8.2–8.9.

Projektisuunnitelmassa määritellyt riskien arvioidut haittavaikutukset vastasivat toteumaa ainoastaan jäsenten toteutuksen tietotaitojen puutteiden ja tavoitteiden rajauksen osalta. Sidosryhmien toiminnan viiveet, projektin hallinnan puutteet sekä projektiorganisaation odottamattomat poissaolot oli arvioitu liian haitallisiksi, kun taas kohdealueen haasteellisuus, analyysikomponenttien puutteet sekä sovelluksen kehittäminen eri käyttöjärjestelmille toteutuivat arvioitua haitallisempina.

Keskisuurina ja suurina toteutuneet riskit vaikuttivat kaikki omalta osaltaan projektin läpivientiin. Kohdealueen haasteellisuus vaikutti projektiryhmän työskentelyyn, koska projektiryhmän jäsenet joutuivat usein kysymään tilaajalta sekä sovelluksen toimintaan että kohdealueeseen liittyviä kysymyksiä. Jäsenten puutteet toteutuksen tietotaidoissa vaikuttivat sovelluksen kehittämiseen. Ryhmän jäsenten piti opetella kehitystyökalujen käyttö, ja osa toteutusratkaisuista tehtiin "vaikeamman kautta". Analyysikomponenttien toiminnan selvittämiseen kului paljon suunniteltua enemmän työtunteja. Tavoitteiden rajaus oli osittain haasteellista, sillä projektiryhmä ei osannut arvioida työtehtävien haasteellisuutta tai niihin kuluvaan aikaan. Sovelluksen kehittäminen eri käyttöjärjestelmille hoidettiin hieman huonosti, sillä projektiryhmä testasi aktiivisesti sovelluksen toimintaa ainoastaan Windowsilla. Tämä aiheutti ongelmia projektin loppupuolella, kun sovellus ei toiminutkaan täysin samoin Linuxilla ja Mac OS:lla.

8.1 Arvioidut riskit, niiden todennäköisyydet ja haittavaikutukset

Riskien arvioidut todennäköisyydet sekä arvioidut ja toteutuneet haittavaikutukset on esitetty taulukossa 8.1. Todennäisyyttä ja haittavaikutusta arvioidaan **kolmiportaisella asteikolla** pieni, keskinkertainen ja suuri.

Luvuissa 8.2–8.9 kuvataan kunkin riskin vaikutusta projektin läpivientiin tai tuloksiin. Kunkin riskin kohdalla esitetään myös tapoja, joilla riskiä ennaltaehkäistiin ja ennakoiitiin sekä miten riskin toteutuessa haittavaikutus minimoitiin.

Riski	Arvioitu todennäköisyys	Arvioitu haittavaikutus	Toteutunut haittavaikutus
Sidosryhmien toiminnan viiveet	keskinkertainen	keskinkertainen	pieni
Kohdealueen haasteellisuus	keskinkertainen	keskinkertainen	suuri
Jäsenten puutteet toteutuksen tietotaidoissa	keskinkertainen	keskinkertainen	keskinkertainen
Analyysikomponenttien puutteet	keskinkertainen	keskinkertainen	suuri
Projektin hallinnan puutteet	keskinkertainen	keskinkertainen	pieni
Projektiorganisaatioon kuuluvien odottamattomat poissaolot	pieni	suuri	pieni
Tavoitteiden rajaus	pieni	keskinkertainen	keskinkertainen
Sovelluksen kehittäminen eri käyttöjärjestelmille	pieni	pieni	keskinkertainen

Taulukko 8.1: Riskien todennäköisyydet ja haittavaikutukset.

Projektin aikana ei ilmennyt ennakoimattomia riskejä.

8.2 Sidosryhmien toiminnan viiveet

Sidosryhmien toiminnan viiveillä tarkoitettiin tämän projektin tapauksessa pääosin **ATK-tuen toiminnan viiveitä**. ATK-tukea tarvittiin kurssin aikana verkkolevyjen perustamiseen sekä ohjelmien asennukseen niin Windowsille kuin Mac OS:llekin.

ATK-tuki hoiti hommansa vähintään kohtuullisesti, kunhan heidän ohjeistuksensa olivat riittävän tarkkoja. Mac OS:lle asennettavissa työkaluissa ja kirjastoissa oli pieni viive. Tämä johtui osittain siitä, että projektiryhmä ei osannut tarkentaa tilannetta, eikä hoputtaa IT-palveluiden toimintaa. Toimituksen hitaus viivästytti järjestelmätestausta hieman, mutta riskin toteutuminen ei vaikuttanut merkittävästi projektin läpivientiin.

Hallitakseen riskiä projektiryhmä lähetti tukipyynnöt ATK-tuelle riittävän ajoissa ja tarvittaessa myös hoputti ATK-tukea. Suunnitelman mukaan tukipyynnöt tuli ohjata myös vastaavalle ohjaajalle. Projektiryhmä ei kuitenkaan aina tehnyt tätä, mutta siitä ei onneksi ollut haittaa projektin läpiviennille tai riskin toteutumiselle.

8.3 Kohdealueen haasteellisuus

Kehitettävän sovelluksen **kohdealue oli haastava ja monimutkainen**, eikä projektiryhmän jäsenillä ollut aiempaa kokemusta siitä. Tämän takia kehitettävän sovelluksen ominaisuuksia jouduttiin usein varmistamaan palavereissa ja sähköpostilla. Tilaaja joutui myös selittämään samaa asiaa useamman kerran, koska projektiryhmä ei välttämättä ymmärtänyt haasteellisia asioita ensimmäisellä kerralla. Tämä johti myös siihen, että palaverit venyivät toisinaan melko pitkiksi.

Kohdealueen haasteellisuus vaikutti merkittävästi projektiryhmän toimintaan. Arvioitu keskinkertainen vaikutus oli ehkä hieman alakanttiin, sillä toisinaan projektin eteneminen viivästy juuri sen takia, että projektiryhmä ei ollut aivan varma siitä, miten sovelluksen tulisi toimia.

8.4 Jäsenten puutteet toteutuksen tietotaidoissa

Projektiryhmän jäsenet olivat opiskelijoita, joiden tietotaidoissa oli puutteita muun muassa työkalujen, toteutusratkaisujen ja tekniikoiden osalta. Tämän takia projektiryhmä ei kyennyt ennakoimaan kaikkia eteen tulevia tilanteita ja haasteita, eivätkä he onnistuneet ratkaisemaan niitä senhetkisillä tietotaidoilla. Tämä johti projektin hidastumiseen etenkin C-kielisten analyysikomponenttien toiminnan haasteellisuu-den takia.

Sekä kehitysympäristö että ohjelmointikieli olivat projektiryhmälle vieraita. Niiden toimintaan jouduttiin kiinnittämään huomiota, joka taas vei aikaa itse sovelluksen kehittämisestä. Lisäksi työkalujen ja ohjelmointikielen hyviä ominaisuuksia ei välttämättä heti osattu hyödyntää, vaan asiat tehtiin niin sanotusti "vaikeamman kautta". Virheellisesti tehdyt asiat selvisivät yleensä vasta katselmoinnissa tai teknisen ohjaajan huomautuksen jälkeen. Niiden korjaaminen vei siten enemmän aikaa kuin se, että virheet olisi huomattu riittävän ajoissa. Osa ryhmän jäsenistä ehti projektin

aikana tutustua työkaluihin paremmin kuin muut, joten kokemattomamat saivat apua myös heiltä.

Tekninen ohjaaja ei omien kiireidensä vuoksi ehtinyt omatoimisesti käydä kysymässä projektiryhmän työn etenemistä, joten kommunikointi tapahtui yleensä projektiryhmältä tekniseen ohjaajaan päin. Projektiryhmän olisi ehkä pitänyt olla aktiivisemmin yhteydessä ohjaajiin ja tilaajiin liittyen sovelluksen toteutukseen, jotta riskiä toteutuksen tietotaidon puutteissa olisi saatu entisestään ehkäistyä.

8.5 Analyysikomponenttien puutteet

Projektiryhmä sai projektin aikana käyttöönsä Kai Arstilan tekemiä C-kielisiä analyysikomponentteja. Arstila ja hänen oppilaansa oli tehnyt komponentit vuosia sitten Linuxilla. Tilaajan toimittamien komponenttien kanssa olikin ongelmia heti alusta alkaen, sillä niitä ei kivuttomasti saatu toimimaan Windowsilla, eikä edes Linuxilla. Koska projektiryhmällä ei ollut kokemusta C-kielestä, ongelmien ratkaisuun meni melko kauan aikaa. Lisäksi analyysikomponenttien toiminnan kanssa oli epäselvyyksiä, joten projektiryhmä joutuikin selvittämään niitä tilaajalta monessa palaverissa.

Riski siis toteutui projektin aikana, ja sen haittavaikutus osoittautui yllättävän suureksi. Jos analyysikomponentit olisi saatu heti toimimaan, olisi projektiryhmälle jäänyt huomattavasti enemmän työtunteja itse sovelluksen kehittämiseen. Nyt analyysikomponenttien muokkaamiseen meni todella paljon aikaa, sillä ne jouduttiin kääntämään myös kaikille eri käyttöjärjestelmille. Ongelmia käyttöjärjestelmien kanssa on kuvattu tarkemmin luvussa 8.9.

Olisi ehkä ollut helpompi toteuttaa algoritmit suoraan itse tekemillä Python-moduuleilla. Toisaalta näihin moduuleihin olisi tarvinnut tilaajan edustajien opastusta. Todennäköisesti tällä olisi kuitenkin säästetty myös tilaajien aikaa, sillä he joutuivat tutkimaan analyysikomponentteja ja selittämään niiden toimintaa useaan kertaan.

8.6 Projektin hallinnan puutteet

Projektiryhmän jäsenillä ei ollut aiempaa kokemusta projektin suunnittelusta, läpiviennistä tai hallinnasta. Myös projektipäällikkö vastasi projektin läpiviennistä ensimmäistä kertaa. Vaikka riski ennakoitiin toteutuvaksi ja haittavaikutukseltaan keskinkertaiseksi, suoriuduttiin projektin läpiviennissä ilman suuria ongelmia. Asioita ei tietenkään tehty täysin oikein, ja hallintaa voi parantaa tulevissa projekteissa.

Suuri projektiryhmä hankaloitti työnjakoa, eivätkä kaikki jäsenet päässeet projektin aikana ohjelmoimaan. Kaikille kuitenkin riitti työtä, ja suuria epäselvyyksiä työnjaossa ei esiintynyt. Projektipäällikkö pyrki pitämään itsensä tietoisena siitä, mitä kukin ryhmän jäsen on tehnyt, tekee tällä hetkellä ja aikoo tehdä seuraavaksi. Myös projektin työtunnit jakaantuivat melko tasaisesti ryhmän jäsenten kesken. Ryhmän jäsenet kirjasivat työtunnit hyvin, täsmällisesti ja ajallaan. Projektipäällikön olikin helppo seurata sovelluksen ja projektin edistymistä.

Kaikki ryhmän jäsenet olivat yleensä tietoisia siitä, missä vaiheessa projekti eteni. Tämä onnistui sen takia, että projektiryhmä työskenteli tiiviisti yhdessä ja asioista keskusteltiin työhuoneessa avoimesti. Kaikki uskalsivat myös kysyä toisilta projektin läpivientiin liittyviä asioita. Tiiviin yhteistyön takia myös ryhmähenki oli hyvä koko projektin ajan. Ryhmän jäseniä kunnioitettiin, joten myös tiedonkulku onnistui hyvin. Lisäksi projektiryhmä auttoi projektipäällikköä. Ryhmän jäsenet muistuttivat projektipäällikköä, jos jokin asia uhkasi jäädä huomiotta. Myös vastaava ohjaaja auttoi projektipäällikköä parissa asiassa projektin läpiviennissä.

8.7 Projektiorganisaatioon kuuluvien odottamattomat poissaolot

Projektin aikana ryhmän jäsenille ei ilmaantunut merkittäviä odottamattomia poissaoloja. Kaikilla ryhmän jäsenillä oli enemmän tai vähemmän velvoitteita muillakin kurseilla, mutta mitkään niistä eivät varsinaisesti olleet odottamattomia. Joihinkin toisten kurssien työtehtäviin meni odotettua enemmän aikaa, mutta se ei vaikuttanut projektin läpivientiin.

Riskiä hallittiin projektin aikana siten, että ryhmän jäsenet keskustelivat projektin aikana työtehtävistään. He myös pitivät huolen siitä, että myös muut tietävät, mitä kukin jäsen on projektin aikana tehnyt. Täten yllättävän poissaolon toteutuessa ryhmän jäsenet olisivat tienneet poissaolijan työtehtävät, ja niitä olisi voinut jakaa

muiden jäsenten kesken.

8.8 Tavoitteiden rajaus

Projektiryhmällä ei ollut kokemusta sovelluksen kehittämisestä ja projektin läpiviennistä. Tämä vaikutti hieman siihen, että projektiryhmä ei riittävän rohkeasti tuonut omia mielipiteitään esille palaverissa, vaan kuunteli tilaajan edustajia kuuliaisesti. Toisaalta se saattoi olla myös hyvä asia projektin kannalta, koska tilaajalla oli selkeä kuva toivomastaan sovelluksesta. Kuitenkin tilaajan edustajien mielipiteet vaatimusten priorisoinnista sekä sovelluksen ulkoasusta poikkesivat toisistaan, joten projektiryhmä joutui toisinaan hieman turhaankin muuttamaan senhetkisiä suunnitelmiaan.

Riskiä pyrittiin ehkäisemään hyvällä vaatimusmäärittelyllä. Tämä onnistuikin, sillä projektin alussa laadittiin tilaajan kanssa onnistunut vaatimusmäärittely prioriteetteineen. Projektiryhmä pyrki toteuttamaan kaikki tärkeimmät ominaisuudet ensin ja siirtyi vasta niiden jälkeen kehittämään vähemmän tärkeitä ominaisuuksia.

Tilaaja oli myös joustava tavoitteiden suhteen. Projektin lopussa paljastui, että muutamaa tärkeintä ominaisuutta ei ehditä toteuttaa. Tämä johtui osittain myös siitä, että analyysikomponenttien vaatima työmäärä oli paljon suunniteltua suurempi. Tilaaja suostuikin niiden siirtämiseen jatkokehitykseen. Jos jatkokehitykseen sopiminen ei olisi ollut mahdollista, olisi projekti mahdollisesti viivästynyt.

8.9 Sovelluksen kehittäminen eri käyttöjärjestelmille

Tilaaja toivoi, että sovellus toimisi Windows-, Linux- ja Mac-ympäristössä. Sovellusta kuitenkin kehitettiin Windowsilla, sekä analyysikomponenttien toimintaan tutustuttiin Windowsilla ja Linuxilla. Mac-ympäristössä toimintaa voitiin testata vasta toukokuun alussa, kun projektiryhmä sai käyttöönsä oman Macin.

Analyysikomponenttien toiminta ei ollut samanlaista erilaisilla käyttöjärjestelmillä. C-kielisten ohjelmien kääntäminen ja ajaminen ei ollut niin yksinkertaista kuin projektiryhmä kuvitteli. Varsinkin kääntämiseen kului todella paljon aikaa, joka oli pois muulta tehokkaalta työltä. Ylimääräistä työtä teetti myös Linux-versioiden (kuten Fedora, Ubuntu ja Debian) erilainen toiminta.

Risikin hallinta oli projektin aikana puutteellista. Projektiryhmän olisi pitänyt testata toimintaa koko projektin ajan kaikilla käyttöjärjestelmillä. Testaus kuitenkin jäi viimeisille viikoille, jolloin paljastuneet puutteet aiheuttivat kiirettä projektin lopussa. Tähän vaikutti myös se, että sovelluksen suunniteltua kehitystä Linux-ympäristössä ei projektin aikana tehty. Lisäksi projektiryhmän olisi pitänyt olla tarkempi toimivan Linux-version valinnassa ja heti alussa pyytää tarkennusta siitä, millä Linux-versiolla sovelluksen on toimittava.

9 Jäsenten kokemuksia ja oppimaa

Projekti oli kaikille ryhmän jäsenille ensimmäinen oikea ohjelmistokehitysprojekti, joka sisälsi projektin kaikki kehitysvaiheet aina vaatimusmäärittelystä projektin tulosten luovutukseen. Jäsenet saivat kattavan kuvan projektin läpiviennistä, sekä ymmärsivät, että sovelluksen kehittäminen on kaikkea muutakin kuin pelkkää ohjelmointia. Kaikki ryhmän jäsenet oppivat projektin aikana ryhmätyö- ja palaveritaitoja. Luvuissa 9.1–9.5 projektiryhmän jäsenet kuvaavat kokemuksiaan ja oppimiaan asioita.

9.1 Jarkko Aallon kokemuksia ja oppimaa

Sovellusprojekti osoittautui suurta työpanosta vaativaksi, kuten kurssin aiemmin käyneet opiskelijat siitä jo varoittelivat. Projektin loppuvaiheessa kuitenkin voi todeta, että projekti ja sen oheiskurssit tarjosivat paljon hyödyllisiä tietoja ja taitoja tulevaisuutta varten.

Projektin toimeksiantona oli uuden graafisen käyttöliittymän toteuttaminen vanhan käyttöliittymän tilalle. Suunnittelua ja toteutustyötä helpotti se, että tilaajilla oli selvä käsitys siitä, mitä ominaisuuksia uuden käyttöliittymän tulisi sisältää. Toteutuksen haasteeksi osoittautuikin tilaajan toimittamien C-kielellä toteutettujen analyysikomponenttien toiminnallisuuden varmistaminen Windows-, Linux- ja Mac OS X -käyttöjärjestelmissä.

Projektin alussa en tuntenut muita jäseniä entuudestaan, mutta projektin edetessä heihin tutustui nopeasti. Ryhmän työskentelytapa ja ryhmähenki olivat koko projektin ajan erittäin hyviä, ja kaikkien kova työpanos takasi projektin onnistuneen läpiviennin. Miinuksena ehdottomasi pidän sitä, että en ehtinyt ollenkaan osallistua graafisen käyttöliittymän toteutukseen Pythonilla, vaan työpanokseni kului C-analyysikomponenttien toiminnallisuuden varmistamiseen.

Sovellusprojekti ja sen oheiskurssit opettivat minulle projektin läpiviennissä hyödyllisiä puhe-, esiintymis-, ja ryhmätyötaitoja sekä kokouskäyttämistä. Hyödyllisimpänä pidän ehdottomasti kokouksissa tarvittavien taitojen oppimista.

9.2 Timo Konun kokemuksia ja oppimaa

Sovellusprojekti-kurssi taitaa olla niitä ensimmäisiä kurssejani, jossa oikeasti tarvitaan kunnollista ryhmätyöskentelyä. Aikaisemmillä kurseillakin ryhmätöitä on tarvinnut tehdä, mutta ne tuntuivat jokseenkin väkisin väännetyiltä tyyliin "joo tehdään nyt näin". Tästä johtuen en varsinaista ryhmätyöskentelyä kurssien parissa ole harrastanut. Sovellusprojekti oli ensimmäinen kurssi, missä ryhmätyötaitoja pääsin kunnolla kehittämään. Projektiryhmän jäsenistä Samuli Rahkonen ja Miika Raunio olivat jo entuudestaan tuttuja, sillä aloitimme samaan aikaan, joten olimme useasti samoilla kursseilla.

Sovellusprojekti-kurssille päädyin opintojen takia, sillä kiinnostusta oli päästä tekemään jotain mielenkiintoisempaa kuin lyhytkestoisempia ohjelmointikurssien harjoitustöitä. Lisäksi maisterivaiheen opinnoissa vaadittiin projektiohjelmaa, joita ei minulla aikaisemmin ollut. Muita kurssia sovellusprojektin ohella en ollut syksyllä suunnitellut, mutta muutamien mutkien kautta niitä kuitenkin loppujen lopuksi tuli valittua. Työmäärältään ne eivät kuitenkaan olleet haastavia, joten ne eivät vaikuttaneet varsinaiseen projektityöskentelyyn lainkaan. Tenttipäiviä varten jouduin varaamaan jonkin verran aikaa kertaukseen.

Projektin aihe vaikutti aluksi haastavalta, mutta pikku hiljaa palaverien yhteydessä pääsi kärryille, mitä tilaajan edustajat haluavat toteutettavalta sovellukselta. Toisin kuin toisella kevään projektiryhmällä, Potkun tilaajan edustajilla oli käytössä jo entuudestaan sovellus, josta oli tarkoitus kehittää uusi ja jatkokehittävämpi sovellus. Mielestäni olemme onnistuneet hyvin näin pienessä aikarajassa, koska koko sovellusprojektin alkutaival oli melkoista sekaannusta ja aihealueeseen tutustumista. Se ei kuitenkaan haitannut projektin läpivientiä, vaikka suurimmat ongelmat ilmenivät tilaajan edustajan C-kielisten analyysikomponenttien yhdistämisessä toimivasti kehitettävään ohjelmaan.

Kokonaisuudessaan projektista jäi hyvin paljon käteen. Oheiskurssien avulla kokoustaitoja tuli kehitettyä projektin palavereissa. Opin lisäksi ryhmätyöskentelyä useamman henkilön kanssa sekä työtehtävien jakoa niin, että toteutettava projekti saatiin vietyä läpi ilman suurempia ongelmia. Python-ohjelmointikielellä en aikaisemmin ole ohjelmoinut, mutta nyt voin sanoa sitäkin osaavani. Havaintona koko projektista voin todeta, että hyvä kommunikaatio ryhmän sisällä sekä myös tilaajan edustajien kanssa on ehdotonta, jotta projekti etenee sulavasti ja aikataulussa.

9.3 Samuli Kärkkäisen kokemuksia ja oppimaa

Minulla oli kurssin alussa hieman kaksijakoiset odotukset kurssista. Toisaalta olin innoissani saamastani mahdollisuudesta osallistua elämäni ensimmäiseen kokonaiseen sovellusprojektiin. Toisaalta omat kykyni ohjelmoida ja hoitaa projektin läpivientiä vaaditulla tasolla epäilyttivät ja ehkä hieman hirvittivät. Asiaa ei lainkaan helpottanut projektin aiheen selviäminen. Kohdealue vaikutti todella haastavalta, ja päätin vielä ottaa projektipäällikön tehtävän vastuulleni.

Projekti kuitenkin osoittautui erittäin opettavaiseksi, eikä lainkaan hirveäksi kokemukseksi. Johtamiskokemusta minulla oli vain armeijasta, eikä armeijan oppeja voinut ainakaan suoraan siirtää sovellusprojektien maailmaan. Projektin aikana kuitenkin koin oppineeni projektipäällikön tehtävät ainakin kurssin vaatimalla tasolla. Sain selkää ohjausta ja tukea myös muilta ryhmän jäseniltä. Nämä asiat helpottivat huomattavasti työtäni ja tehostivat oppimistani merkittävästi.

Projektipäällikön tehtävät osoittautuivat melko haastaviksi ja yllättävän paljon aikaa vieviksi. Erilaisten dokumenttien kirjoittaminen tuntui välillä turhauttavalta, mutta ymmärrän kyllä, että ilman suunnittelua ja raportointia projektin läpiviennistä ei tulisi mitään. Dokumentit onnistuivat mielestäni varsin hyvin, sillä niistä tuli selkeitä, ymmärrettäviä ja kieliasultaan riittävän oikeita. Sain paljon palautetta dokumenteista myös vastaavalta ohjaajalta, mikä vaikutti dokumenttien laatuun. Toisinaan tosin tuntui, että palautetta tuli jo liian paljon.

Projektin aikana pyrin pitämään itseäni koko ajan tietoisena siitä, mitä ryhmän jäsenet tekevät ja mitä he tulevat tekemään. Mielestäni onnistuin tässä erinomaisesti. En tiedä, kuinka ahdistavaksi ryhmän jäsenet kokivat päivittäiset kysymykseni, mutta niiden avulla pystyin ainakin omasta mielestä hallitsemaan projektin läpivientiä paremmin kuin ilman kysymyksiä. Jäsenet myös itse ilmoittivat kiitettävästi tekemistään asioista, eikä minun tarvinnut tarkastaa niitä ajankäyttöraportista.

Projektipäällikön tehtävät kuitenkin veivät työaikaani varsinaiselta ohjelmoinnilta. Python ohjelmointikielenä ei ollut entuudestaan tuttu, joten olisin halunnut tutustua siihen tarkemminkin. Projektin aikana toteutin vain sovelluksen lokin toiminnallisuuden ja tein pieniä muutoksia käyttöliittymään. Tekemäni ohjelmoinnin avulla sain vain pintaraapaisun siitä opista, jonka Konu, Rahkonen ja Raunio saivat. Tämä ei kuitenkaan varsinaisesti haittaa, sillä kuten mainittu, sain tärkeää kokemusta projektin läpiviennistä.

Projektiryhmä oli erinomainen. Kaikki ymmärsivät vastuunsa ja tekivät töitä todel-

la tunnollisesti. Välillä tuntui, että muilla jäsenillä oli paljon aikaa tehdä projektia, enkä itse pysy lainkaan mukana työtunneissa. Tämä johtui osittain myös siitä, että suoritin keväällä enemmän opintoja kuin Konu, Rahkonen ja Raunio. Työaikojen suhteen pysyin kuitenkin kelkassa kohtuullisesti, eivätkä työaikamme lopulta poikenneet mahdollisesti toisistaan.

Kokonaisuutena sovellusprojekti oli siis erinomainen kokemus. Opin kurssilla kenties enemmän kuin millään suorittamallani kurssilla, ja koen saaneeni todella tärkeää kokemusta projektimuotoisesta työtavasta.

9.4 Samuli Rahkosen kokemuksia ja oppimaa

Osallistuin sovellusprojekti kurssille sillä ennako-odotuksella, että kurssi vie paljon aikaa. Kävinkin samaan aikaan vain pari muuta kurssia, jotta voisin keskittyä kurssin suoritukseen mahdollisimman hyvin. Yllätyin, miten paljon aikaa kurssilla kului kaikkeen muuhun kuin itse sovelluksen suunnitteluun ja toteutukseen. Tosin palaverit ja muistioid olivat kuitenkin tärkeitä aihealueen ja vaatimusten hahmottamisen kannalta.

Itselläni kesti pitkään, ennen kuin selvitin itselleni uuden ohjelmointikielen ja PyQt-kirjaston koukerot. Alussa paneuduin paljon käyttöliittymän hahmotteluun, joka näin jälkikäteen ajateltuna ei ollut kovin hyödyllistä. Tärkeämpää olisi ollut aloittaa testiohjelmien koodaaminen nopeasti, jotta siihen ei olisi myöhemmin kulunut niin paljoa aikaa. Qt-kirjaston käytön opiskelu kangerteli myös siksi, ettei tekninen ohjaaja tuntenut kyseistä kirjastoa itsekään kovin hyvin. Opettelussa jouduinkin useimmiten tukeutumaan itseopiskeluun.

Projekti eteni koko kevään ajan kiihtyvällä tahdilla. Projektin alussa suurimmat ongelmat olivat aihealueen ja vaatimusten ymmärtäminen. Lisäksi oli vaikea hahmottaa, millä tavoilla tilaaja haluaa sovellusta käyttää. Vaikka sovellukselle oli ”mallina” olemassa oleva analysointisovellus Finlandia, se ei ohjannut mitenkään, mitä käyttäjä haluaa sovelluksella analysoida ja missä järjestyksessä. Näistä asioista piti projektin aikana kysellä moneen otteeseen. Projektin suurimmat tekniset ongelmat liittyivät useimmiten C-kielisten ohjelmien kutsumiseen käyttöliittymästä.

Kokonaisuudessaan projekti eteni mukavasti ja kokemus oli positiivinen. Kurssi opetti kevään aikana enemmän kuin mikään ohjelmointikurssi tähän asti ja antoi jonkinlaisen kuvan projektityöstä ja ohjelmistokehityksestä yleensä. Iso kiitos kuu-

luu myös kurssin vetäjälle Jukka-Pekka Santaselle, joka antoi paljon hyvää palautetta ja ohjasi jäseniä projektin joka vaiheessa.

9.5 Miika Raunion kokemuksia ja oppimaa

Lähdin kurssille odottaen melko työlästä kurssia, josta saisin kokemusta projektityöskentelystä ja ohjelmointityöstä. Suoritin sovellusprojektin aikana 17 opintopisteen edestä muita opintoja (poislukien projektin oheiskurssit), joista haastavin lienee ollut tietotekniikan syventävä kurssi Future Internet. Näiden lisäksi kirjoitin loppuun kandidaatintutkielmaani, jonka olin aloittanut syksyllä. Huhtikuun loppuun mennessä muut kurssit ja kandidaatintutkielma maturiteetteineen olivat suoritettuja ja saatoin omistaa kaiken työskentelyni projektille.

Projektin jäsenistä minulle entuudestaan olivat tuttuja Timo Konu ja Samuli Rahkonen. Jarkko Aaltoa ja Samuli Kärkkäistä olen aika-ajoin yliopistolla nähnyt, mutten heitä tätä paremmin tuntenut. Heihin tutustuminen kuitenkin onnistui kivuttomasti, ja ryhmä muodostui ilman suuria ongelmia. Ryhmän kanssa työskentely oli mukavaa.

Opin projektin aikana rutkasti projektimuotoisesta työskentelystä. Kokemusta projekteista ei minulla aikaisemmin ollut muuta kuin luennoilta opitut asiat. Opin Python-ohjelmoinnin hienouksia, sekä käyttöliittymien toteutusta Pythonilla. Hie-man yllättäen opin myös C-ohjelmointia, jouduttuani kamppailemaan C-koodien kanssa rutkasti sovelluksen toteutuksen alkuvaiheissa.

Sanoisin, että kokonaisuudessaan sovellusprojekti oli hieno kurssi. Kaikki ei onnistunut aina kivutta, ja etenkin C-kielisten ohjelmien kanssa kamppailu aiheutti harmaita hiuksia. Onneksi niiden ongelmista selvittiin. Sain arvokasta kokemusta projektityöskentelystä.

10 Yhteenveto

Potku-projekti toteutti Sovellusprojekti-kurssilla keväällä 2013 Jyväskylän yliopiston fysiikan laitoksen kiihdytinpohjaisen materiaalfysiikan tutkimusryhmälle käyttöliittymän rekyylispektrometrin mittausedatan analyysisovellukseen. Rajoitettuun tuotantokäyttöön soveltuvan version avulla rekyylispektrometrillä kerättyä mittausedataa voidaan analysoida lentoaika-energia -histogrammien, energiaspektrien, alkuaineiden poistumien ja syvyysprofiilien avulla.

Kehitetyllä sovelluksella korvattiin Jyväskylän yliopistossa käytössä ollut vanha sovellus. Toteutettua sovellusta on tarkoitus markkinoida ja jakaa ilmaiseksi kiihdytinlaboratorioille ympäri maailmaa. Tilaaja esittelee sovellusta ensimmäistä kertaa kansainvälisesti konferenssiabstraktissa [13]. Tilaaja jatkokehittää projektissa kehitettyä sovellusta ja onkin palkannut projektiryhmässä työskennelleen Timo Konun jatkokehittämään sitä kesän ajaksi.

Projektiin käytettyjen työtuntien toteuma ylittyi hieman. Jokainen jäsen teki projektissa reilut 430–460 tuntia töitä, joten projektiryhmän kokonaispanokseksi muodostui noin 2200 tuntia. Projekti läpiviettiin aikataulullisesti sille suunnitellun pelivaran puitteissa, eikä projekti täten myöhästynyt lainkaan.

Projektin suurimpina riskeinä toteutuivat kohdealueen haasteellisuus sekä analyysikomponenttien integroinnin haasteellisuus. Kyseisiä riskejä saatiin kuitenkin hallittua projektin aikana, ja ne eivät estäneet projektin läpivientä. Näiden riskien toteutumisesta johtuneiden ongelmien ratkaisemiseen jouduttiin kuitenkin käyttämään suunniteltua enemmän työtunteja, ja pari sovelluksen ominaisuuksista jouduttiin sopimaan jatkokehitykseen.

Sovelluksen kehittämisessä projektiryhmä onnistui suunnitellusti. Vaikka sovellus ei vielä sisällä kaikkia vaadittuja ominaisuuksia, tilaajan edustajat voivat käyttää sitä analyyseissaan perustoiminnallisuuksien osalta. Myös projektin läpivienti onnistui erinomaisesti huomioiden, että se oli jäsenille ensimmäinen laajempi ohjelmistokehitysprojekti. Jäsenet saivatkin mainiota kokemusta projekteista oppien muun muassa vaatimusmäärittelyn ja hyvän suunnittelun tärkeyden onnistuneessa projektissa. Jäsenet oppivat myös tarvittavia ryhmätöitäitoja sekä saivat soveltaa muilla kursseilla oppimaansa teoriaa käytäntöön. Sekä projekti että Sovellusprojekti-kurssi olivat kaikin puolin onnistuneita.

11 Lähteet

- [1] Petri Heinonen, Ajankäytönseurantasovellus, saatavilla Excel-muodossa <URL: <http://appro.mit.jyu.fi/tools/ajankaytto/ajankaytonseuranta.xls>>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, viitattu 13.3.2013.
- [2] Pekka Iso-Ahola, Jussi Perttola ja Tommi Tuovinen, "Kuvatus-sovellusprojekti, Projektiraportti", Jyväskylän yliopisto, tietotekniikan laitos, 24.4.2012.
- [3] Python Software Foundation, "Python Tutorial", saatavilla HTML-muodossa <URL: <http://docs.python.org/3/tutorial/index.html>>, viitattu 18.3.2013.
- [4] Jaakko Julin, "Development of a High Energy Resolution Gas Ionization Detector for a Recoil Spectrometer", Jyväskylän yliopisto, fysiikan laitos, 28.6.2011.
- [5] Tapio Keränen, Toni Salminen, Jari Salokangas ja Lauri Satokangas, "Paattisovellusprojekti, Projektiraportti", Jyväskylän yliopisto, tietotekniikan laitos, 7.6.2012.
- [6] Amit Patel, Antoine Picard, Eugene Jhong, Jeremy Hylton, Matt Smart and Mike Shields, "Google Python Style Guide", <URL: <http://google-styleguide.googlecode.com/svn/trunk/pyguide.html>>, viitattu 26.3.2013.
- [7] Jukka-Pekka Santanen, "Tietotekniikan Sovellusprojektien ohje", saatavilla HTML-muodossa <URL: <http://www.mit.jyu.fi/opetus/sovellusprojektit/projohje.pdf>>, Jyväskylän yliopisto, tietotekniikan laitos, 29.1.2013.
- [8] Timo Sajavaara, "Sovellusprojekti: Rekyylispektrometrin mittausdatan analyysiohjelma", Jyväskylän yliopisto, fysiikan laitos, 7.2.2013.
- [9] Jarkko Aalto, Timo Konu, Samuli Kärkkäinen, Samuli Rahkonen and Miika Rautio, "Potku Project, Requirements Specification 1.0.0", The University of Jyväskylä, Department of Mathematical Information Technology, 22.5.2013.
- [10] Jarkko Aalto, Timo Konu, Samuli Kärkkäinen, Samuli Rahkonen ja Miika Rautio, "Potku-sovellusprojekti, projektisuunnitelma 1.0.0", Jyväskylän yliopisto, tietotekniikan laitos, 18.4.2013.

- [11] Jarkko Aalto, Timo Konu, Samuli Kärkkäinen, Samuli Rahkonen and Miika Raunio, "Potku Project, Application Report 1.0.0", The University of Jyväskylä, Department of Mathematical Information Technology, 27.5.2013.
- [12] Jarkko Aalto, Timo Konu, Samuli Kärkkäinen, Samuli Rahkonen and Miika Raunio, "Potku Project, System Testing Plan 1.0.0", The University of Jyväskylä, Department of Mathematical Information Technology, 22.5.2013.
- [13] Timo Sajavaara et al., "Potku - A New Analysis Software for Heavy Ion Elastic Recoil Detection Analysis", 11th European Conference on Accelerators in Applied Research and Technology (ECAART11), Namur, Belgium, 8–13.9.2013.