

Groundhog-sovellusprojekti

Iiro Iivanainen
Harri Linna
Jere Pakkanen
Riikka Vilavaara

Projektiraportti

Versio: 1.0.0
Julkinen
21.10.2021

Jyväskylän yliopisto
Informaatioteknologian tiedekunta

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2021		
Tilaaja	__.__.2021		
Ohjaaja	__.__.2021		

Tietoa dokumentista

Tekijät:

Iiro Iivanainen (II)	<code>iiro.h.iivanainen@student.jyu.fi</code>
Harri Linna (HL)	<code>harri.s.linna@student.jyu.fi</code>
Jere Pakkanen (JP)	<code>jere.j.pakkanen@student.jyu.fi</code>
Riikka Vilavaara (RV)	<code>riikka.k.vilavaara@student.jyu.fi</code>

Dokumentin nimi: Groundhog-projekti, Projektiraportti

Sivumäärä: 62

Tiedosto: projektiraportti.tex

Tiivistelmä: Groundhog-projekti kehitti ohjelmiston Geologian tutkimuskeskukselle tukemaan tomografialaboratorion näytteiden analyysiä. Ohjelmisto helpottaa erityyppisten tutkimusdatojen yhdistämistä, hallintaa ja visualisointia. Dokumentissa kuvataan Groundhog-projektin toteutunutta läpivientiä ja eroja suunnitelmaan verrattuna tavoitteiden ja tulosten, tehtävien ja työmäärien, prosessin ja aikataulun sekä riskiarvioiden ja riskienhallinnan osalta. Lisäksi dokumentissa kuvataan projektiin osallistuneiden opiskelijoiden kokemuksia ja oppimaa. Toteumaa verrataan projektisuunnitelman versioon 1.0.0.

Avainsanat: Aikataulu, kokemukset, käytänteet, projektiraportti, prosessi, resurssit, riskit, roolit, sovellusprojekti, tavoitteet, tehtävät, tulokset, työmäärät, työvaiheet.

Muutoshistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.0.1	31.5.2021	Kirjoitettiin johdanto.	HL
0.0.2	2.6.2021	Kirjoitettiin sovelluksen tavoitteista ja päivitettiin termilista.	HL
0.0.3	4.6.2021	Kirjoitettiin sovelluksen tavoitteista, tuloksista ja oppimistavoitteista.	HL
0.0.4	8.6.2021	Kirjoitettiin projektiorganisaatiosta.	HL
0.0.5	15.6.2021	Kirjoitettiin tiloista ja laitteista, kehitystyökaluista sekä dokumentointityökaluista.	HL
0.0.6	16.6.2021	Kirjoitettiin luennoista ja perehdytyksistä.	HL
0.0.7	29.6.2021	Kirjoitettiin palaverista ja tiedotuksesta.	HL
0.0.8	30.6.2021	Kirjoitettiin tiedostojen nimeämisestä, hakemistorakenteesta, lähdekoodista, testauksesta, versiohallinnasta ja -numeroinnista sekä katselmoinneista ja tulosten hyväksymisestä.	HL
0.0.9	1.7.2021	Kirjoitettiin tulosten koostamisesta ja toimittamisesta.	HL
0.0.10	4.7.2021	Kirjoitettiin rooleista ja vastuualueista.	HL
0.0.11	6.7.2021	Asemoitiin kansilehti uudelleen.	HL
0.0.12	7.7.2021	Sirrettiin tietoa projektista dokumenttiluokkaan ja päivitettiin termilistaa.	HL
0.0.13	8.7.2021	Muokattiin sovelluksen toteutuneita tavoitteita.	HL
0.0.14	9.7.2021	Muokattiin tavoitteiden toteutumista ja tuloksia.	HL
0.0.15	10.7.2021	Kirjoitettiin tehtäväkokonaisuuksista ja työmääristä.	HL
0.0.16	11.7.2021	Kirjoitettiin projektinhallinnan työmääristä, palaverien työmääristä, esitutkimuksen työmääristä, vaatimusmäärittelyn työmääristä, suunnittelun työmääristä ja toteutuksen työmääristä.	HL
0.0.17	13.7.2021	Muokattiin tehtäväkokonaisuuksia ja työmääriä.	HL
0.0.18	14.7.2021	Kirjoitettiin uudelleen tehtäväkokonaisuudet ja työmäärät sekä resursseista ja työmääristä.	HL
0.0.19	19.7.2021	Kirjoitettiin prosessista ja muokattiin käytänteitä.	HL
0.0.20	20.7.2021	Kirjoitettiin prosessista ja projektin kehitysvaiheista.	HL
0.0.21	21.7.2021	Kirjoitettiin uudelleen projektin kehitysvaiheista ja muokattiin lähdeluettelo.	HL
0.0.22	16.8.2021	Kirjoitettiin riskeistä ja niiden hallinnasta sekä tiivistelmä.	HL

Versio	Päivämäärä	Muutokset	Tekijät
0.1.0	17.8.2021	Versio 0.0.22 numeroitiin uudelleen julkistamista varten.	HL
0.1.1	23.8.2021	Kirjoitettiin jäsenten kokemuksia ja oppimaa sekä muokattiin palautteen mukaan.	HL
0.1.2	25.8.2021	Kirjoitettiin jäsenten kokemuksia ja oppimaa.	HL
0.1.3	10.9.2021	Korjattiin lukua 7.	HL
0.1.4	12.9.2021	Korjattiin lukua 8.	HL
0.1.5	14.10.2021	Korjattiin lukua 6.	HL, JP
0.1.6	17.10.2021	Korjattiin lukua 5.	HL
0.2.0	18.10.2021	Tehtiin palautteen mukaiset korjaukset.	HL
0.2.1	20.10.2021	Päivitettiin hakemistorakennetta.	HL
1.0.0	21.10.2021	Tehtiin palautteen mukaiset korjaukset.	HL

Tietoa projektista

Tekijät:

Iiro Iivanainen (II)	<code>iiro.h.iivanainen@student.jyu.fi</code>
Harri Linna (HL)	<code>harri.s.linna@student.jyu.fi</code>
Jere Pakkanen (JP)	<code>jere.j.pakkanen@student.jyu.fi</code>
Riikka Vilavaara (RV)	<code>riikka.k.vilavaara@student.jyu.fi</code>

Tilaaajan edustajat:

Jukka Kuva	<code>jukka.kuva@gtk.fi</code>
Arttu Miettinen	<code>arttu.i.miettinen@jyu.fi</code>

Ohjaajat:

Jukka-Pekka Santanen	<code>santanen@mit.jyu.fi</code>
Juuso Tuononen	<code>juuso.j.tuononen@student.jyu.fi</code>

Yhteystiedot:

Sähköpostilistat	<code>groundhog@korppi.jyu.fi,</code> <code>groundhog_opetus@korppi.jyu.fi</code>
Sähköpostiarkistot	<code>korppi.jyu.fi/kotka/servlet/list-archive/groundhog/,</code> <code>korppi.jyu.fi/kotka/servlet/list-archive/groundhog_opetus/</code>
WWW-sivut	<code>sovellusprojektit.it.jyu.fi/groundhog/,</code> <code>mit.jyu.fi/palvelut/sovellusprojektit/groundhog/</code>

Sisällys

1	Johdanto	1
2	Termit	2
3	Tavoitteiden toteutuminen ja tulokset	4
3.1	Taustaa ja tarpeita	4
3.2	Sovelluksen toteutuneet tavoitteet	4
3.3	Sovelluksen toteutumattomat tavoitteet	5
3.4	Toteutetun sovelluksen rakenne	6
3.5	Dokumentit ja muut materiaalit	8
3.6	Oppimistavoitteet	10
4	Resurssit	11
4.1	Projektiorganisaatio	11
4.2	Tilat ja laitteet	12
4.3	Kehitystyökalut	12
4.4	Dokumentointityökalut	13
4.5	Luennot ja perehdytykset	14
5	Käytänteet	16
5.1	Palaverit	16
5.2	Tiedotus	17
5.3	Tiedostojen nimeäminen	18
5.4	Hakemistorakenne	19
5.5	Lähdekoodi	20
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	Roolit, vastualueet ja tehtävät	27
6.1	Roolit ja vastualueet	27
6.2	Tehtäväkokonaisuudet ja työmäärät	29
6.3	Projektinhallinnan työmäärät	32
6.4	Vaatimusmäärittelyn työmäärät	34
6.5	Suunnittelun työmäärät	35
6.6	Toteutuksen työmäärät	36

6.7	Testauksen työmäärät	37
6.8	Resurssit ja työmäärät	38
6.9	Iivanaisen työtunnit tehtäväkokonaisuuksittain	38
6.10	Linnan työtunnit tehtäväkokonaisuuksittain	39
6.11	Pakkasen työtunnit tehtäväkokonaisuuksittain	40
6.12	Vilavaaran työtunnit tehtäväkokonaisuuksittain	41
7	Prosessi ja aikataulu	43
7.1	Prosessi	43
7.2	Projektin kehitysvaiheet	44
7.3	Aikataulu	45
7.4	Ryhmän työtunnit viikoittain	46
7.5	Iivanaisen työtunnit viikoittain	48
7.6	Linnan työtunnit viikoittain	49
7.7	Pakkasen työtunnit viikoittain	50
7.8	Vilavaaran työtunnit viikoittain	51
8	Riskit ja niiden hallinta	52
8.1	Riskien todennäköisyydet ja vakavuudet	52
8.2	Aikaresurssien rajallisuus	53
8.3	Tilojen ja laitteiden puute	54
8.4	Ryhmän tietotaitojen puute	55
8.5	Ryhmän jäsenten muut sitoumukset	55
8.6	Ominaisuuksien toteuttamisen vaikeus	56
9	Jäsenten kokemuksia ja oppimaa	57
9.1	Iivanaisen kokemuksia ja oppimaa	57
9.2	Linnan kokemuksia ja oppimaa	58
9.3	Pakkasen kokemuksia ja oppimaa	58
9.4	Vilavaaran kokemuksia ja oppimaa	59
10	Yhteenveto	60
	Lähteet	61

1 Johdanto

Groundhog-projekti kehitti ohjelmiston Geologian tutkimuskeskukselle tukemaan tomografialaboratorion näytteiden analyysiä. Ohjelmisto helpottaa erityyppisten tutkimusdatojen yhdistämistä, hallintaa ja visualisointia. Dokumentissa kuvataan Groundhog-projektin toteutunutta läpivientiä tavoitteiden ja tulosten, tehtävien ja työmäärien, prosessin ja aikataulun sekä riskiarvioiden ja riskienhallinnan osalta. Lisäksi dokumentissa kuvataan projektiin osallistuneiden opiskelijoiden kokemuksia ja oppimaa. Toteumaa verrataan projektisuunnitelman versioon 1.0.0.

Dokumentin laatimisessa on hyödynnetty projektien Liikkuva [12], Peltihamsteri [11] ja Potku [1] raportteja sekä *Tietotekniikan Sovellusprojektien ohjetta* [14]. Dokumentin lisäksi projektiryhmä laati vaatimusmäärittelyn [10], jossa kuvataan kehitetyn sovelluksen suunnitellut sekä toteutuneet tiedot ja toiminnot. Projektiryhmä suunnitteli vaatimusten pohjalta arkkitehtuurin [2], käyttöliittymän [4] ja luokkarakenteen [5]. Kehitetyn sovelluksen toteutusratkaisuja ja jatkokehitysideoita kuvataan sovellusraportissa [7]. Sovelluksen laadunvarmistus hoidettiin järjestelmätestauksena, josta laadittiin testaussuunnitelma [9] ja testauskertojen raportit [8].

Dokumentin luvussa 2 määritellään projektissa käytettyä käsitteistöä. Luvussa 3 arvioidaan kuinka projektille asetetut tavoitteet toteutuivat. Luvussa 4 esitellään projektiin osallistuneet henkilöt sekä heidän käyttöönsä annetut tilat, laitteet ja ohjelmat. Luvussa 5 kuvaillaan projektin käytänteiden toteutumista. Luvussa 6 esitetään henkilöiden vastualueita sekä tehtävien jakautumista ja työmääriä. Luvussa 7 kuvataan projektin toteutunut prosessi sekä verrataan toteutunutta aikataulua suunniteltuun. Luvussa 8 arvioidaan riskienhallinnan toteutumista. Luvussa 9 pohditaan ryhmän jäsenten oppimaa ja kokemuksia projektista.

2 Termit

Dokumentissa esiintyvät aihealueen termit ovat seuraavat:

Dataliitos	on kartan ja vähintään yhden liitedatan välinen liitos, johon liittyy liitoskohta ja metadataa.
Kartta	on kolmiulotteinen harmaasävykuva, jota voidaan kuvata esim. kuvasiivujen avulla.
Karttatiedosto	on tiedosto, joka luetaan sovellukseen kartaksi.
Kuvakulma	on koordinaatiston avulla ilmoitettava suunta, josta näytekuvaa tarkastellaan.
Kuvapino	on näytekuvasta muodostettu pino kaksiulotteisia kuvia.
Leikkauskuva	on näytekuvan halki kulkevaa leikkausta esittävä kaksiulotteinen kuva.
Liitosdata	sisältää näytteeseen liittyvät mittaustulokset sekä muut tiedostot ja tiedot.
Liitoskohta	on kartalla sijainti, johon yksittäinen dataliitos liittyy. Se voi olla piste, taso, suora tai muu geometrinen muoto.
Malli	on karttatiedoston esitystapa, jossa karttatiedoston kolmiulotteinen data projisoidaan kaksiulotteiseksi kuvaksi.
Metatieto	on näytteeseen tai liitedataan liittyvää kuvailevaa tietoa, kuten tiedoston laatijan nimi, maantieteellinen sijainti tai päivämäärä.
Näyte	on kokonaisuus, joka sisältää kaiken sovelluksessa auki olevan tiedon liittyen yhteen geologiseen näytekappaleeseen.
Näytekuva	on geologista näytekappaletta esittävä kuva, joka voi olla kolmiulotteinen.
Pääakseli	on näytekuvan sivua vastaan kohtisuorassa oleva X-, Y- tai Z-koordinaattiakseli.
Siivu	on yksittäinen kaksiulotteinen kuva kuvapinosta.
Syvyys	kertoo, monesko kuva tietystä kuvapinosta on kyseessä.
Tilakartta	on malli, joka esittää kartan avaruuden dimensioita.

Dokumentissa esiintyvät kehitysvälineisiin, kehitystekniikoihin ja projektin hallintaan liittyvät termit ovat seuraavat:

AvalonDock	on käyttöliittymäkirjasto, joka mahdollistaa WPF-sovellusten ikkunoinnin, kuten moderneissa IDE:issä.
C#	on Microsoftin kehittämä ohjelmointikieli .NET-alustalle.
ExcelDataReader	on Microsoft Excel -tiedostojen käsittelyyn kehitetty C#-kirjasto.
Git	on hajautettu versiohallintatyökalu, jonka avulla voidaan hallita lähdekoodia.
ImageMagick	on tehokas kuvankäsittelyohjelma, joka tukee yli 100 kuvaformaattia.
Magick.NET	on ImageMagick-ohjelmaa tukeva kuvankäsittelyn mahdollistava C#-kirjasto.
SharpGL	on OpenGL-kirjasto, jolla voidaan piirtää tietokonegrafiikkaa WPF-sovelluksen komponenttiin.
WPF	on Windows-käyttöjärjestelmän graafisten käyttöliittymien kehittämiseen tarkoitettu sovelluskehys .NET-alustalle.
WpfExToolkit	on käyttöliittymäkirjasto, joka sisältää valmiita WPF-sovelluksissa käytettäviä komponentteja.

3 Tavoitteiden toteutuminen ja tulokset

Luvussa käsitellään sovellukselle asetettujen tavoitteiden toteutumista, ryhmän oppimiselle ja muille tuloksille asetettuja tavoitteita ja niiden toteutumista. Projektin aikana toteutettiin sovelluksesta prototyyppi, joka sisältää suurimman osan pakollisista toiminnoista. Toteutusvaiheen loppupuolella sovittiin tilaajan kanssa joitakin tärkeitä toimintoja ohjelmiston jatkokehitykseen. Projektiryhmän laatima sovellusraportti [7] ja vaatimusmäärittely [10] antavat hyvät lähtökohdat jatkokehitykseen. Dokumentit ja muut materiaalit saatiin laadittua suunnitellusti sekä ryhmän jäsenten oppimistavoitteet toteutuivat.

3.1 Taustaa ja tarpeita

Projektin tilaajana toiminut Geologian tutkimuskeskus on työ- ja elinkeinoministeriön alainen kansainvälisesti suuntautunut asiantuntijaorganisaatio, jonka toimipaikkoja on Suomessa kuudella eri paikkakunnalla. Geologian tutkimuskeskus tutkii maaperää ja sen luonnonvaroja tuottaen ratkaisuja elinkeinotoiminnan kestävään kehittämiseen. Tutkimusalueita ovat esimerkiksi ilmastonmuutos, materiaalit ja kiertotalous, liikenteen sähköistyminen sekä hiilineutraali yhteiskunta.

Tomografialaboratoriolla on käytössään röntgensädeskanneri, jonka tuottamaa kolmiulotteista matriisimuotoista harmaasävykuvaa analysoidaan FEI PerGeos -ohjelmistolla. Projektissa kehitettävän sovelluksen päämääränä oli muun tyyppisen tutkimusdatan liittäminen tomografialaitteiston tuottamaan harmaasävykuvaan.

Sovellukselle asetetut vaatimukset olivat suurimmaksi osaksi toiminnallisia vaatimuksia. Sovelluksen suunnittelu suoritettiin tilaajan edustajien kanssa palavereissa. Projektiryhmä esitteli tilaajan edustajille sovellussuunnitelmia ja teki niihin ehdotettuja muutoksia saamansa palautteen perusteella. Toteutusvaihetta edeltäneiden palaverien pöytäkirjat antavat hyvän kuvan ohjelmiston käyttöliittymän, arkkitehtuurin ja luokkarakenteen suunnittelusta.

3.2 Sovelluksen toteutuneet tavoitteet

Pakollisten ja tärkeiden vaatimusten osalta toteutettu sovellus tarjoaa käyttäjälle käyttöliittymän, jonka avulla käyttäjä voi liittää haluamansa tiedoston karttaa esittävään koordinaatistoon. Ohjelma tukee teksti-, kuva-, ja taulukkomuotoisten tiedostojen esitystä, jotka merkitään karttaan pistemäisinä liitoskohtina. Kolmiulotteista tomografiakuvaa esitetään ohjel-

massa kolmella pääakselien suuntaisilla leikkauskuvilla. Liitedataa visualisoidaan sovelluksessa graafisesti sekä esitetään listana.

Kehitetty sovellus on teksti-, kuva- tai taulukkomuotoisen tutkimusdatan hallintajärjestelmä. Sitä voidaan käyttää millä tahansa perustietokoneella, johon on asennettuna Windows-käyttöjärjestelmä. Sovelluksesta toteutettiin toiminnoiltaan suunniteltua rajoittuneempi prototyyppi, joka täyttää tärkeimmät sovellukselle asetetuista tavoitteista. Kehitettävän sovelluksen tärkeimmät kokonaistavoitteet liittyivät liitedatan lukemiseen, näyttämiseen ja yhdistämiseen. Kokonaistavoitteet muodostivat selkeät tehtäväkokonaisuudet, joiden ryhmitely on nähtävissä vaatimusmäärittelyssä [10].

Sovelluksella voidaan käsitellä suuria kolmiulotteisia kuvatiedostoja lukematta niitä kokonaan muistiin. Sovelluksella voidaan selata pääakselien suuntaisia kaksiulotteisia leikkauskuvia, jotka ovat yleisimmät katselukulmat. Kuvapisteisiin pystyy liittämään muun tyyppistä liitedataa ja liitoskohdat visualisoidaan pisteinä. Käyttöliittymä ja luokkarakenne sijoitettiin lähdekoodin tasolla erillisiin projekteihinsa, minkä ansiosta niitä voidaan kehittää toisistaan riippumattomasti.

Tilaaajan edustajien laatimassa aihekuvauksessa työmäärien arvioitiin painottuvan edellä kuvattuihin toimintoihin ja käyttöliittymään. Toteutuneet tehtäväkokonaisuudet vastasivat tarkasti suunniteltua, joten kehitetty sovellus vastasi täsmällisesti tilaaajan asettamia kokonaistavoitteita, vaikkakin tavoitteet saavutettiin vain osittain. Tilaaajan edustajat olivat kertomansa mukaan varautuneet tällaiseen lopputulokseen, joten tämän kaltainen prototyyppi oli tilaaajan puolelta odotettavissa. Tästä syystä projektin tavoitteet toteutuivat hyväksyttävästi. Toimintojen rajaus onnistui siinä mielessä hyvin, etteivät prototyyppiin toteutetut ominaisuudet jääneet keskeneräisiksi. Teknisen ohjaajan mukaan sovellus ei ole vielä valmis tuotantokäyttöön, joten sovellukselle laadittiin suunnitelmat jatkokehitystä varten.

3.3 Sovelluksen toteutumattomat tavoitteet

Sovelluskehityksen vaatimuksien osalta uusien liitedatojen tyyppien lisääminen tai näyttäminen ei ole tehty kovinkaan helpoksi. Vaikka sovellus tukee uusien käyttöliittymän komponenttien lisäämistä, luokkarakenteen laajentaminen voi tuottaa ongelmia. Liitedatat eivät nimittäin toteuta mitään dokumentoitua rajapintaa, mikä edellyttää jatkokehittäjiltä rajapintojen suunnittelua. Tämä pyrittiin huomioimaan jatkokehittäjille suunnatuissa ohjeissa.

Raportointityökalun vaatimusten osalta raporttien tekeminen siirrettiin kokonaisuudessaan jatkokehitykseen. Raportointityökalut olivat alhaisen prioriteetin toimintoja, joiden toteuttaminen olisi edellyttänyt vaatimusmäärittelyn tarkentamista vähintäänkin raporttien sisällön, muotoilun ja tiedostomuodon osalta. Tämän tyyppisessä näytteiden analyysiä ja tut-

kimusdataa yhdistelevässä ohjelmistossa raportointityökalun rooli on työnkulun kannalta merkittävä. Raportit mahdollistaisivat tiedon saavutettavuuden myös ohjelman ulkopuolella.

Sovelluksen tallentamisen ja lataamisen vaatimusten osalta käyttäjä pystyy säilyttämään tekemänsä muutokset ohjelmassa, mutta toteutusratkaisu ei ole kuitenkaan taaksepäin yhteensopiva. Tekninen ohjaaja ehdotti keskitettyä tallennustoimintoa, joka päätettiin jättää jatkokehitykseen sovelluksen oltua muilta osin vielä keskeneräinen. Tästä syystä jatkokehityksessä tulee suunnitella ja toteuttaa järjestelmä, joka vastaa kokonaan ohjelman tilan tallentamisesta ja käyttää dokumentoitua tallennusmuotoa varmistamaan versioiden taaksepäin yhteensopivuus.

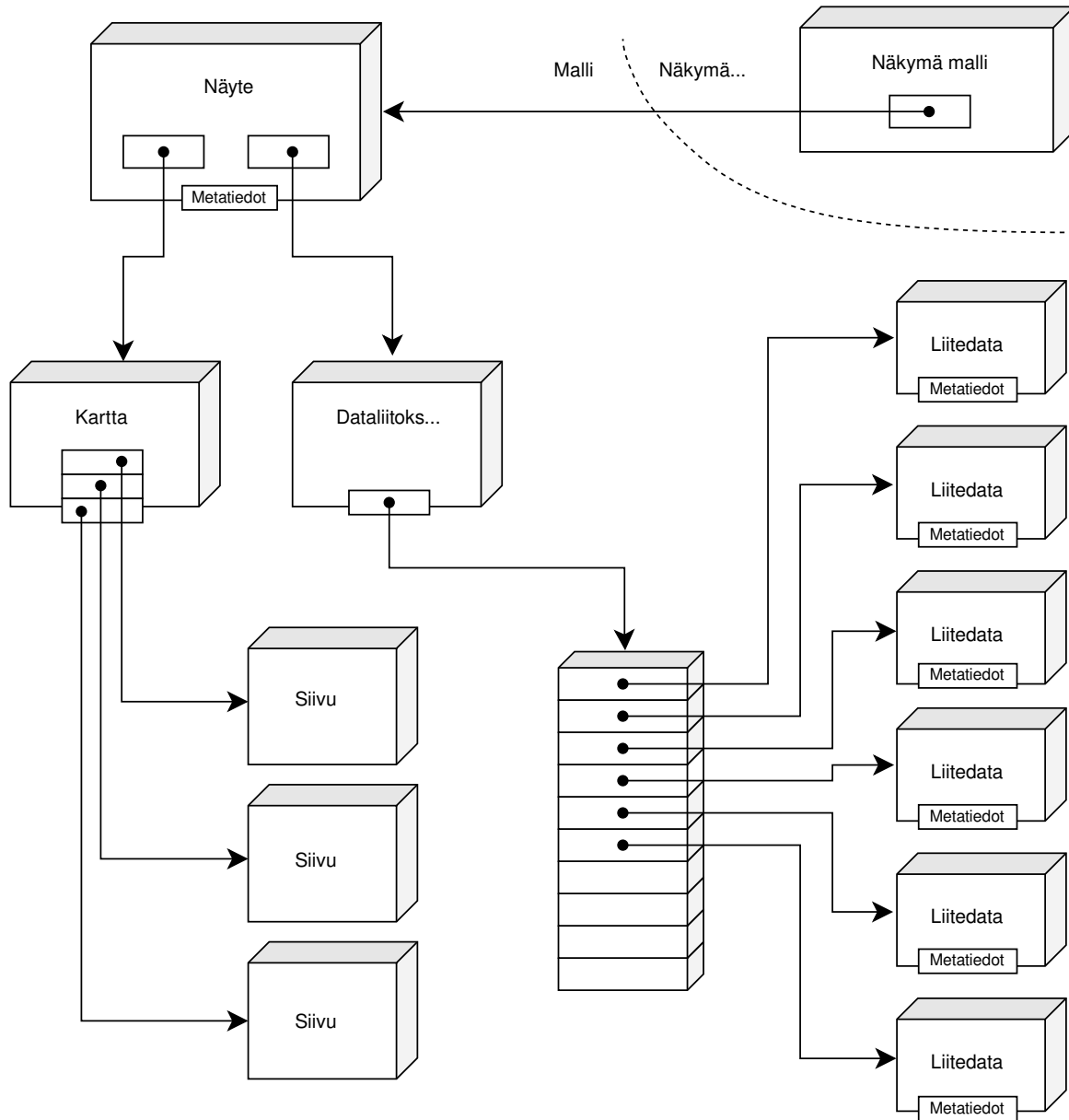
Suuri osa valinnaisista toiminnoista siirrettiin jatkokehitykseen. Vaikka toiminnot olivat projektin aikana valinnaisia, niiden perusteella kykenee hahmottamaan sovelluksen kehitysuunnan. Erityisesti vapaan kuvakulman toiminnot tulevat vaikuttamaan sekä tiedoston käsittelyyn että käyttöliittymään. Karttakuvia tulisi voida käsitellä vapaasta kuvakulmasta kopioimatta alkuperäistä kuvaa eri kuvakulmista. Vasta tämän jälkeen vapaata kuvakulmaa voidaan esittää käyttöliittymässä. Kolmiulotteisen tilakartan prototyypin kykenee esittämään karttakuvaa vain kolmesta eri kuvakulmasta. Niinpä kuvapinojen kopioiminen kovalevyille joudutaan valitettavasti myöhemmin korvaamaan toteutuksella, joka kykenee käsittelemään alkuperäistä karttakuvaa riittävän nopeasti sekä ilman tiedostokoon rajoitteita. Kaikista tärkein ja innovatiivisin ominaisuus jäi siten vielä toteuttamatta.

3.4 Toteutetun sovelluksen rakenne

Sovellus rakentuu kolmiulotteisen matriisimuotoisen harmaasävykuvan ympärille, johon yhdistetään muuta tutkimusdataa ikään kuin kartaksi. Suuresta karttatiedoston koosta johdun kuvapinoja vältetään lukemasta kokonaan muistiin. Kuvapinot tallennetaan kovalevyille kolmesta eri katselukulmasta, jolloin kuvien selaaminen on nopeaa. Kaksiulotteiset kuvat näytetään käyttöliittymässä valitusta syvyydestä. Käyttäjä voi myös tarkentaa ja kohdentaa kuvia vierityspalkkien avulla sekä määrittää kuvien harmaasävyaskaalan. Kuvankäsittely tapahtuu nopeasti ajoaikana eivätkä muutokset vaikuta alkuperäiseen karttatiedostoon.

Kuvassa 3.1 esitetään sovelluksen arkkitehtuuri käyttäen luvun 2 vakiintunutta sanastoa. Arkkitehtuuri on jaettu karkeasti käyttöliittymään ja luokkarakenteeseen. Käyttöliittymän osalta noudatettiin MVVM-arkkitehtuuria (*Model-View-ViewModel*), jonka mukaisesti käyttöliittymän elementit sidottiin ohjelman sisältämiin tietorakenteisiin. Käyttöliittymän ja niin kutsutun näkymämallin sitominen erottaa menetelmän perinteisestä tapahtumalähtöisestä MVC-arkkitehtuurista (*Model-View-Controller*). Luokkarakenne on taas käyttöliittymästä

erillinen projekti, joka noudattaa kerrosarkkitehtuuria. Kuvassa 3.1 kuvataan luokkarakenteen tasolla sovelluksessa esiintyvien muuttujien suhteita, mikä muodostaa sovelluksen arkkitehtuurisuunnitelman.



Kuva 3.1: Sovelluksen rakenne ja arkkitehtuuri.

Valinnaisten vaatimusten osalta sovellukseen suunniteltiin ja toteutettiin kolmiulotteisen tilakartan prototyyppi. Tilakarttaa suunniteltiin projektin alkuvaiheista lähtien, mutta prototyyppi saatiin toteutettua vasta projektin väliesittelyn jälkeen toukokuun alussa. Muutokset olisivat vaikuttaneet merkittävästi sovelluksen rakenteeseen, joten kolmiulotteinen tilakartta päätettiin siirtää jatkokehitykseen. Tilakartan avulla käyttäjä voisi tarkastella karttaa PerGeos-ohjelman tavoin vapaavalintaisesta kuvakulmasta. Fiji-ohjelman kaltainen vapaa-

valintainen leikkauskuvan ja näyttäminen edellyttäisi karttatiedoston lataamista kokonaan muistiin, mikä jätettiin toteuttamatta. Ennen tilakartan suunnittelua ja toteuttamista oli vielä epäselvää voisiko sellaista ylipäänsä toteuttaa kohtuullisessa ajassa tai soveltuisiko sellainen suorituskykynsä puolesta käytettäväksi. Vaikka prototyyppi rajattiin toteutuksen ulkopuolelle, sillä saatiin vahvistettua merkittävän jatkokehitysidean toteuttamiskelpoisuus.

3.5 Dokumentit ja muut materiaalit

Sovelluksen lisäksi projektiryhmä toteutti seuraavat tulokset:

- **Ajankäyttöraportti** sisältää ryhmän jäsenten työtunnit, sekä niiden jakautumisen eri tehtäväkokonaisuuksille ja tehtäville.
- **Arkkitehtuurisuunnitelma** sisältää korkean tason kuvauksen ohjelmiston rakenteesta käyttäen projektin vakiintunutta termistöä.
- **Esittelymateriaali** sisältää väliesittelyn esitysgrafiikan. Sovellusprojektin aikana järjestettiin yksi esittelytilaisuus.
- **Itsearviointit** sisältävät ryhmän jäsenten arviot omasta toiminnastaan, kokemuksistaan ja oppimisestaan projektin aikana. Ryhmän jäsenet päättävät itsenäisesti sijoitetaanko heidän itsearviointinsa julkiseen projektikansioon.
- **Kehitysympäristön pystytysohjeet** sisältävät ohjeet ryhmän käyttämän kehitysympäristön käyttöönottoon jatkokehityksen helpottamiseksi. Jatkokehittäjille suunnatut ohjeet päätettiin sisällyttää sovellusraporttiin.
- **Käyttöliittymäsuunnitelma** sisältää sovelluksen päänäkymän ja sen sisältämien komponenttien näkymät.
- **Käyttöohjeet** sisältävät kuvauksen käyttöliittymästä ja neuvovat, miten sovellusta käytetään. Käyttöohjeessa esitettiin käyttäjien työnkulun vaiheet luonnollisessa järjestyksessä sekä mahdolliset käyttäjiin vaikuttavat rajoitteet.
- **Luokkadokumentaatio** generoidaan automaattisesti lähdekoodin sisältämistä dokumentaatiokommenteista.
- **Luokkasuunnitelma** sisältää ohjelmiston sisäisen rakenteen kuvattuna UML-notaation kaltaisena luokkakaaviona.
- **Lähdekoodi** sisältää sovelluksen lähdekoodin kommentteineen.
- **Palaverien dokumentit** sisältävät palaverien esityslistat, tilakatsaukset ja pöytäkirjat.
- **Projektiraportti** kuvaa projektin toteutunutta läpivientiä, jota verrataan projektisuunnitelmaan kuvaten poikkeamien syitä ja vaikutuksia.
- **Projektisopimus** sisältää projektin sopimusehdot mukaanlukien lähdekoodin ja muiden tulosten lisensointiehdot.
- **Projektisuunnitelma** on projektin läpivientiä kuvaava suunnitelma, jossa kuvataan

projektin tavoitteet sekä projektinhallinnan keinot tavoitteiden saavuttamiseksi.

- **Sovellusraportti** kuvaa toteutetun sovelluksen keskeiset toteutusratkaisut ja mahdolliset jatkokehitysideoita. Lisäksi sovellusraportti sisältää ohjeet kehitysympäristön pystyttämiseen ja käyttöliittymän kuvauksen.
- **Sähköpostiarkistot** sisältävät projektin yhteisille sähköpostilistoille lähetetyt sähköpostiviestit sekä ryhmän jäsenten ja ohjaajien väliseen viestintään suunnatun sähköpostilistan sähköpostiviestit.
- **Testausraportit** kuvaavat eri testauskerroilla suoritettujen testitapausten havaintojen kuvaukset mukaan lukien perustelut ja jatkotoimenpiteet havaittujen virheiden osalta.
- **Testausraporttipohja** on lomake, johon testauskerran huomiot kirjataan. Täydennetty raporttipohja julkistetaan testausraporttina.
- **Testaussuunnitelma** kuvaa testausympäristön kokoonpanon ja suoritettavat testitapaukset.
- **Vaativuusmäärittely** sisältää sovellukseen kehitettävät tiedot ja toiminnot vaatimuksina sekä niiden prioriteetit ja tilat projektin päättyessä
- **Valmisohjelma** sisältää ajettavan ohjelman sekä tarvittavat ohjelmakirjastot yhdessä kansiossa.
- **Verkkoesite** sisältää projektin WWW-esittelyn.

Kääntö-, asennus-, käyttö- ja kehitysympäristön pystytysohjeet suunniteltiin koostettavan yhteen kattavaan käyttäjien manuaaliin. Projektissa ei tarvinnut laatia erillistä asennusohjetta, sillä valmisohjelma sisältää kaiken tarvittavan sovelluksen käyttöönottoon. Valmisohjelmasta tehtiin jakelupaketti, josta ajettava ohjelma voidaan asentaa tietokoneelle. Käännös-ohjeet sisällytettiin kehitysympäristön pystytysohjeisiin, sillä kehitysympäristön toimivuus oli helpointa todentaa käännettävällä ohjelmalla.

Sovellusta kehitettiin Microsoftin Visual Studiolla, jonka sisältämän nuget-pakettihallinnan kautta ulkoiset kirjastot saatiin liitettyä helposti mukaan valmisohjelmaan. Kehitysympäristön pystytysohjeet sisällytettiin sovellusraporttiin [7] erillisten ohjeiden sijaan. Jatkokehityksessä voisi olla järkevää kääntää kirjastot lähdekoodista ilman nuget-pakettihallintaa, jotta kirjastopäivityksiä voidaan ylläpitää hallitusti.

Käyttöohjeet päädyttiin sijoittamaan versiohallintaan HTML-dokumenttina, joka tulee valmisohjelman mukana ja avautuu lokaalisti sovelluksen *Help*-valikosta käyttäjän oletusselaimen. Käyttöohje kirjoitettiin kaikista muista dokumenteista poiketen englanniksi. Sovellusraportista löytyvät vastaavat käyttöliittymän kuvaukset suomeksi.

Ryhmän jäsenet sitoutuivat sijoittamaan lähdekoodin kommentteineen avoimen lähdekoodin MIT-lisenssin alle ja muut tulokset avointen dokumenttien CC BY 4.0 -lisenssin alle. Projektipäällikön laatimaa erillistä lisenssisitoumusta ei tarvittu, sillä lisenssisitoumus sisällytettiin projektisopimukseen. Lähdekoodin ja luokkadokumentaation paperitulosteet so-

vittiin rajattavan pois projektikansiosta, sillä ajantasaiset versiot molemmista sijaitsevat versiohallinnassa. Ryhmän jäsenien tuntimääriä koskeva ajankäyttöraportti sisällytettiin projektikansioon. Tulokset saatiin toteutettua pääosin suunnitellusti, mutta sovelluksen ohjeissa jouduttiin tekemään rajauksia.

3.6 Oppimistavoitteet

Projektiryhmän jäsenten oppimistavoitteet toteutuivat suunnitellusti. Projekti toteutettiin osana Jyväskylän yliopiston informaatioteknologian tiedekunnan Sovellusprojekti-nimistä projektiopintojaksoa. Projektin päätyttyä projektiryhmän jäsenet tuntevat sovellusprojektin keskeiset projektinhallinnan menetelmät ja osaavat soveltaa niitä käytännön projektityössä.

Keskeiset projektinhallinnan menetelmät olivat

- projektin suunnittelu, seuranta ja edistymisen raportointi,
- projektin ohjaus, ryhmätyöskentely ja viestintä,
- projektin virallisten dokumenttien tuottaminen ja niiden hallinta sekä
- projektin muutosten ja riskien hallinta.

Kaikilla projektiryhmän jäsenillä oli entuudestaan ohjelmointikokemusta. Projektin loputtua jäsenet osaavat valita tavoitteiden kannalta sopivimmat toteutustyökalut sekä käyttää niitä toteutusprojektin edellyttämässä laajuudessa. Projekti antoi myös selkeyttä siihen, haluavatko projektin jäsenet tulevissa projekteissa toimia asiantuntijan vai projektipäällikön tehtävissä.

4 Resurssit

Luvussa käsitellään sovellusprojektin jäsenet ja ohjaajat sekä tilaajan edustajat. Luvussa kuvataan myös projektissa käytössä olleita tiloja, laitteita ja kehitystyökaluja. Projektin resurssit toteutuivat projektisuunnitelmaan kirjatulla tavalla.

4.1 Projektioorganisaatio

Projektioorganisaatio toteutui suunnitellusti. Projektiryhmään kuuluivat Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. Kaikki jäsenet ovat tietotekniikan maisteriopiskelijoita ohjelmisto- ja tietoliikennetekniikan maisteriohjelmassa. Harri Linna toimi projektipäällikkönä ja Jere Pakkanen varapäällikkönä.

Kaikilla projektiryhmän jäsenillä oli entuudestaan ohjelmointikokemusta, mutta projektiryhmän kokemus kyseessä olevan kokoluokan sovellusprojektista sekä siihen liittyvästä projektityöskentelystä oli vähäistä. Sovellusprojektin oheiskursseilla käsiteltiin kuitenkin projektin läpiviennin ja hallinnan taitoja sekä viestintätaitoja, joita projektissa hyödynnettiin ahkerasti.

C#-ohjelmointikieli oli kaikille ryhmän jäsenille tuttu ennestään. Kehitettävä ohjelmisto koostui kolmesta osasta, joihin kehitystiimin jäsenet erikoistuiivat. Kukin jäsen kehitti pääosin yksinään hänen vastuullaan olleita sovellusosioita. Linna keskittyi projektinhallintaan ja dokumenttien laatimiseen. Iivanainen keskittyi liitetietojen hallintaosion kehittämiseen. Pakkanen keskittyi puolestaan kolmiulotteisten karttatiedostojen käsittelyyn ja säikeistämiseen. Vilavaara keskittyi MVVM-arkkitehtuurin mukaisen käyttöliittymän toteuttamiseen.

Tilaaajaorganisaatio oli Geologian tutkimuskeskus ja tilaajan edustajat olivat Jukka Kuva ja Arttu Miettinen. Kuva toimii Geologian tutkimuskeskuksella tutkijana. Miettinen toimii Jyväskylän yliopiston fysiikan laitoksella tutkijana.

Projektin vastaavana ohjaajana toimi informaatioteknologian tiedekunnalta Jukka-Pekka Santanen ja teknisenä ohjaajana Juuso Tuononen. IT-palvelut perustivat projektin käyttöön verkkolevyt ja asettivat projektin jäsenille rajattoman tulostuskiintiön.

Sovellusprojekti-kurssin yhteydessä järjestettävän *Projektiviestintä IT-alalla* -kurssin puheviestinnän opettajana toimi Hanna Kivimäki ja kirjoitusviestinnän opettajana Katja Konturi. *Sovellusprojektin hallintaa, viestintää ja työkaluja* -kurssin opettajana toimi Jukka-Pekka Santanen. Käytettävyysspäivää ei järjestetty ja kolmesta suunnitellusta esittelystä järjestettiin ainoastaan ensimmäinen väliesittely.

4.2 Tilat ja laitteet

Projektin käytössä oli WWW-sivusto, verkkolevy, kaksi sähköpostilistaa ja rajaton tulos-tuskiintiö yliopiston tilojen koronarajoituksista johtuen. Projektiryhmälle varatut työhuo-neet AgC417.1 ja AgC418.1 eivät olleet jäsenten käytettävissä, joten ryhmän jäsenet toteutti-vat sovelluksen etänä käyttäen henkilökohtaisia tietokoneitaan. Sovellus kehitettiin pääosin Windows-käyttöjärjestelmällä.

Sovellusprojekti-kurssin yhteiset tapaamiset järjestettiin Teams-palvelun kautta videopuhe-luina. Niin ikään etäpalaverit järjestettiin Teams-videopuheluina Arttu Miettisen kutsusta, sillä opiskelijoilla oli rajoitetut oikeudet kalenteritapahtumien järjestämiseen. Projektiryh-mä käytti ryhmän sisäisten dokumenttien jakamiseen kurssin etätyöskentelytilan sisältä-mää Teams-kanavaa, joka mahdollisti Office365-palvelun käyttämisen. Esimerkiksi työajan-seurantalomakkeen päivittäminen tehtiin kätevästi Excelin verkkoversiolla.

Maaliskuussa ensimmäisen *Projektiviestintä IT-alalla* -kurssin kahvilatapaamisen aikaan ryh-män jäsenet siirtyivät käyttämään Discord-palvelua. Ryhmä kokoontui noin kolmesti vii-kossa keskustelemaan projektin tilanteesta ja seuraavan tapaamisen ajankohdasta ja tavoit-teista. Discord-palveluun siirtymisen jälkeen viestittely keskustelualueella lisääntyi merkit-tävästi Teams-palveluun verrattuna. Aluksi ryhmän jäsenet olivat hieman vastahakoisia ot-tamaan Discord-palvelua käyttöön henkilökohtaisilla tunnuksillaan, mutta kahvilatapaami-sen jälkeen päätettiin ottaa uusi viestintäkanava käyttöön, kun asiasta keskusteltiin uudel-leen.

4.3 Kehitystyökalut

Projektissa kehitetyn sovelluksen ohjelmointikielenä käytettiin C#:n versiota 8.0 ja ohjelmis-tokehystenä .NETin versiota 5.0. Kehitysympäristöinä käytettiin Visual Studio 2019 -ja Vi-sual Studio Code -ohjelmia. Graafinen käyttöliittymä toteutettiin sovelluskehityksellä WPF (Windows Presentation Foundation). Versiohallintaan käytettiin yliopiston GitLab-versio-hallintajärjestelmää. Luokkadokumentaatio suunniteltiin muodostettavan suoraan lähde-koodista käyttäen DocFX- tai Doxygen-työkalua. Luokkadokumentaatiota ei kuitenkaan lo-pulta muodostettu, mistä sovittiin tilaajan edustajien ja ohjaajien kanssa.

Kehitystyökalut toteutuivat suunnitellusti, mutta työkaluja olisi kannattanut kokeilla käy-tännössä heti suunnitteluvaiheessa. C#-ohjelmointikieli valittiin jäsenten aiemman koke-muksen ja WPF-käyttöliittymän perusteella. Tästä seurasi kuitenkin merkittävä rajoite, että sovellusta kehitettäisiin ja ajettaisiin jatkossa vain Windows-ympäristössä. Tilaajan edusta-jat hyväksyivät kyseisen ratkaisun. Lisäksi Agoran mikroluokkien tietokoneisiin oli asen-

nettuna vanhempi .NET 3.1, joten sovellusta ei voinut myöskään kehittää tai ajaa yliopiston mikroluokissa. Ohjelmointikielen valinnan jälkeen kartoitettiin erilaisia ohjelmakirjastoja, joita hyödyntäen olisi voitu toteuttaa karttatiedostojen lukemisen ja käyttöliittymän ikkunoinnin.

Sopivien kirjastojen Magick.NET ja AvalonDock löydyttyä siirryttiin välittömästi toteutusvaiheeseen 11.3.2021. Koska sovellus tulisi olemaan riippuvainen näistä kriittisen toiminnallisuuden tarjoavista kirjastoista, ohjelmointikieltä ei ollut enää kannattavaa muuttaa jälkeenkään. Erottamalla WPF-käyttöliittymä ja sovelluksen luokkarakenne erillisiin .NET-projekteihin, kirjastojen valinnan aiheuttamien riippuvuuksien vaikutuksia pystyttiin kuitenkin vähentämään. Samaan tapaan luotiin MSTest-projekti kehityksenaikaista yksikköttestaamista varten sekä WPFOpenGL-projekti kolmiulotteista tilakarttaa varten. Erillisillä projekteilla saatiin tehostettua kehitystyökalujen käyttöä. Kehitysympäristön ja -työkalujen rajoituksista huolimatta sovellus toteutui suunnitellusti.

4.4 Dokumentointityökalut

Dokumenttien kirjoittamiseen käytettiin toimisto-ohjelmistona LibreOfficea ja Overleaf-palvelua. Sovelluksen käyttöliittymäsuunnitelma hahmoteltiin GIMP-kuvankäsittelyohjelmalla ja sovelluksen rakennetta kuvaavat kaaviot diagrams.net-palvelussa, joka kuuluu nykyään Office365-verkkopalveluun. Vastaavasti dokumenttien kommentointiin käytetty Kami-palvelu kuuluu Googlen pilvipalveluihin, johon yliopiston tunnukset oikeuttavat rajattoman tallennustilan. Tietokoneen kovalevyn kautta muokattavat tiedostot kannattaakin ilman muuta säilyttää pilvipalvelussa, jonka kautta onnistuu myös dokumenttien muokkaaminen.

Overleaf-palvelu toimii dokumentointityökaluna erinomaisen hyvin, sillä kaikki dokumentit olivat muokattavissa verkossa yhdessä paikassa. Dokumenttien tarkastamisessa käytettiin Kami-palvelua, jolla voidaan tehdä merkintöjä suoraan pdf-dokumenttiin. Kami-palvelu tarjoaa siis saman toiminnallisuuden kuin toimisto-ohjelmien versiohistoria, mutta pdf-dokumentteja ei tarvitse muuntaa docx- tai odt-tiedostoiksi, missä saatetaan kadottaa muotoiluja tai jopa tekstiä. Overleaf-palvelun kautta voidaan myös varmuuskopioida dokumentteja yliopiston GitLab-versiohallintaan. Projektin aikana dokumenttien versiohallintaa ei tarvittu, mutta sitä kannattaa ehdottomasti käyttää, jos useampi henkilö tekee muutoksia samaan dokumenttiin. Vanhan tekstin kommentteihin siirtämisen sijaan voidaan tehdä heti muutoksia alkuperäiseen, sillä edelliseen versioon voidaan aina palata muutoshistorian ja versiohallinnan aikaleiman perusteella. Overleaf-, GitLab- ja Kami-palvelut toimivat hyvin yhteen kirjoittamisen teknisenä työympäristönä ilman erillisiä ohjelmistoasennuksia.

Sovellusprojektin alkuperäiset dokumenttipohjat olivat edelleen ajankohtaisia, mutta ne vaativat pientä päivitystä vastaamaan viime vuosien aikana hioutuneita hyviä käytänteitä. Overleaf-palvelu soveltuu erinomaisesti teknisten raporttien kirjoittamiseen, kunhan dokumenttipohjat ovat ajan tasalla. Projektiraporttien Liikkuva [12], Peltihamsteri [11] ja Potku [1] rakenteet ja tekstikappaleet olivat nimittäin melkein identtiset, joten Sovellusprojektin alkuperäiset dokumenttipohjat kannattaisi päivittää ainakin rakenteen ja ulkoasun muotoilujen osalta.

Projektin aikataulu laadittiin GanttProject-ohjelmistolla ja työajanseurantaan käytettiin Petri Heinosen kehittämää Excel-työkirjaa. GanttProject-ohjelman sijaan olisi kannattanut hyödyntää valmista Excel-lomakepohjaa samaan tapaan kuin ajankäytönseurannassa. Tällöin suunnitelmien ylläpitäminen projektin aikana olisi vienyt vähemmän aikaa ja taulukkolaskentaohjelmalla olisi voinut koostaa monipuolisia kaavioita tilakatsauksiin. Tilakatsaukset olisi kannattanut latoa Beamer-paketilla.

Ryhmän jäsenet jakoivat toisilleen runsaasti tex-tiedostoista käännettyjä pdf-dokumentteja kommentoitavaksi. Raakatekstin ja ranskalaisten viivojen sijaan korjausehdotusten merkitseminen tarkastettavaan dokumenttiin olisi säästänyt merkittävästi työtunteja sekä kirjoittajalta että tarkistajalta.

4.5 Luennot ja perehdytykset

Luennot ja perehdytykset toteutuivat suunnitellusti. Sovellusprojektin ohella kaikki ryhmän jäsenet suorittivat molemmat oheiskurssit. Näillä kursseilla opittiin projektinhallintaan ja projektiviestintään liittyviä taitoja. Oheiskurssien työtunnit kirjattiin työajanseurantaan erillisenä tehtäväkokonaisuutena.

Projektiviestintä IT-alalla -kurssin suoritettuaan opiskelija:

- ymmärtää viestinnän ja vuorovaikutuksen merkityksen projektityöskentelyssä,
- tuntee projektiviestintään liittyviä tekstilajeja ja niiden konventioita,
- osaa laatia keskeisimmät projektissa tarvittavat dokumentit,
- tuntee projektipalaverihin liittyviä käytänteitä,
- osaa valita tilanteeseen sopivan tavan viestiä projektin eri vaiheista sekä
- tuntee projektityöskentelyyn liittyviä ryhmä- ja vuorovaikutusilmiöitä.

Sovellusprojektin hallintaa, viestintää ja työkaluja -kurssi sisälsi seuraavat luennot:

- aloitusluento,
- vaatimusmäärittely,
- versiohallinta,

- projektinhallinta ja johtaminen,
- tekijänoikeusluento.

Dokumentointityökaluista aikataulun ja tehtävienhallinnan suunnitteluun olisi kaivattu perehdytystä, sillä projektinhallinnan kannalta ne olivat tärkeimmät projektipäällikön työkalut. Dokumentointityökaluista toimisto-ohjelmien versiohistoriaan ja muokkaustyökaluihin olisi kaivattu perehdytystä.

5 Käytänteet

Luvussa kuvataan projektin käytänteitä, jotka edesauttoivat projektin läpivientiä, asetettujen tavoitteiden saavuttamista ja tulosten toteuttamista. Käytänteissä ei tapahtunut merkittäviä muutoksia suunniteltuun verrattuna. Käytänteissä olisi ollut runsaasti parantamisen varaa ja projektipäällikön olisi kannattanut nähdä enemmän vaivaa niiden suunnitteluun. Käytänteet oli kirjattu projektisuunnitelmaan liian tiivistetysti, mikä saattoi hankaloittaa yhteisten toimintatapojen omaksumista ja noudattamista.

Sovelluksen suunnittelu olisi kannattanut rajata jo projektin alussa ohjelmoijien vastuualueeksi, sillä he toteuttivat sovelluksen. Projektipäällikön olisi kannattanut keskittyä projektin läpiviennin suunnitteluun enemmän. Toisaalta projektipäällikön on vaikea suunnitella tehtävienjakoa, ellei hänellä ole selkeää kokonaiskäsitystä sovelluksen rakenteesta, mikä edellyttää väistämättä suunnitteluun ja toteutukseen osallistumista.

5.1 Palaverit

Palavereja pidettiin aluksi viikon välein projektin määrittely- ja suunnitteluvaiheessa. Kun projekti eteni toteutusvaiheeseen ja tapaamistarve väheni, palavereja pidettiin enää noin kahden viikon välein. Seuraavan palaverin ajankohta päätettiin aina edellisessä palaverissa. Palaverihin osallistuivat kaikki projektiryhmän jäsenet, ohjaajat ja tilaajan edustajat. Palaverit todettiin laillisiksi, jos niiden esityslistat lähetettiin projektiorganisaatiolle vähintään vuorokautta ennen kyseisen palaverin alkua. Palaveri todettiin päätösvaltaiseksi, jos paikalla oli vähintään yksi projektiryhmän edustaja, yksi tilaajan edustaja ja vastaava ohjaaja.

Palavereissa käsiteltiin edellisen palaverin pöytäkirja, projektipäällikön tilakatsaus (ensimmäistä palaveria lukuun ottamatta), seuraavan palaverin ajankohta, läsnäolijoille sovitut toimenpiteet ja mahdolliset muut esille tulleet asiat. Palavereissa keskusteltiin kehitettävän sovelluksen suunnitteludokumenteista ja toteutusratkaisuista sekä esiteltiin kehitettävän sovelluksen toimintaa. Palavereissa keskusteltiin erityisesti tomografiakuvien käsittelyyn liittyvistä asioista. Keskustelun aikana tehdyt päätökset kirjattiin pöytäkirjaan, josta sovellukseen kehitettävien tietojen ja toimintojen kuvaus kopioitiin myöhemmin vaatimusmäärittelyyn.

Ryhmän jäsenten roolit palavereissa muodostuivat niin, että projektipäällikkö Linna ja varaprojektipäällikkö Pakkanen vuorottelivat puheenjohtajan tehtävässä. Puheenjohtaja johti keskustelua ja huolehti esityslistan mukaisesta keskustelun etenemisestä. Iivanainen ja Vilavaara vuorottelivat sihteerin tehtävässä, johon kuului palaverin muistion ja pöytäkirjan

laatiminen. Jokainen jäsenistä toimi vähintään kerran sekä sihteerinä että puheenjohtajana. Puheenjohtaja tarkasti ja hyväksyi sihteerin laatiman pöytäkirjan ennen sen julkistamista projektiorganisaatiolle. Esityslistat laadittiin projektiryhmän tapaamisissa, joiden aikana laadittiin viikoittainen tilakatsaus ja tehtävienhallinnan tehtävät.

Palaverien pöytäkirjojen laatimiseen käytettiin Overleaf-palvelua. Muita yhteisiä käytänteitä ei sovittu muistioiden tai pöytäkirjojen laatimisesta. Erillisen dokumenttiluokan puuttuessa mallipohjana käytettiin edellisen palaverin pöytäkirjaa, jonka rakenne ja muotoilut vaihtelivat sihteerinä toimineen henkilön mukaan. Tästä syystä projektiryhmän olisi kannattanut käyttää yhteistä työtilaa Overleaf-palvelussa, joka toimii erinomaisena kirjoittamisen työympäristönä ilman erillisiä ohjelmistoasennuksia. Yhteisten käytänteiden puuttumisesta huolimatta pöytäkirjat olivat korkealaatuisia ja vain yksi muistio kadotettiin projektin aikana.

Etätyöskentelyn vuoksi ryhmän tapaamisia järjestettiin Teams- tai Zoom-palvelussa aluksi kerran viikossa. Näitä palveluja jäsenet olivat tottuneet käyttämään muiden kurssien etätaapaamisissa. Toteutusvaiheen alkaessa ryhmän tapaamiset siirtyivät Discord-palveluun, jonka seurauksena tapaamisten määrä kasvoi kolmeen kertaan viikossa. Myöhemmin huomattiin, että kaksi tapaamista viikossa riitti mainiosti. Tekninen ohjaaja kutsuttiin muutaman kerran projektiryhmän suulliseen neuvonpitoon. Muun aikaa hän oli nopeasti tavoitettavissa viestiyhteyden päässä.

5.2 Tiedotus

Tiedotuksen osalta noudatettiin projektisuunnitelmaan kirjattuja käytänteitä. Projektipäällikkö vastasi ensisijaisesti tiedotuksesta projektiorganisaatiolle. Projektiryhmän ulkoisessa viestinnässä jäsenet vastasivat omien vastuualueidensa viestinnästä. Projektiryhmän sisäisessä viestinnässä jäsenet raportoivat työmäärät, töiden etenemisen ja mahdolliset ongelmat projektipäällikölle. Projektiin liittyvistä valinnoista, muutoksista ja vaihtoehtoisista toteutusratkaisuksista keskusteltiin pääasiassa projektipalaverissa. Projektipäällikkö toimitti projektiorganisaatiolle tilakatsauksen sähköpostitse sellaisina viikkoina, jolloin ei pidetty projektipalaveria.

Projektiorganisaation tiedottamista varten perustettiin kaksi sähköpostilistaa, joiden viestit tallentuivat sähköpostiarkistoihin. Sähköpostilistat ja niiden arkistot kuvataan dokumentin metatiedoissa ennen sisällysluetteloa. Ensisijainen sähköpostilista oli tarkoitettu viestintäkanavana koko projektiorganisaatiolle suunnattuun tiedotukseen. Vastaanottajiin kuuluivat ryhmän jäsenet, ohjaajat ja tilaajan edustajat. Toinen sähköpostilista oli tarkoitettu opetuskäyttöön lähinnä ryhmän jäsenten ja ohjaajien väliseen viestintään. Maaliskuun jälkeen ky-

seistä listaa ei juurikaan käytetty, sillä useimmat sähköposteista koskivat myös tilaajan edustajia.

Tiedotus toteutui ilman suurempia ongelmia, mutta aktiivisempi sähköpostiviestintä olisi tehnyt ryhmän toiminnasta läpinäkyvämpää. Esimerkiksi tilaajan edustajat mainitsivat muutaman kerran, että asioista voi kysyä myös sähköpostitse ennen seuraavaa palaveria. Vastaava ohjaaja joutui muutaman kerran kysymään palauttamattomien tulosten perään. Toteutusvaiheen loppupuolella projektiorganisaatio kutsuttiin seuraamaan projektiryhmän tehtävienhallintaa, minkä seurauksena ryhmän jäsenet alkoivat käyttää tehtävienhallintaa enemmän. Projektiryhmän sisäinen tiedotus tapahtui alkuun Teams-työtilassa. Ensimmäisen *Projektiviestintä IT-alalla* -kurssin kahvilatapaamisen aikoihin, joka järjestettiin etätapaamisena, projektiryhmä siirtyi käyttämään Discord-palvelua. Discord-kanavan perustamisen jälkeen ryhmä kääntyi useammin teknisen ohjaajan puoleen koodauksen aikana.

5.3 Tiedostojen nimeäminen

Lähdekooditiedostojen nimeämisessä käytettiin C#-ohjelmointikielen yleisiä käytänteitä [13]. Hakemistot ja tiedostonimet kirjoitettiin englanniksi. Lähdekoodi ja sen kommentit kirjoitettiin englannin kielellä käyttäen UTF-8-merkistöä.

Dokumentit nimettiin sisältöä kuvaavilla nimillä ja dokumentin kielen mukaisesti. Tiedostonimet kirjoitettiin pienillä kirjaimilla ja välilyöntien sijaan käytettiin alaviivoja. Dokumenttien nimiin lisättiin versionumero luvun 5.7 käytänteiden mukaisesti.

Esimerkit ovat palaverien dokumenteista:

- esityslista_2021-03-05.txt
- groundhog_palaveri11_08_6_2021.pdf
- tilakatsaus26.pdf

Esimerkit ovat sovellusprojektin dokumenteista:

- vaatimusmaarittely_0.5.0.pdf
- TestauskerranSuunnitelma_0.3.3.pdf
- groundhog_sovellusraportti_0.1.0.pdf

Dokumenttien nimeämiskäytännöt vaihtelivat suuresti. Projektin nimi jätettiin monesti merkittämättä, sillä asiayhteys vaikutti selvältä. Tiedostonimessä saatettiin käyttää välilyöntien tilalla alaviivaa tai suuraakkosia. Versionumeron sijaan saatettiin käyttää päivämäärää, viikkonumeroa tai juoksevaa järjestysnumeroa. Harvemmin nimiä kuitenkaan muokattiin jälkeenpäin siitä syystä, että dokumentit jaettiin projektiorganisaatiolle sähköpostitse hyper-

linkkien avulla. Muutokset olisivat tehneet linkeistä lukukelvottomat. Käytännössä sallittiin dokumenttikohdaiset käytänteet, kunhan niissä pysyttiin jatkossakin. Projektikansion koostamisen yhteydessä tarkistettiin ovatko tiedostonimet yhteneviä ja asianmukaisia käyttötarkoituksiinsa.

5.4 Hakemistorakenne

Projektin tulokset tallennettiin CD-levylle ja sijoitettiin WWW-sivuille päähakemiston alle listauksen 5.1 hakemistorakenteen mukaisesti.

- ajankaytonseuranta
 - ajankaytonseuranta
- dokumentit
 - projektiraportti
 - projektisuunnitelma
 - sovellusraportti
 - vaatimusmaarittely
- esittelyt
 - lehdistotiedote
 - valiesittely
 - verkkoesite
- jarjestelmatestaus
 - raporttipohja
 - testausraportti
 - testaussuunnitelma
- palaverit
 - esityslistat
 - katselmoinnit
 - poytakirjat
 - tilakatsaukset
- sahkopostiarkistot
 - groundhog
 - groundhog_opetus
- sopimus
 - aihekuvaus
 - projektisopimus
 - yhteen veto
- sovellus
 - geolab_documentation
 - geolab_image
 - geolab_manual
 - geolab_release
 - geolab_source

```
geolab_testdata
sovellussuunnitelmat
  arkkitehtuurisuunnitelma
  kayttoliittymasuunnitelma
  luokkasuunnitelma
taustat-ja-tarpeet
```

Listaus 5.1: Lopputuoteluettelo.

Hakemistorakenne toteutui pääosin suunniteltuna. Suunnitelmaan verrattuna tulokset ryhmiteltiin useampaan hakemistoon. Kehitysympäristön pystytysohjeet siirrettiin sovellusraporttiin [7].

5.5 Lähdekoodi

Lähdekoodin käytänteitä ei ollut määritelty projektisuunnitelmassa. Lähdekoodissa käytetyt aliohjelmat, luokat ja muuttujat nimettiin mahdollisimman kuvaavilla englanninkielisillä nimillä luvun 2 terminlistan mukaisesti sekä noudattaen C#-ohjelmointikielen käytänteitä [13]. Myös lähdekoodin kommentointi tapahtui englanniksi. Lähdekooditiedostojen alkuun lisättiin listauksessa 5.2 kuvattu MIT-lisenssiteksti, jota ei ollut huomioitu suunnitelmassa.

```
Copyright (c) 2021 Iiro Iivanainen, Harri Linna, Jere Pakkanen,
Riikka Vilavaara
```

```
Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
```

```
The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.
```

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
```

SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Listaus 5.2: Ohjelmistolisenssi.

Lähdekoodin kirjoittamisessa huomioitiin .NETin version 5.0 ja C#:n version 8.0 versiokohtaiset ominaisuudet ja käytänteet esimerkiksi lokaalien muuttujien tyyppityksestä. Rivipituuudet rajattiin 80–90 merkkiin, jotta koodin lukeminen olisi miellyttävää jaetulla näytöllä. Listauksen 5.3 katkelma lähdekoodista edustaa kommentointikäytänteitä.

```
#region 3D-tif
/// <summary>
/// Reads a 2D image from 3D tif file.
/// </summary>
/// <param name="filepath">Filepath of the image.</param>
/// <param name="depth">The depth of the 2D slice.</param>
/// <returns>A MagickImage object from the read file.</returns>
/// <exception cref="ImageReadException">
/// Thrown when failed to read image
/// </exception>
public static MagickImage Read3Dtif(string filepath, int depth)
{

    if (!File.Exists(filepath))
        throw new ImageReadException("Image_does_not_exist");
    try
    {
        MagickImageCollection images = new();
        MagickImageCollection meta = new();
        meta.Ping(filepath);
        if (depth < meta.Count && depth >= 0)
        {
            MagickReadSettings settings = new();
            settings.FrameIndex = depth;
            settings.FrameCount = 1;

            images.Read(filepath, settings);
            MagickImage image = (MagickImage)images[0];
            image.Density = new Density(96);
            return image;
        }
        else throw new ImageReadException("Image_out_of_bounds");
    }
    catch (MagickException e)
    {
        throw new ImageReadException("Error_reading_image:_ " +
            e.Message);
    }
}
#endregion
```

Listaus 5.3: Lähdekoodin kommentointiesimerkki.

5.6 Testaus

Sovelluksen käyttöliittymää testattiin kehityksen aikana koko ajan jäsenten toimesta, jotta varmistuttiin sen toiminnasta ja laadusta. Toteutetun sovelluksen laadunvarmistus tapahtui järjestelmätestauksella, joka suoritettiin kaksi kertaa. Järjestelmätestauksella varmistuttiin siitä, että toteutettu sovellus toimi suunnitellusti sekä sovellus täytti sille asetetut toiminnalliset ja laadulliset vaatimukset. Tilaajan edustajat koekäyttivät sovelluksen prototyyppejä sekä antoivat runsaasti hyödyllistä palautetta havaitsemistaan virheistä ja tarvittavista muutoksista. Lisäksi esittivät jatkokehitysideoita kirjattavaksi vaatimusmäärittelyyn [10] ja sovellusraporttiin [7]. Suuri osa palautteesta ehdittiin tiukasta aikataulusta huolimatta toteuttaa. Tekninen ohjaaja suoritti lähdekoodin hyväksymistarkistuksen, jonka huomioiden liittyivät katselmointien kirjallisten huomioiden tavoin suurelta osin virheenkäsittelyyn. Tekninen ohjaaja koekäytti sovellusta kehityksen aikana ja antoi siitä palautetta kirjallisen muistion muodossa.

Järjestelmätestauksesta vastaava henkilö laati testaussuunnitelman ja taulukkomuotoisen testausraporttipohjan [8]. Testaussuunnitelma sisälsi testauskerroilla suoritettavat testitapaukset kuvattuina vaihe vaiheelta. Testauskerran suorittaneen jäsenen laatimassa testausraportissa kuvattiin testauskerran havainnot sekä mahdolliset havaitut virhetilanteet ja puutteet. Vastaavan jäsenen lisäksi projektiryhmän muut jäsenet olivat mukana järjestelmätestauksen suorittamisessa. Projektipäällikkö ei osallistunut käyttöliittymän kehittämiseen, joten hänen suoritti ensimmäisen testauskerran ja laati kyseisen testauskerran raportin. Vuorostaan järjestelmätestauksen vastuhenkilö suoritti toisen testauskerran ja laati vastaavasti testausraportin. Muut jäsenet osallistuivat aktiivisesti havaittujen häiriöiden selvittämiseen ja korjaamiseen.

Projektissa oli tarkoitus, että ohjelmoija suorittaa tarvittavat yksikkötestaukset kirjoittamalleen lähdekoodille. Yksikkötestausta varten luotiin erillinen testausprojekti versiohallintaan vastaavasti kuin käyttöliittymälle ja luokkakirjastolle. Testausprojektilla oli tarkoitus testata CoreLibraryn toiminnallisuus, mutta projektin lopussa yksikään testimetodeista ei testannut käytössä olevaa ohjelmakoodia. Testausprojekti päätettiin siten ylimääräisenä poistaa versiohallinnasta.

Käyttöliittymää ei voinut ajaa Linux-koneella, joten testausprojektiä oli tarkoitus käyttää TDD-henkisesti (Test Driven Development) kehitettävän ohjelmaosan toiminnan määrittelyssä. Testitapauksia olikin kertynyt jonkin verran varmistamaan, että ohjelma käyttäytyy edelleen suunnitellusti. Ajattelutavasta seurasi kuitenkin kohtalokas virhe, sillä etukäteen kirjoitettuja apufunktioita (vektorilaskuihin ja tiedostojen käsittelyyn) tai ylipäänsä testausprojektiä ei koskaan tarvittu. Tätä periaatetta kutsutaan XP-ohjelmoinnissa (Extreme Programming) lyhenteellä YAGNI (You aren't gonna need it). Toisin sanoen luokkakirjaston ke-

hittäminen erillään käyttöliittymästä ja tiedostonluvusta oli äärimmäisen lyhytkatseista toimintaa, jolla oli kauaskantoiset seuraukset. Toisaalta siinä tilanteessa ei voinut juuri muutaakaan, sillä eihän ohjelmaa voinut suorittaa Linux-koneella.

5.7 Versiohallinta ja -numerointi

Sovelluksen lähdekoodin versiohallintaan käytettiin Git-versiohallintaohjelmistoa. Lähdekoodi sijoitettiin yliopiston GitLab-palveluun, jossa se oli projektin ajan koko projektiorganisaation saatavilla. Dokumenteille ei ollut erillistä versiohallintaa, mutta projektissa laaditut dokumentit aina muokattaessa numeroitiin uudella versionumerolla. Dokumentit sijaitsivat Overleaf-palvelussa projektiryhmän yhteisessä työtilassa, joten dokumentin viimeisin tekstiversio oli aina saatavilla. Projektissa laadittujen dokumenttien uusimpia versioita julkistettiin saataville projektin WWW-sivuilla.

Julkistettavissa dokumenteissa käytettiin kolmiportaista numerointia (i.j.k). Projektiryhmän sisäisiä versioita numeroitiin pienimmällä merkitsevällä numerolla paikassa k. Projektionnisaatiolle julkistettavat versiot numeroitiin toisella merkitsevällä numerolla paikassa j. Projektionnisaation hyväksymän dokumentin versio numeroitiin paikassa i. Esimerkiksi projektisopimusta muokattiin ensimmäisen hyväksymisen jälkeen vielä lisenssien osalta, jolloin lopullisen allekirjoitettavan sopimuksen versionumeroksi muodostui 2.0.0.

Tiedostojen nimeämisen kohdalla versionumeroinnin noudattaminen tuotti vaikeuksia, sillä esimerkiksi suunnitteludokumenteista jouduttiin tekemään identtiset kopiot, jos ne haluttiin julkistaa. Toisaalta jos julkistettavat dokumentit olivat kokonaan erillisiä (kuten käyttöliittymä- ja arkkitehtuurisuunnitelman muutosehdotuksen yhteydessä) niin järjestysnumero menetti merkityksensä. Esimerkiksi 3D-tilakarttaa suunnitellessa versionumerointi osoitautui osittain riittämättömäksi, sillä lopulta päätettiin olla seuraamatta uutta kehityshaaraa tai se siirrettiin jatkekehitykseen. Tällaisessa tilanteessa olisi suotavaa sallia julkistettavan myös ryhmän sisäisiä versioita, kuten 0.1.1 tai 0.1.2. Projektionnisaatio voi silloin valita kumman version kehittämistä jatketaan versiona 0.2.0.

Overleaf-palvelua olisi kannattanut hyödyntää myös pöytäkirjojen laatimisessa, sillä todennäköisesti puheenjohtaja ehdotti vähintäänkin pieniä muutoksia pöytäkirjaan. Kirjoitusvirheet olisivat olleet helposti korjattavissa välittömästi oikoluvun yhteydessä ja suuremmat sisällön muutokset olisi voinut kirjata suoraan joko kommentteihin tai erillisellä muokkaustyökalulla. Harvemmin sihteerin kirjaamia projektipalaverien pöytäkirjojen hahmotelmia tarvitsee säilyttää, joten muistiot olisi kannattanut kirjoittaa suoraan tex-tiedostoon, joka käyttäisi projektiryhmän yhteistä cls-dokumenttiluokkaa pöytäkirjoille. Muistioiden rakenne olisi kannattanut vuorostaan kopioida esityslistasta ennen palaveria, jotta dokumentti-

pohjan muotoilussa olisi säästetty aikaa palaverin jälkeen.

5.8 Katselmoinnit ja tulosten hyväksyminen

Projektin aikana sovelluksen lähdekoodi katselmoitiin teknisen ohjaajan toimesta kaksi kertaa. Ensimmäinen katselmointi pidettiin 21.4 ja toinen 3.6. Ennen molempia katselmointeja tekninen ohjaaja laati kirjalliset kommentit, jotka hän välitti sähköpostitse koko projektiorganisaatiolle ennen katselmointitilaisuutta. Katselmointeihin pyrkivät osallistumaan koko projektiorganisaatio, kuten muihinkin projektin yhteisiin tapahtumiin. Katselmoinneissa kiinnitettiin huomiota mm. ammattimaisen lähdekoodin muotoiluun sekä edistyneisiin C#-ohjelmointikielen ominaisuuksiin, kuten säieturvallisuuteen ja metodien nullable-tyyppisten paluuarvojen merkitsemiseen. Katselmointien palautteet olivat täsmällisen lyhyitä, mutta sitäkin opettavaisempia. Palautteen kuuleminen etukäteen vähensi merkittävästi jäsenten muutosvastarintaa. Edistyneiden C#-kielen ominaisuuksien ansiosta teknisen ohjaajan palautteet sisällytettiin tilaajan edustajien laatiman aihekuvauksen tavoin projektikansioon. Erinomaisen kirjallisen palautteen ansiosta projektiryhmän ei tarvinnut laatia katselmoinneista erillistä muistiota.

Projektin lopussa laadittavat dokumentit sovittiin hyväksyttävän ensin vastaavalla ohjaajalle ja vasta sitten tilaajan edustajilla. Menettely johtui projektin viivästyisestä, jolloin tilaajan edustajat jäivät lomalle ennen vastaavaa ohjaa ja palasivat lomaltaan ennen vastaavaa ohjaajaa. Vastaava ohjaaja ja tilaajan edustajat hyväksyivät keskeiset dokumentit, joita olivat projektisuunnitelma, vaatimusmäärittely, testaussuunnitelma, sovellusraportti ja projektiraportti. Keskeiset dokumentit hyväksyttiin sähköpostitse vastaavan ohjaajan, tilaajan edustajan ja projektipäällikön toimesta. Tilaajan edustajille riitti lähdekoodin osalta teknisen ohjaajan hyväksyntä sähköpostitse. Projektiryhmän tuloksista ainoastaan projektisopimus allekirjoitettiin yliopiston koronarajoituksista johtuen.

5.9 Tulosten koostaminen ja toimittaminen

Projektiryhmä kokosi luvussa 3.5 kuvatut projektin tulokset luvussa 5.4 kuvatun hakemistorakenteen mukaisesti. Informaatioteknologian tiedekunnalle toimitettiin tuloksista koostettu projektikansio ja kaksi CD-levyä, joista toinen oli arkistointia varten. CD-levy koostettiin vasta, kun kaikki projektin tulokset oli hyväksytty ja projektikansio sijoitettiin projektitilan kokoushuoneessa olevaan kirjahyllyyn. Tulokset toimitettiin tilaajalle pakattuna zip-tiedostona projektin WWW-sivuston kautta.

Lähdekoodien osalta sovittiin, ettei niitä tulosteta projektikansioon. Prototyyppien tai järjestelmätestauksen testidataa ei tulostettu projektikansioon. Suurin osa näistä on väliaikais-tiedostoja, mutta jotkin dokumentit saattaisivat osoittautua hyödylliseksi jatkokehityksessä. Sen takia tex-tiedostot ja niiden cls-dokumenttiluokat sisällytetään CD:lle ja tulospakettiin.

6 Roolit, vastualueet ja tehtävät

Luvussa kuvataan projektiryhmän jäsenten roolit, vastualueet ja tehtävät. Lisäksi luvussa esitetään ryhmän jäsenten arvioidut ja toteutuneet työtunnit tehtäväkokonaisuuksittain. Tehtäviin tarvittavia työtunteja ei osattu arvioida riittävän tarkasti, joten projektinhallinnan toteutuneet työtunnit olivat suunniteltua suuremmat ja toteutuksen suunniteltua pienemmät. Projektihallinnan tehtäväkokonaisuuteen kirjattiin siihen kuulumattomia tehtäviä, joten toteutuneita työtunteja ei voitu luotettavasti verrata suunnitelmaan.

Roolit vastasivat suunniteltua ja jäsenet omaksuivat projektin aikana selkeät vastualueet, jotka liittyivät projektin tuloksiin. Ryhmän jäsenet suunnittelivat itsenäisesti tehtäviään ja ajankäyttöään, joten projektipäällikön toimeen kuului nimellisesti vain projektin suunnittelu ja raportointi. Projekti toteutettiin kokonaan etänä, mikä edellytti itsenäistä vastuunottoa tehtävien tuloksista. Tuloksiin liittyvät vastualueet noudattivat projektiorganisaatiolla hyväksyttävien tulosten osalta melko tarkasti suunniteltua, mutta suunnitelmassa olisi kannattanut esitellä yksi vastuuhenkilö olemaan päävastuussa yksikkötesteistä. Yksikkötestausta suunniteltiin ja toteutettiin projektin alkupuolella, mutta yksikkötestausta ei juurikaan tehty enää toteutusvaiheessa.

6.1 Roolit ja vastualueet

Projektipäällikkönä toimi Harri Linna ja varapäällikkönä Jere Pakkanen. Projektipäällikön vastuulla oli projektin suunnittelu, hallinta ja raportointi. Lisäksi projektipäällikön vastuulla oli myös tehtävien jakaminen projektiryhmän jäsenille ja säännöllinen projektin edistymisen valvominen. Varapäällikkö hoiti palaverissa puheenjohtajan tehtäviä sekä huolehti palaverien esityslistojen julkistamisesta.

Taulukossa 6.1 kuvataan jäsenten toteutuneet projektin tulosten vastualueet. Vastuuhenkilö vastasi pääosin kyseisen tuloksen saattamisesta valmiiksi, mutta vastuuhenkilö ei välttämättä toteuttanut kyseistä vastuualuetta yksin. Vastuuhenkilö tiedotti projektiorganisaatiota vastuullaan olevien tulosten valmistumisesta.

Taulukossa 6.2 kuvataan jäsenten toteutuneet ominaisuuskokonaisuuksien vastualueet. Vastuuhenkilö vastasi pääosin kyseisen tehtäväkokonaisuuden saattamisesta valmiiksi, mutta vastuuhenkilö ei välttämättä toteuttanut kyseistä vastuualuetta yksin. Vastuuhenkilö tiedotti projektiorganisaatiota vastuullaan olevien ohjelmaosien valmistumisesta ja niiden esittelystä palaverissa. Projektisuunnitelmassa keskityttiin dokumenttien vastuualueisiin ja sovellusosoiden vastualueet realisoituivat projektin edetessä vaatimusmäärittelyn tarkentumisen myötä. Taulukko on otettu valmiista vaatimusmäärittelystä [10], mutta teknisten

Vastuualueen kuvaus	Vastuuhenkilö
Ajankäyttöraportti	Harri Linna
Arkkitehtuurisuunnitelma	Harri Linna
Esittelymateriaali	Kaikki jäsenet
Esityslistat	Harri Linna, Jere Pakkanen
Itsearviointit	Kaikki jäsenet
Kehitysympäristön pystytysohjeet	Riikka Vilavaara
Käyttöliittymäsuunnitelma	Harri Linna
Käyttöohjeet	Iiro Iivanainen
Luokkadokumentaatio	Kaikki jäsenet
Luokkasuunnitelma	Kaikki jäsenet
Lähdekoodi	Kaikki jäsenet
Palaverien dokumentit	Iiro Iivanainen, Riikka Vilavaara
Projektiraportti	Harri Linna
Projektisuunnitelma	Harri Linna
Sovellusraportti	Riikka Vilavaara
Sähköpostiarkistot	Harri Linna
Testausraportti	Iiro Iivanainen, Harri Linna
Testaussuunnitelma	Iiro Iivanainen
Tilakatsaukset	Harri Linna
Vaatimusmäärittely	Jere Pakkanen
Valmisohjelma	Jere Pakkanen
Verkkoesite	Riikka Vilavaara

Taulukko 6.1: Dokumenttien toteutuneet vastualueet.

vaatimusten osalta rajoituttiin Windows-ympäristöön ja raportointityökalu siirrettiin kokonaisuudessaan jatkokehitykseen. Tehtäväkokonaisuudet tiivistävät sovellusosioiden vastualueiden jakoa.

Käyttöliittymän toteutuksen työtunteihin kirjattiin käyttöliittymäsuunnitelman mukaisten näkymien toteuttamisen. Vilavaara oli päävastuussa pääikkunan ikkunoinnista ja MVVM-arkkitehtuurin mukaisesta ohjelmoinnista. Lisäksi pääikkunan karttanäkymä ja karttakontrolleri kuuluivat hänen tehtäviinsä. Iivanainen oli päävastuussa liitoskohtia visualisoivan liukusäätimen ja dataliitosten hallintaan käytettävän listanäkymän toteuttamisesta. Lisäksi hänen tehtäviinsä kuuluivat metatietojen ja käyttäjäasetusten tekeminen. Pakkanen oli pää-

Vastuualueen kuvaus	Vastuuhenkilö
Karttatiedostojen lukemisen hallinta	Jere Pakkanen
Karttatiedostojen näyttämisen hallinta	Riikka Vilavaara
Liitetyn datan lukemisen hallinta	Iiro Iivanainen
Liitetyn datan näyttämisen hallinta	Iiro Iivanainen
Metatietojen hallinta	Iiro Iivanainen
Karttatiedoston ja liitetyn datan hallinta	Riikka Vilavaara
Sovelluksen tilan tallentamisen ja lataamisen hallinta	Jere Pakkanen
Sovelluksen käyttöliittymän hallinta	Riikka Vilavaara
Ohjeiden hallinta	Iiro Iivanainen
Rajoitteiden hallinta	Jere Pakkanen

Taulukko 6.2: Sovellusosioitten toteutuneet vastuualueet.

vastuussa erilaisten karttatiedostojen käsittelystä, sovelluksen tilan tallentamisesta, säikeistämistä ja poikkeuskäsittelystä. Linna suunnitteli sovellusta ja toteutti tilakartan prototyypin. Yksikkötestausta suunniteltiin tehtävän TDD-henkisesti koko toteutusvaiheen ajan, mutta ohjelmistossa ei sinänsä ollut testattavaa. Esimerkiksi tiedoston lukemista oli helpompi arvioida silmämääräisesti kuin yksittäisiä pikseleitä vertaamalla. Muita toteutusvaiheen jäsenten yhteisiä tehtäviä olivat virheenetsintä ja koodikatselmointien palautteen pohjalta tehtävä lähdekoodin refaktorointi.

Vastuualueet tulosten osalta toteutuivat suunnitellusti. Suunniteltu vastuuhenkilö vastasi omasta vastuualueestaan projektin aikana. Ryhmän jäsenet osallistuivat joustavasti toistensa vastuualueisiin sisältyneiden tehtävien suorittamiseen sekä dokumenttien että lähdekooditiedostojen osalta.

6.2 Tehtäväkokonaisuudet ja työmäärät

Ryhmän suunniteltiin käyttävän sovellusprojektiin yhteensä 1600 tuntia, mutta toteutunut tuntimäärä oli oheiskurseja lukuun ottamatta 1635 tuntia. Kokonaisuudessaan työtunteja käytettiin suunniteltua enemmän. Suunnitelmassa palaverit sisältyivät projektinhallinnan tehtäväkokonaisuuteen. Esitutkimus katsottiin kuuluvan projektin suunnitteluvaiheeseen. Suunnitelmassa keskityttiin Sovellusprojektin työtunteihin ja siksi oheiskurseja ei mielletty Sovellusprojektin tehtäväkokonaisuudeksi. Taulukossa 6.3 esitetään eri tehtäväkokonai-

suuksien suunnitellut ja toteutuneet tuntimäärät.

Tehtäväkokonaisuus	Suunniteltu	Toteutunut	Erotus
Vaatimusmäärittely	48	18	-30
Suunnittelu	400	310	-90
Toteutus	816	566	-250
Testaus	96	63	-33
Projektinhallinta	240	678	438
Sovellusprojekti	1600	1635	35

Taulukko 6.3: Projektin tehtäväkokonaisuuksien työtunnit.

Tehtävien työmäärät suunniteltiin aikaisempien sovellusprojektien toteutuneiden tuntien pohjalta. Jäsenten työnjako oli tasapuolinen ja vaiheiden vaatimaa työmäärää kyettiin ennakkoimaan edellisten ryhmien ajankäyttöraporttien perusteella. Näistä ajankäyttöraporteista saatiin suuntaa antavat prosentuaaliset osuudet kokonaistyömäärän jakautumisesta vaiheittain, mitkä sitten suhteutettiin tämän projektin arvioituun kokonaistyömäärään. Työmääräarviot määräytyivät kokonaistyömäärästä kuvan 6.1 mukaisesti, kun kokonaistyömäärän arvio laskettiin kaavalla

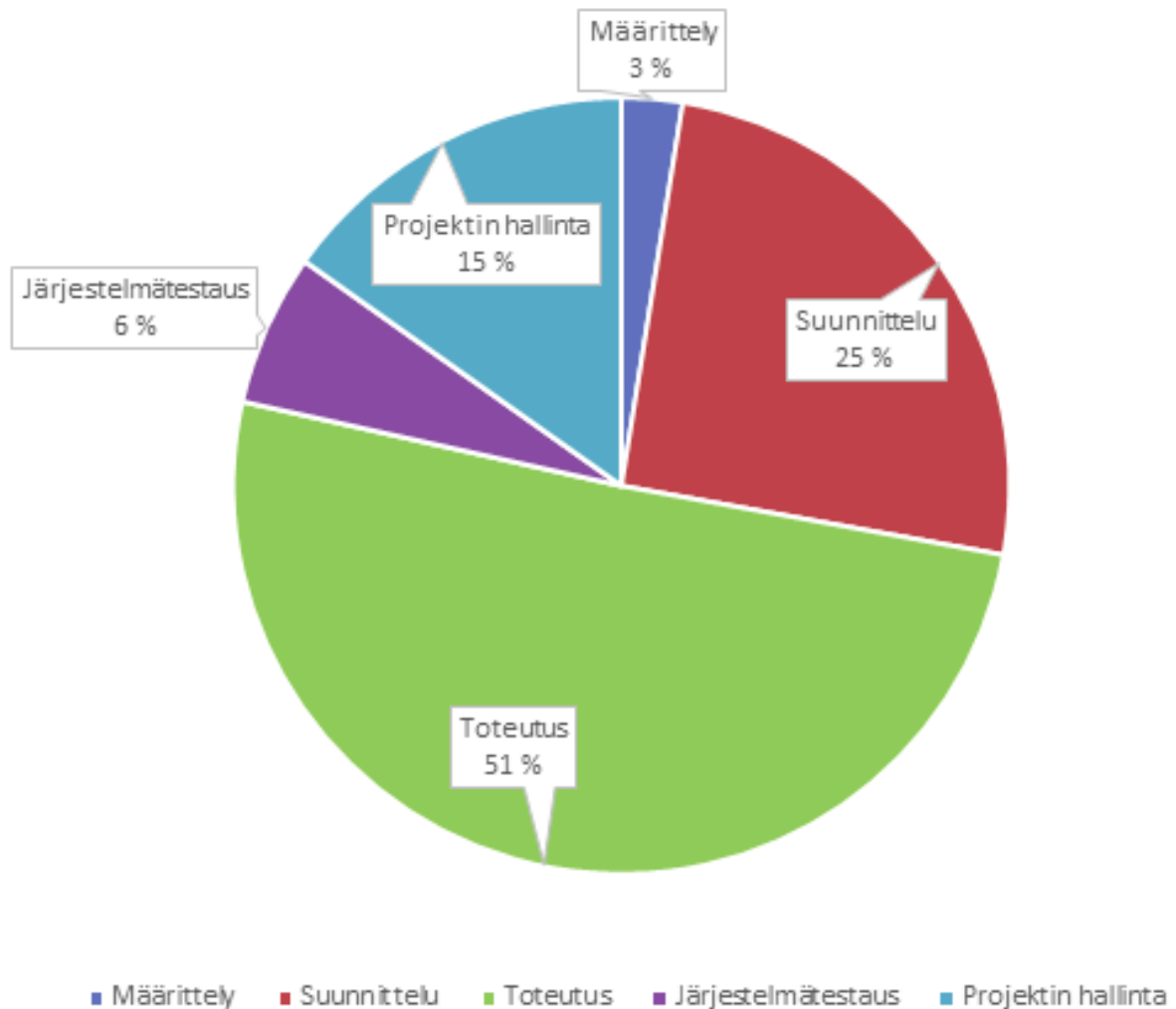
$$26,67 \frac{T}{OH} \cdot 15 O \cdot 4 H = 400,05 \frac{T}{H} \cdot 4 H = 1600,2 T, \text{ missä}$$

- T = Työtunnit,
- O = Opintopisteet ja
- H = Henkilömäärä.

Arviointimenetelmä erosi merkittävästi projektisuunnitelmaan kirjatusta kolmen pisteen työmääräarvioinnista, jossa projektiryhmän jäsenet olisivat antaneet tehtäville kolme eri työmääräarviota: optimistinen, pessimistinen ja todennäköinen. Lopullinen työmääräarvio olisi saatu näiden kolmen painotetusta keskiarvosta. Nykyinen arviointi perustuu valistuneiden arvausten sijaan aiempien sovellusprojektien dokumentoituun tietoon.

Dokumenttien laatimiseen käytetyt työtunnit kirjattiin virheellisesti projektinhallinnan tehtäväkokonaisuuteen, vaikka dokumentit olivat eri tehtäväkokonaisuuksien tuloksia. Ainoastaan projektisuunnitelman ja -raportin työmäärät olisi kannattanut merkitä projektinhallintaan ja muut dokumentit kyseisten tehtäväkokonaisuuksien alle.

Kaikki tapaamiset mukaan lukien ryhmän sisäiset tapaamiset kirjattiin virheellisesti palaverien tehtäväkokonaisuuteen. Tämä aiheutti vääristymää projektinhallinnan työmäärissä, koska suunnitelmassa palaverit sisältyivät projektinhallinnan tehtäväkokonaisuuteen. Ainoastaan projektipalaverit olisi kannattanut merkitä projektinhallintaan ja muut suunnittelupalaverit kyseisen tehtäväkokonaisuuden alle.

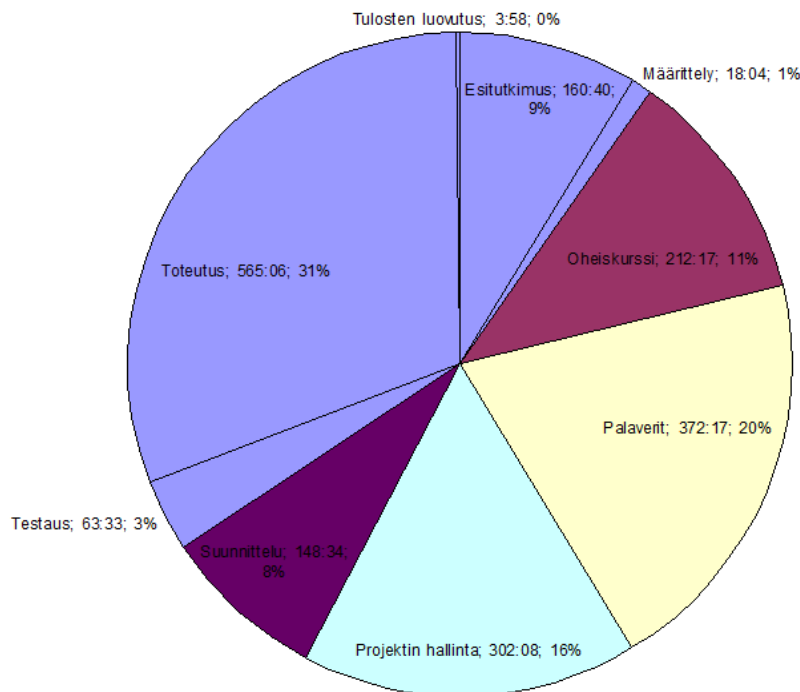


Kuva 6.1: Suunnitellut työmäärät tehtäväkokonaisuuksittain.

Suunniteltujen ja toteutuneiden työtuntien välinen erotus selittyy suurimmaksi osaksi näillä työtuntien kirjaamisen käytänteillä. Projektinhallinnan suuri erotus selittyy edelleen dokumenttien laatimiseen käytetyillä työtunneilla. Kuvassa 6.2 esitetään projektin toteutuneet työmäärät vaiheittain, jotka erosivat suunnitelmasta.

Esimerkiksi jos vaatimusmäärittelyn (määrittely), testaussuunnitelman (testaus) ja sovellusraportin (toteutus) työtunnit siirrettäisiin niitä vastaaviin tehtäväkokonaisuuksiin, niin projektinhallinnan osuuden voisi kuvitella laskevan. Vastaavasti muiden tehtäväkokonaisuusien osuudet kasvaisivat, jolloin työmäärien toteuma lähestyisi suunniteltua. Luotettavia johtopäätöksiä toteuman ja suunnitelman työtuntien eroista ei kuitenkaan voi tehdä.

Ryhmän jäsenet merkitsivät tunnollisesti projektiin käyttämiään työtunteja. Ryhmän olisi kuitenkin kannattanut heti projektin alussa sopia työtuntien kirjauskäytännöistä huomatt-



Kuva 6.2: Ryhmän toteutuneet työmäärät tehtäväkokonaisuuksittain.

tavasti tarkemmin. Kirjauskäytänteisiin olisi voinut vielä vaikuttaa, jos projektisuunnitelma olisi valmistunut maaliskuussa. Joka tapauksessa tuntien kirjaaminen suunnitelman mukaisille tehtäväkokonaisuuksille, olisi tehnyt työtuntien vertailusta projektipäällikölle helpompaa.

6.3 Projektinhallinnan työmäärät

Projektinhallintaan oli suunniteltu 240 tuntia ja siihen käytettiin 678 tuntia. Kaikkia tulosten viimeistelyyn ja kokoamiseen käytettyjä tunteja ei voitu kirjata taulukkoon 6.4, sillä kokoamista tehtiin vielä projektiraportin kirjoittamisen jälkeen. Projektinhallinnan työtunteihin kirjattiin pääasiassa dokumenttien laatimiseen käytettyjä työtunteja.

Projektipäällikkö otti tehtävän vastaan vastahakoisesti ja suunnitteli tekevänsä projektinhallintaa muiden tehtäviensä ohella. Projektin jäsenet kokoontuivat kaksi tai kolme kertaa viikossa keskustelemaan projektin läpiviennistä, mikä selittää arvioitua suuremmat palaverien työtunnit. Poikkeavista järjestelyistä johtuen ryhmän jäsenet tapasivat vain kerran projektisopimuksia allekirjoittaessaan. Tiedottaminen tapahtui suurimmaksi osaksi projektiryhmän

yhteisissä tapaamisissa.

Projektinhallinta on yleensä 15–20 prosenttia kokonaistyömäärästä, joten projektipäällikön olisi kannattanut keskittyä ensisijaisesti projektinhallinnan tehtäviin. Muut ryhmän jäsenet eivät tästä huomauttaneet, joten ehkäpä yhteisvastuu koettiin toimivaksi. Taulukossa 6.4 kuvataan projektinhallinnan työmäärät.

Tehtävä	RV	JP	II	HL	Yhteensä
kirjoitus -ja ulkoasu	5	0	29	0	34
muut tehtävät	11	3	9	9	32
raportointi	66	27	2	52	147
seuranta ja hallinta	0	4	3	6	13
sopimukset	0	1	0	22	23
suunnittelu	0	0	0	8	8
tiedotus	6	1	0	2	9
toteutus	0	0	4	0	4
tukitehtävät	0	0	3	10	13
tutustuminen	2	3	7	0	12
valmisteltu ja tutustuminen	1	0	0	0	1
Projektinhallinta	91	39	57	109	296

Taulukko 6.4: Projektinhallinnan toteutuneet työtunnit.

Suunnitelmassa palaverit sisältyivät projektinhallintaan, mutta niihin käytettiin 372 tuntia. Palaverien suuret työtunnit johtuvat siitä, että ryhmän jäsenet kirjasivat projektipalavereiksi myös ryhmän tapaamiset, joita tehtiin ajoittain jopa kolmesti viikossa. Ryhmän sisäisiä tapaamisia merkittiin nimikkeellä muut tehtävät. Toteutuneet työtunnit olivat siis suunnitellen kolminkertaiset edellisvuosien projekteihin verrattuna, koska Groundhog-projekti toteutettiin kokonaan etänä.

Palaverien työtunteja olisi kannattanut arvioida edellisvuosien pöytäkirjojen lukumäärän ja palaverien keskimääräisen keston perusteella. Koko ryhmä käytti palavereihin kahdeksan tuntia helmikuussa joka viikko ja maaliskuusta kesäkuulle joka toinen viikko. Lisäksi niihin liittyi esityslistojen, tilakatsausten ja pöytäkirjojen valmistelua. Ryhmän tapaamisia pidettiin kaksi tai kolme kertaa viikossa, mikä vastaa 16–24 henkilötyötuntia viikoittain. Lisäksi projektin viivästyessä valmiiksi korkeat tuntimäärät kasvoivat entisestään.

Palaverit vaikuttivat rutiinitehtäviltä, mutta niihin käytettävien työtuntien aliarviointi varmaankin johti aikaresurssien loppumiseen projektin toteutusvaiheessa, jolloin toteutukseen varatut työtunnit jäivät merkittävästi alle suunnitellun. Palaverien vaatiman työmäärän väheksyminen oli siten merkittävä virhearvio projektin läpiviennin kannalta. Esimerkiksi pa-

lavereihin käytettiin enemmän aikaa kuin projektinhallintaan. Taulukossa 6.5 esitetään palaverien toteutuneet työtunnit tehtävittäin.

Tehtävä	RV	JP	II	HL	Yhteensä
katselmoinnit	1	1	1	2	5
muut tehtävät	55	66	52	48	221
raportointi	2	8	6	10	26
seuranta ja hallinta	0	0	0	3	3
sopimukset	0	0	0	1	1
suunnittelu	9	11	7	4	31
tiedotus	0	0	0	2	2
tukitehtävät	18	7	9	15	49
tutustuminen	2	2	0	2	6
valmistelu ja tutustuminen	4	7	4	8	23
Palaverit	91	102	79	95	367

Taulukko 6.5: Palaverien toteutuneet työtunnit.

6.4 Vaatimusmäärittelyn työmäärät

Vaatimusmäärittelyyn arvioitiin 48 tuntia, mutta käytettiin 18 tuntia. Vaatimusmäärittelyn suunnittelu ja raportointi kirjattiin projektinhallinnan tehtäväkokonaisuuteen. Muutoshistorian perusteella dokumenttia päivitettiin säännöllisesti lähes viikoittain.

Vaatimusmäärittelyn vastuhenkilö toimi puheenjohtajana useimmissa palavereissa vastaten useimpien pöytäkirjojen tarkastamisesta. Pöytäkirjojen tarkastamisen yhteydessä havaitut sovellukseen kehitettävät tiedot ja toiminnot kirjattiin vaatimusmäärittelyyn, jolloin vaatimusmäärittelyn työtunnit kirjattiin samaan tehtäväkokonaisuuteen kuin pöytäkirjojen tarkastaminen. Taulukossa 6.6 on määrittelyvaiheen toteutuneet työtunnit tehtävittäin.

Tehtävä	RV	JP	II	HL	Yhteensä
kirjoitus- ja ulkoasu	0	0	0	4	4
raportointi	0	7	0	0	7
seuranta ja hallinta	0	0	0	7	7
Määrittely	0	7	0	11	18

Taulukko 6.6: Vaatimusmäärittelyn toteutuneet työtunnit.

6.5 Suunnittelun työmäärät

Sovelluksen suunnitteluun arvioitiin 400 tuntia, mutta käytettiin 148 tuntia. Suunnitelmasa esitutkimus miellettiin yhdeksi suunnitteluvaiheen tehtäväksi. Toteutusteknologioihin ja ohjelmakirjastoihin tutustuminen liittyy nimittäin erottamattomasti sovelluksen suunnitteluun. Suunnitteluun käytettiin 310 työtuntia esitutkimus mukaan lukien. Suunnitteluun kuulumattomia tehtäviä kirjattiin projektinhallinnan ja vaatimusmäärittelyn dokumenttien osalta.

Suunnittelun työtunnit kohdistuivat karttatiedostojen lukemisen ja käyttöliittymän karttanäkymän toteutusratkaisuihin. Sovellussuunnitelmien laatimiseen käytettiin arvioitua vähemmän aikaa, sillä suunnitelmat olivat toteutuskelpoisia jo viikolla 10. Toteutusvaiheeseen voitiin siirtyä viikolla 11. Sovellussuunnitelmia päivitettiin seuraavan kerran vasta viimeistelyvaiheessa jatkokehitysideoiden dokumentoimista varten. Taulukossa 6.7 esitetään suunnitteluvaiheen työtunnit tehtävittäin.

Tehtävä	RV	JP	II	HL	Yhteensä
kirjoitus- ja ulkoasu	0	0	0	2	2
käyttöliittymä	7	4	3	22	36
muut tehtävät	4	0	0	12	16
rajapinnat	0	0	0	15	15
raportointi	0	19	0	0	19
seuranta ja hallinta	0	0	0	4	4
suunnittelu	0	10	0	31	41
toteutus	0	3	0	0	3
tukitehtävät	0	0	0	9	9
Suunnittelu	11	36	3	95	145

Taulukko 6.7: Suunnittelun toteutuneet työtunnit.

Esitutkimus suunniteltiin kuuluvan suunnitteluun, mutta siihen käytettiin 160 tuntia. Monet työkalut olivat tuttuja ryhmän jäsenille jo entuudestaan. Työkaluihin ja laitteisiin tutustumista tehtiin myös sovelluskehityksen aikana, joten niitä lienee kirjattu toteutukseen. Aihealueeseen tutustuttiin myös paljon palavereissa tilaajan edustajien kanssa keskustellessa ja osin myös ryhmätapaamisissa. Työtunteja oli hankala arvioida, koska ei ollut selkeää käsitystä siitä, kuinka paljon uusia työkaluja projektin jäsenet joutuvat opettelemaan. Taulukossa 6.8 esitetään esitutkimuksen toteutuneet työtunnit tehtävittäin.

Esitutkimukseen sisältyi aihealueen tuntemuksen hankkimista tutustumalla Fiji- ja ImageJ-ohjelmiin sekä pi2-kirjaston dokumentaatioon. Lisäksi esitutkimukseen sisältyi kehitysympäristön Visual Studio asentaminen ja WPF-projektin käytänteiden kertaaminen. Merkittä-

vin osa työtunneista liittyi kuitenkin MVVM-arkkitehtuurin ja WPF-projektin tutkimiseen sekä ulkoisten kirjastojen kartoittamiseen. Esitutkimuksen aikana harkittiin eri ohjelmointikieliä sekä kymmenittäin erilaisia ohjelmakirjastoja, joiden lisenssiehtojen yhteensovittaminen lisäsi työmäärää. Monet vapaan ohjelmistolisenssin kirjastot kielsivät kaupallisen käytön tai olivat muuten tarttuvia GPL-lisenssejä.

Jälkeenpäin ajatellen esitutkimuksen tuntimäärät olisi kannattanut sisällyttää suunnitelmaan omana tehtäväkokonaisuutenaan. Projektin käynnistämiseen liittyi paljon sellaisia käytännön asioita, jotka eivät varsinaisesti liittyneet vaatimusmäärittelyyn eivätkä projektin tai sovelluksen suunnitteluun, vaan enemmänkin tietopohjan ja aihealueen ymmärryksen hankkimiseen. Projektin alussa sovellettiin ketterää ohjelmistokehitystä Scrumin muodossa. Projektissa siirryttiin vasta myöhemmin takaisinkytkettyyn vesiputousmalliin. Esitutkimuksen työtunnit olisi kannattanut kirjata suunnitteluvaiheen tehtäväksi.

Tehtävä	RV	JP	II	HL	Yhteensä
esitutkimus	0	0	1	0	1
kirjoitus- ja ulkoasu	0	0	1	0	1
käyttöliittymä	17	1	17	1	36
muut tehtävät	4	0	5	0	9
suunnittelu	0	15	2	1	18
toteutus	0	2	13	2	17
tukitehtävät	1	0	8	8	17
tutustuminen	13	4	13	9	39
valmistelu ja tutustuminen	1	5	0	12	18
Esitutkimus	36	27	60	33	156

Taulukko 6.8: Esitutkimuksen toteutuneet työtunnit.

6.6 Toteutuksen työmäärät

Toteutukseen suunniteltiin 816 tuntia, mutta käytettiin 566 tuntia. Huomattavasti suunniteltua vähempi työmäärä selittyy viikoittaisten palaverien ja ryhmän sisäisten tapaamisten suurella määrällä. Kaksi tapaamista viikossa osoittautui etäaikana toimivaksi järjestelyksi. Sovellukseen verrattuna muutaman tunnin tapaamiset saattoivat vaikuttaa rutiininomaisilta tehtäviltä, mutta ryhmän jäsenet joutuivat sovittamaan ajankäyttöään puolenpäivän aikaan järjestettäviin tapaamisiin. Jäsenten kertoman mukaan saatiin enemmän aikaa sellaisina päivinä, jolloin palavereja ei järjestetty. Ryhmän jäsenet tunnustivatkin tarpeen pitää välillä taukoa ryhmän sisäisistä tapaamisista, jotta voisivat keskittyä paremmin vastualueitensa itsenäisiin ohjelmointitehtäviin. Yksi tapaaminen viikossa koettiin riittämättömäksi ja

kolmas tapaaminen viikossa koettiin ylimääräisenä. Sopivat ajankohdat olivatkin alkuvii-
kosta maanantaina tai tiistaina sekä toisen kerran keskiviikkona tai torstaina. Viikonlop-
puisin ei edellytetty ohjelmointia, mutta monet varmasti käyttivät yhtäjaksoisen työrauhan
hyväkseen. Taulukossa 6.9 esitetään toteutuksen toteutuneet työtunnit tehtävittäin.

Tehtävä	RV	JP	II	HL	Yhteensä
kirjoitus- ja ulkoasu	0	0	1	0	1
käyttöliittymä	122	0	47	0	169
muut tehtävät	1	0	0	0	1
rajapinnat	0	0	0	6	6
suunnittelu	0	3	0	1	4
toteutus	19	186	79	88	372
tukitehtävät	3	0	0	4	7
valmistelu ja tutustuminen	0	0	0	2	2
Toteutus	145	189	127	101	562

Taulukko 6.9: Toteutuksen toteutuneet työtunnit.

6.7 Testauksen työmäärät

Järjestelmätestaukseen suunniteltiin 96 tuntia, mutta käytettiin 63 tuntia. Ero johtuu siitä,
että kolmatta testauksetta ei ollut tarpeen suorittaa. Testauskerroilla havaitut häiriöt liit-
tyivät suurimmaksi osaksi poikkeuskäsittelyyn, jolloin virheiden korjaaminen oli suoravii-
vaista. Taulukossa 6.10 esitetään järjestelmätestauksen toteutuneet työtunnit tehtävittäin.

Tehtävä	RV	JP	II	HL	Yhteensä
kirjoitus- ja ulkoasu	0	0	34	0	34
käyttöliittymä	0	0	1	13	14
muut tehtävät	2	0	0	0	2
raportointi	0	0	0	9	9
toteutus	0	0	3	0	3
Testaus	2	0	38	22	62

Taulukko 6.10: Testauksen toteutuneet työtunnit.

Pakkanen ja Vilavaara merkitsivät testauksella havaittujen häiriöiden tai puutteiden kor-
jaamiseen käyttämänsä työtunnit muihin tehtäväkokonaisuuksiin, mikä selittää osaltaan
testauksen vähäiset kokonaistyötunnit. Testausvaihe kesti noin kaksi viikkoa, joten työtun-
nit olivat todellisuudessa suuremmat.

6.8 Resurssit ja työmäärät

Tehtävät pyrittiin jakamaan tasaisesti ryhmän jäsenten kesken. Ryhmän jäsenten toteutuneet työtunnit noudattavat yhteistä tavoitetta hyvin. Projektisuunnitelmassa arvioitiin, että kaikki tekevät vähintään 400 tuntia. Ryhmän jäsenet kirjasivat työtunteja ajankäytönseurantaan vielä projektiraportin laatimisen jälkeen, joten taulukon projektiraportin sisältämät tuntimäärät ovat vain suuntaa antavia. Ajankäytönseurannan [3] mukaan kaikki ryhmäläiset saavuttivat 400 tunnin tavoitteen, kun oheiskurssien työtunnit poistetaan laskuista. Taulukko 6.11 sisältää jäsenten toteutuneet työtunnit oheiskursseja lukuun ottamatta.

Resurssi	Suunniteltu	Toteutunut	Erotus
Iiro Iivanainen	400	368	-32
Harri Linna	400	477	77
Jere Pakkanen	400	406	6
Riikka Vilavaara	400	384	-16
Yhteensä	1600	1635	35

Taulukko 6.11: Projektin resurssien työtunnit.

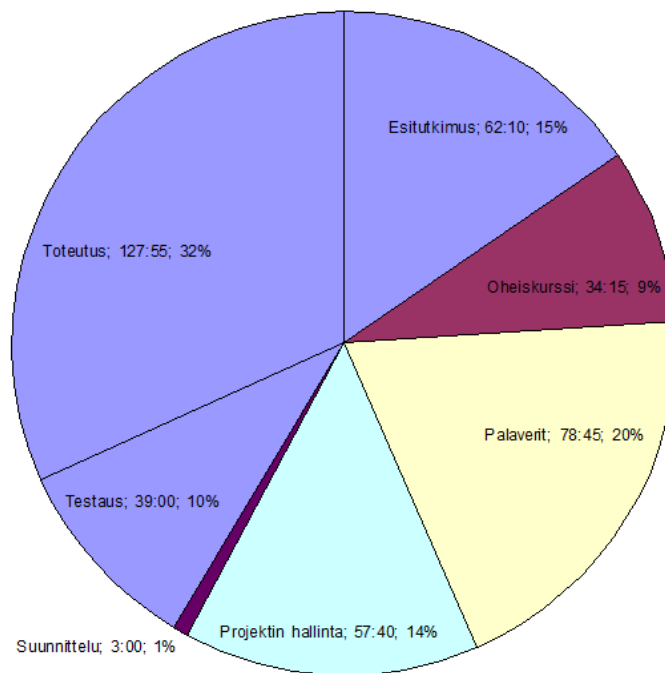
Ryhmän jäsenet tekivät yhteistyötä tehtävien jakamisessa osallistumalla sekä projektinhallintaan että sovelluksen kehittämiseen. Tehtävät olivat nimellisiä ja niiden hajauttaminen paransi ryhmän yhteishenkeä. Jäsenet vastasivat vastuualueensa tuloksista, mutta muut jäsenet osin kehittivät niihin kuuluvia tuloksia.

Projektin kestosta huolimatta yli 100 työtunnin ero jäsenten kesken on suuri etenkin, kun se vastaa noin neljäsosaa yksittäisen jäsenen kokonaistyötunneissa. Toisaalta onko syytä mennä projektin jäsenten tasolle, jos kokonaisuus pysyy budjetissa. Taulukon 6.11 mukaan jäsenten työtuntien keskimääräinen erotus vastasi koko projektin erotuksen suuruusluokkaa. Projektinhallintaan liittyi kaikissa projektin vaiheissa viikoittain toistuvia velvoitteita ja johdotehtäviä, mikä näkyi suurempina tuntimäärinä.

6.9 Iivanaisen työtunnit tehtäväkokonaisuuksittain

Iivanainen oli päävastuussa järjestelmätestauksesta ja toteutti liitedatoihin ja metatietoihin liittyvät ohjelmaosiot. Iivanainen toimi palavereissa useamman kerran sihteerinä osallistuen sitä kautta projektinhallintaan. Hän osallistui kaikkiin projektin tehtäväkokonaisuuksiin vaatimusmäärittelyä lukuun ottamatta. Iivanainen aloitti toteutuksen kustomoidun liukusäätimen vastuualueensa parissa, jonka valmistuttua muita sovellusosioita oli ehditty

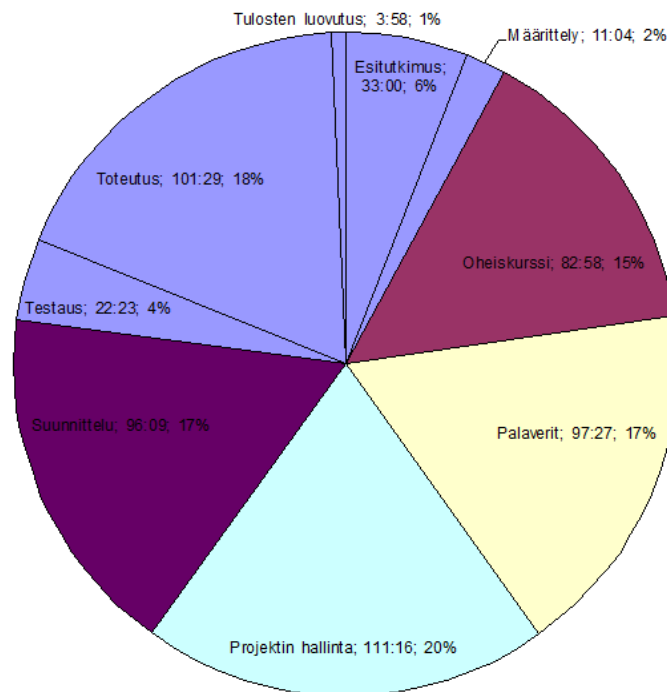
suunnitella yhdessä palavereissa valmiiksi. Liukusäädintä ennen Iivanainen perehtyi useamman viikon ajan SharpGL-kirjastoon, jonka tunnit kirjattiin esitutkimusvaiheeseen. Kuva 6.3 esittää Iivanaisen työtuntien jakautumista tehtäväkokonaisuuksiin.



Kuva 6.3: Iivanaisen työmäärät tehtäväkokonaisuuksittain.

6.10 Linnan työtunnit tehtäväkokonaisuuksittain

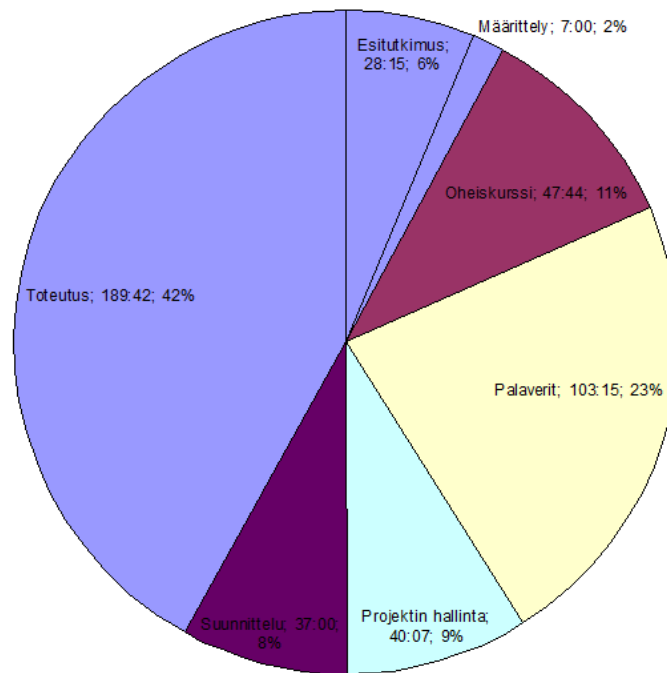
Linna oli projektipäällikkönä päävastuussa projektinhallinnasta, sovellussuunnitelmista ja toteutti tilakartan prototyypin. Linna toimi palavereissa puheenjohtajana ja neuvotteli asioista ryhmän puolesta. Projektisuunnitelman ja -raportin ulko- ja kirjoitusasun muokkaamiseen käytetyt työtunnit kirjattiin oheiskurssille, mikä selittää verraten suuret tuntimäärät. Kuva 6.4 esittää Linnan työtuntien jakautumista tehtäväkokonaisuuksiin.



Kuva 6.4: Linnan työmäärät tehtäväkokonaisuuksittain.

6.11 Pakkanen työtunnit tehtäväkokonaisuuksittain

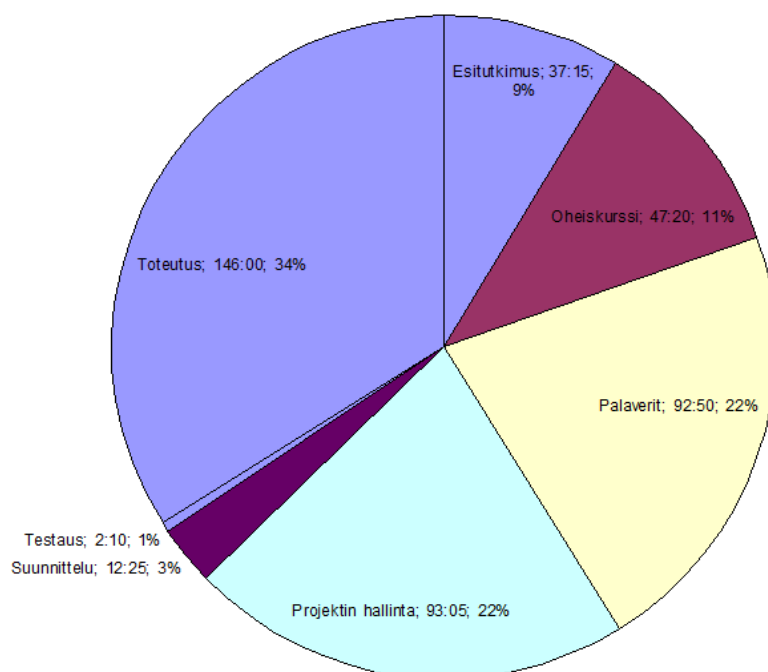
Pakkanen oli päävastuussa vaatimusmäärittelystä ja toteutti luokkarakenteen karttatiedostoihin liittyvät ohjelmaosiot. Pakkanen oli varapäällikkö ja toimi palavereissa toistuvasti puheenjohtajana. Vaativa ohjelmointitehtävä vei suurimman osan työtunneista. Kuva 6.5 esittää Pakkanen työtuntien jakautumista tehtäväkokonaisuuksiin.



Kuva 6.5: Pakkasen työmäärät tehtäväkokonaisuuksittain.

6.12 Vilavaaran työtunnit tehtäväkokonaisuuksittain

Vilavaara oli päävastuussa sovellusraportista sekä toteutti käyttöliittymään mukaan lukien sen arkkitehtuuriin liittyvät ohjelmaosiot. Vilavaara toimi palavereissa useamman kerran sihteerinä ja osallistui aktiivisesti projektinhallintaan. Sihteerinä toimiessaan hän kirjasi ylös palaverien päätöksiä ja valvoi näiden päätösten toteutumista. Kuva 6.6 esittää Vilavaaran työtuntien jakautumista tehtäväkokonaisuuksiin.



Kuva 6.6: Vilavaaran työmäärät tehtäväkokonaisuuksittain.

7 Prosessi ja aikataulu

Luvussa kuvataan projektissa noudatettua prosessia ja toteutunutta aikataulua. Projektiin suunniteltua alkuperäistä vaihejakoa ei tarkalleen ottaen noudatettu, koska valittu prosessi vaihdettiin toteutusvaiheen aikana. Lisäksi kaikkia ajankäyttöraportin määrittämiä vaiheita ei sisällytetty suunnitelmaan.

Prosessi epäilemättä vaikutti projektin tuloksiin, sillä projektin päätyminen viivästyí suunnitellusta kolmella kuukaudella. Tilaajan edustajat eivät olleet asettaneet projektille takarajaa ja tulokset luovutettiin projektisopimuksen päättymisen jälkeen. Projektin alkuperäinen aikataulu osoittautui optimistiseksi, sillä 1600 työtunnin tavoite olisi edellyttänyt suunnitellulla aikataululla projektiryhmältä noin 85 tunnin viikoittaista työmäärää.

7.1 Prosessi

Sovellusprojektin läpiviennissä sovellettiin takaisinkytkettyä vesiputousmallia, johon liittyvät päävaiheet ovat määrittely, suunnittelu, toteutus, testaaminen ja viimeistely. Prosessin vaiheita kuvataan tarkemmin projektisuunnitelmassa [6]. Projektissa noudatettu prosessimalli ei ollut ensimmäinen prosessi, jota projektissa sovellettiin, sillä alkuvaiheessa suunniteltiin käytettävän Scrumia, mutta jo suunnitteluvaiheessa havaittiin, ettei siihen mennessä noudatettu prosessi ollut riittävän ketterä ja projektin läpiviennissä oli nähtävissä selkeät vesiputousmallin kehitysvaiheet.

Scrumin jäänteet ovat nähtävissä tehtävienhallinnassa, jossa ryhmän viikkopalaverissa laadittiin vaiheen työlísta, johon nostettiin korkean prioriteetin viikoittaisia tehtäviä ominaisuus-listasta. Scrumin soveltamisessa epäonnistuttiin siinä, että viikoittaisten syklien (listojen) tehtävät (kortit) liittyivät samaan tehtäväkokonaisuuteen (vaiheeseen). Toteutusvaiheessa sovelluksen hahmotelmia ja toimintaa olisi kannattanut laajentaa inkrementaalisesti pienissä erissä, mutta toteumassa tähdättiin valmiiseen sovellukseen ennen ensimmäisten prototyyppien julkistamista.

Tehtävien jakaminen kunkin jäsenen vastuualueen mukaisesti aiheutti sen, että ominaisuuksia kehitettiin toisistaan erillään, jolloin ohjelmasta saatiin ajettava prototyyppi vasta toteutusvaiheen loppupuolella. Havainto otettiin puheeksi ryhmän jäsenten kanssa viikkopalaverissa, jossa projektipäällikkö ehdotti, että sovellusta kehitettäisiin sykleissä lisäten toimintoja pienissä erissä. Selkeät vastuualueet koettiin toimivana järjestelynä, joten ryhmän yhteiseksi kannaksi muodostui prosessin vaihtaminen vesiputousmalliin. Päätöksestä tiedotettiin projektiorganisaatiota keskustelua seuranneessa projektipalaverissa ja ryhmä siirtyi noudattamaan takaisinkytkettyä vesiputousmallia vastaavan ohjaajan ehdotuksesta. Tämä

tarjosi vaiheisiin kaivattua joustavuutta, jolloin tuloksia ei tarvinnut kiinnittää kuin vasta projektikansiota koostettaessa. Prosessimallin vaihtamiseen ajaututtiin siksi, että projektin alkuvaiheessa suunnitellun scrum-mallin käyttöönotossa epäonnistuttiin. Scrum-malli edellyttää yleensä kokenutta tiimiä ja scrum masteria, jolla on menetelmästä vankka kokemus. Kun näin ei ollut, oli prosessin vaihtaminen oikea ratkaisu. Etätyöskentely lisää haasteita scrum-mallilla toimittaessa.

Takaisinkytkettyvän vesiputousmallin eräs suurimmista hyödyistä liittyi kuitenkin aika-
taulujen suunnittelun suoraviivaisuuteen, sillä kehitysvaiheet asettuivat järjestyksessä Gantt-
kaavion aikajanelle. Aikataulusta tarvitsi määrittää vain tehtäväkokonaisuuksien sisältä-
mien tehtävien tulokset ja niiden työmääräarviot. Keskeisten tulosten valmistuttua siirryttiin
seuraavaan vesiputousmallin vaiheeseen, joka sisälsi hyvin samanlaisen tehtäväkoko-
naisuuden kuin edellisessä vaiheessa, mikä on nähtävissä Gantt-kaaviossa. Scrumin aikana
kokonaistyömäärää oli vaikeampi suunnitella tai arvioida etukäteen. Projektipäällikön pro-
jektinhallinnan tiedot olivat kenties riittämättömät ketterien prosessimallien suunnitteluun.

7.2 Projektin kehitysvaiheet

Esitutkimusvaiheen aikana tutustuttiin projektin toimintaympäristöön ja tilaajan esittämiin
tarpeisiin. Lisäksi kartoitettiin projektin tavoitteet, keskeiset reunaehdot ja ongelma-alueet.
Näiden perusteella hahmoteltiin projektille alustava aikataulu- ja työmääräarvio.

Määrittelyvaiheessa kartoitettiin sovellusvaatimukset. Vaiheen tuloksena tilaajalle toimitet-
tiin vaatimusmäärittely.

Suunnitteluvaiheessa laadittiin tilaajan edustajien kanssa keskustellen dokumentit, jotka
kuvaavat tarkasti sovellukselle asetetut tavoitteet ja tarpeet. Vaiheen aikana kuvattiin käyt-
töliittymä, arkkitehtuuri ja luokkarakenne. Suunnitteluvaiheen katsotaan käynnistyneen,
kun aloitettiin sovellussuunnitelmien laatiminen. Sovellussuunnitelmien tarkennuttua to-
teutuskelpoisiksi, siirryttiin toteutusvaiheeseen. Projektin aikana kehitettävä sovellus oli
uusi, joten toteutusvaiheeseen siirtyminen edellytti kehitysympäristön ja työkalujen valin-
taa.

Toteutusvaiheessa projektiryhmän kesken sovittiin kullekin ryhmän jäsenelle selkeät vas-
tuualueet, jotka määrittivät luonnollisesti tehtävänjakoa. Sovellussuunnitelmia noudatettiin
tarkasti. Joitain merkittäviä vaatimuksia jouduttiin rajaamaan jatkokehitykseen ennen tes-
tausvaiheeseen siirtymistä.

Sovelluksen testaaminen suunniteltiin toteutettavan testauksen V-mallin periaatteita nou-
dattaen. Vaatimusten toteutuminen todennettiin hyväksymistestauksen sijaan tilaajan edus-

tajien suorittamalla sovelluksen koekäytöllä. Sovelluksen toimivuuden projektiryhmä varmisti järjestelmätestauksella, mutta yksittäisiä aliohjelmia ei yksikkötestattu. Yksikkötestit olisi kannattanut lisätä projektin tuloksiin ja niiden toteutus vastuuttaa.

Viimeistelyvaiheessa laadittiin sovelluksen käyttöohjeet, sovellus- ja projektiraportti. Tulokset viimeisteltiin ja kerättiin projektikansioon arkistoitavaksi. Tilaajalle luovutettiin sähköinen versio projektikansiosta.

7.3 Aikataulu

Projekti alkoi 2.2.2021 ja sen suunniteltiin päättyvän 11.6.2021. Projektin toteutuneita tarkistuspisteitä olivat väliesittely, kaksi koodikatselmointia ja lähdekoodin hyväksymistarkistus. Projekti päätettiin ennen tulosten luovuttamista 31.8.2021. Aikataulullisesti projekti myöhästyi yksitoista viikkoa suunnitellusta. Keskeiset syyt myöhästymiseen olivat viivästymät toteutuksessa ja testauksessa, jotka johtivat siihen, että viimeistelyvaiheen tehtävät sijoittuivat keskelle loma-aikaa.

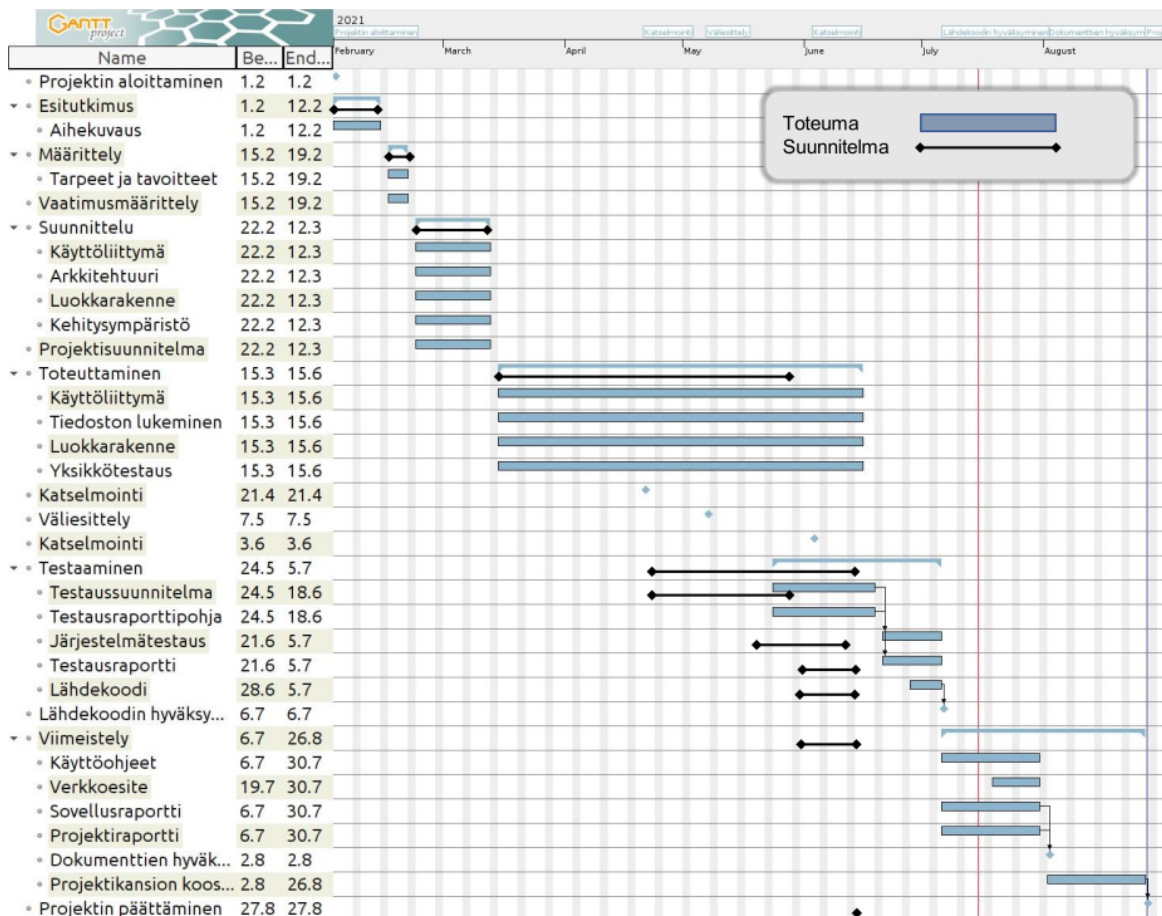
Esitutkimus-, määrittely- ja suunnitteluvaihe alkoivat ja päättyivät suunnitelman mukaisesti.

Toteutusvaihe aloitettiin suunnitelman mukaisesti, mutta toteuttaminen viivästyi suunnitellusta runsaat kaksi viikkoa valmistuen 15.6.2021. Viivästymisen olisi ollut tätäkin suurempi ilman toiminnallisuuteen tehtyjä rajauksia, jotka on kuvattu luvussa 3.3. Toteutusvaiheen viivästymisessä suurin yksittäinen syy oli projektiryhmän kokemuksessa, joka johti tiettyyn tehottomuuteen toteutusratkaisuja valittaessa sekä projektinhallinnan ja viestinnän vaatiman ajan aliarviointi.

Viivästymä toteutusvaiheessa vaikutti suoraan testausvaiheeseen, joka käynnistettiin neljä viikkoa suunniteltua myöhemmin. Testaus itsessään onnistuttiin toteuttamaan suunniteltua nopeammin ja se valmistui vain kolme viikkoa suunniteltua myöhemmin 5.7.2021.

Viimeistelyvaihe käynnistettiin viisi viikkoa suunniteltua myöhemmin ja pääosa työskentelystä ajoittui kesäaikaan, jonka vuoksi valmistuminen kesti suunniteltua pidempään. Viimeistelyvaiheeseen sisältyi neljän viikon ajanjakso, jolloin ei pidetty projekti- tai ryhmäpalavereja ja jäsenet keskittyivät itsenäiseen raporttien viimeistelyyn. Viimeistelyvaihe ja koko projekti valmistui 20 viikkoa suunniteltua myöhemmin 29.10.2021. Viimeistelyvaiheen viivästymässä konkretisoitui projektiryhmän kokemattomuus dokumentoinnissa. Projektidokumentointin kokoaminen ja viimeistely veivät kommenttikierroksineen suunniteltua enemmän kalenteriaikaa.

Projektin suunniteltu ja toteutunut aikataulu esitellään kuvassa 7.1, jossa otettiin pohjaksi projektisuunnitelman versio 1.0.0.

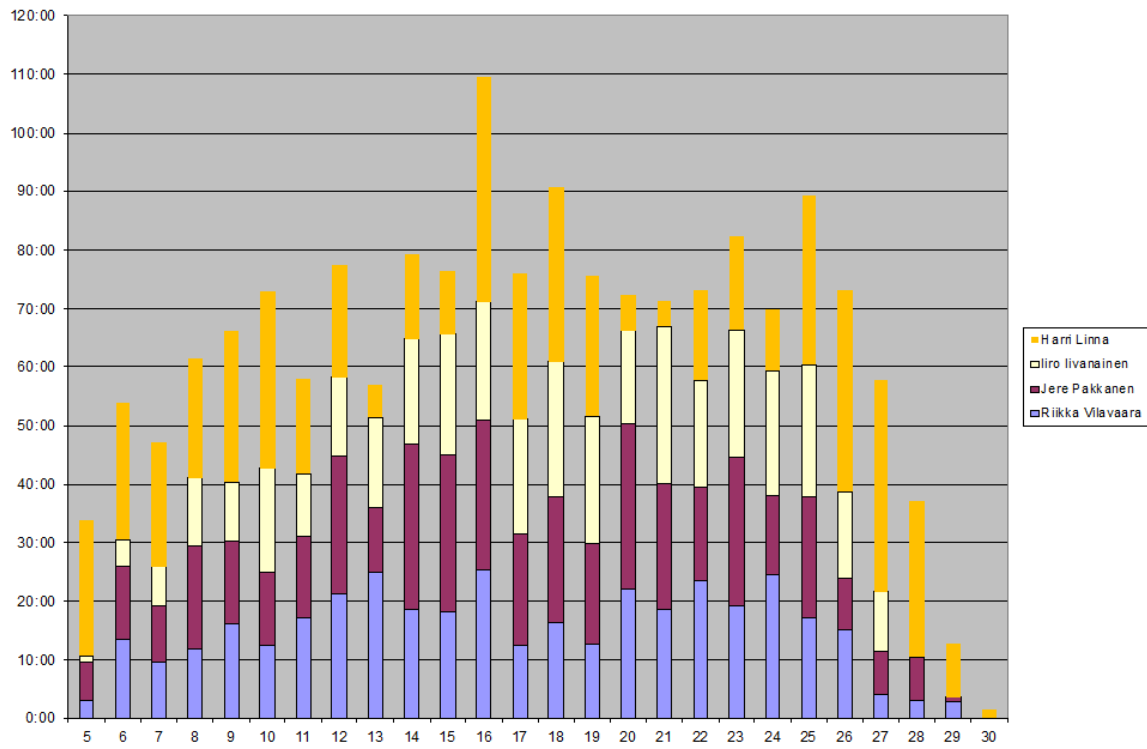


Kuva 7.1: Projektin aikataulu.

7.4 Ryhmän työtunnit viikoittain

Projektin alussa asetettua koko ryhmän viikoittaista tuntitavoitetta laskettiin 100 tunnista 70 tuntiin viikolla 14, mikä tarkoitti yhdeltä jäseneltä alle 20 tunnin työmäärää viikoittain. Projektiohjeen mukaan toteutusaikaa olisi noin 13–16 viikkoa, minkä perusteella jäsenten olisi tullut varata viikoittain vähintään 20–30 tuntia projektityöhön. Projektin työmäärätaavoite 17,5 tuntia viikossa oli huomattavasti vähemmän kuin suositeltu vähimmäistyömäärä, mutta se oli saavutettavissa. Ryhmän käyttämien työtuntien jakaantuminen eri viikoille esitetään kuvassa 7.2.

Projektissa päästiin tuntitavoitteeseen suurimpana osana viikoista. Ryhmän jäsenten työtunnit eivät jakautuneet tasaisesti projektin aikana, mutta yhteenlasketut tuntimäärät vastasivat ryhmän yhteistä tavoitetta. Projekti käynnistyi kunnolla huhtikuussa, kun sovelluksen

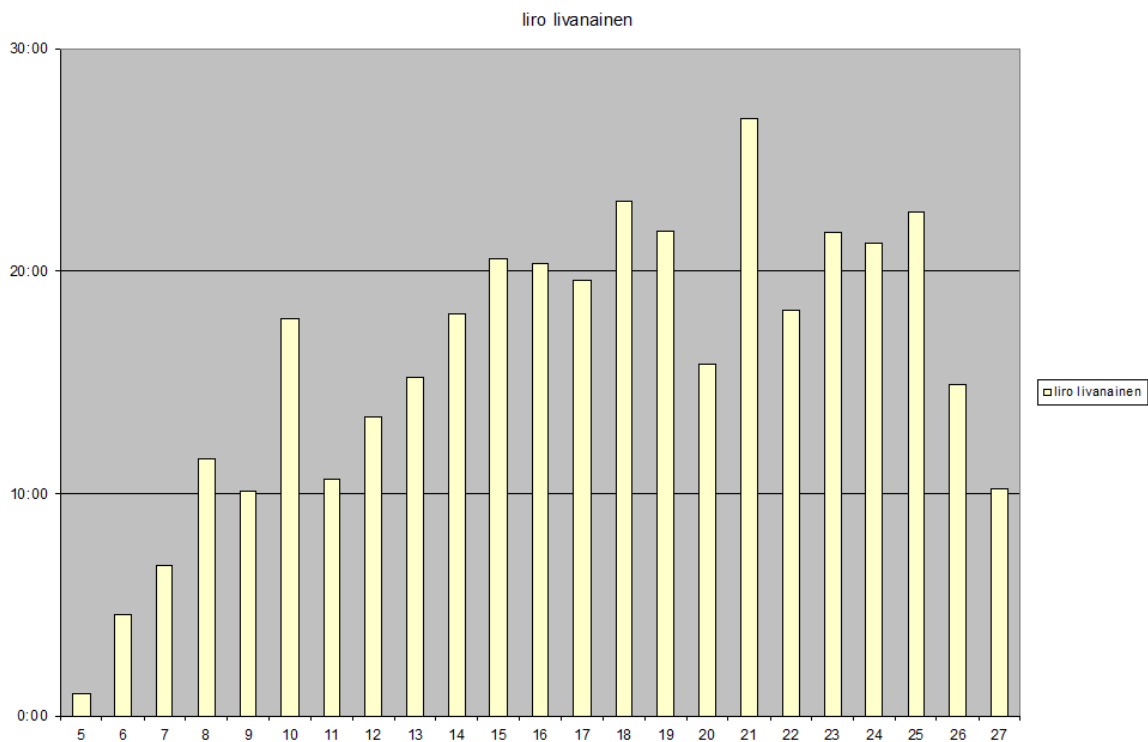


Kuva 7.2: Ryhmän työtunnit viikoittain.

tekninen puoli alkoi vähitellen hahmottua ja voitiin siirtyä toteutusvaiheeseen.

7.5 Iivanaisen työtunnit viikoittain

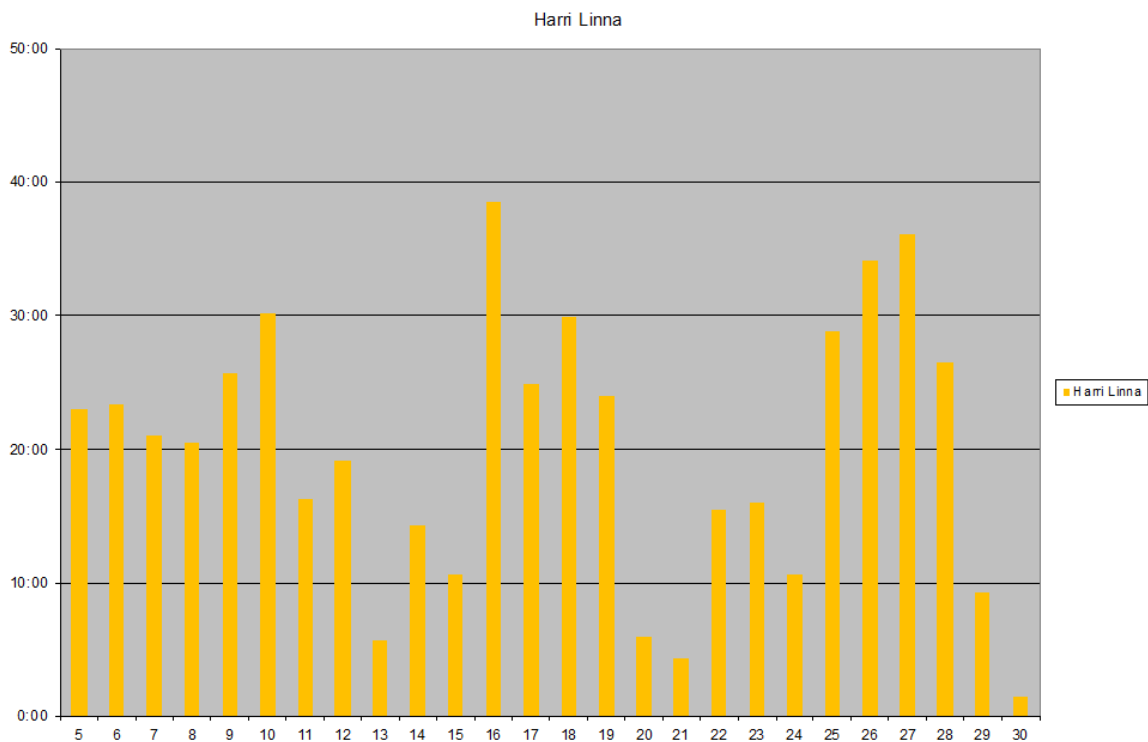
Suunnitteluvaiheessa Iivanainen jäi tuntitavoitteesta projektin alkupuolella useampana viikkona johtuen grafiikkakirjaston toiminnan haastavuudesta. Viikosta 14 lähtien hän pääsi joka viikko tavoitteeseen ja useampana viikkona yli tavoitteen, lukuun ottamatta kahta viimeistä viimeistelyvaiheen viikkoa. Projektin loppupuolella tunteja kertyi erityisesti testausvaiheen dokumenttien laatimisesta.



Kuva 7.3: Iivanaisen työtunnit viikoittain.

7.6 Linnan työtunnit viikoittain

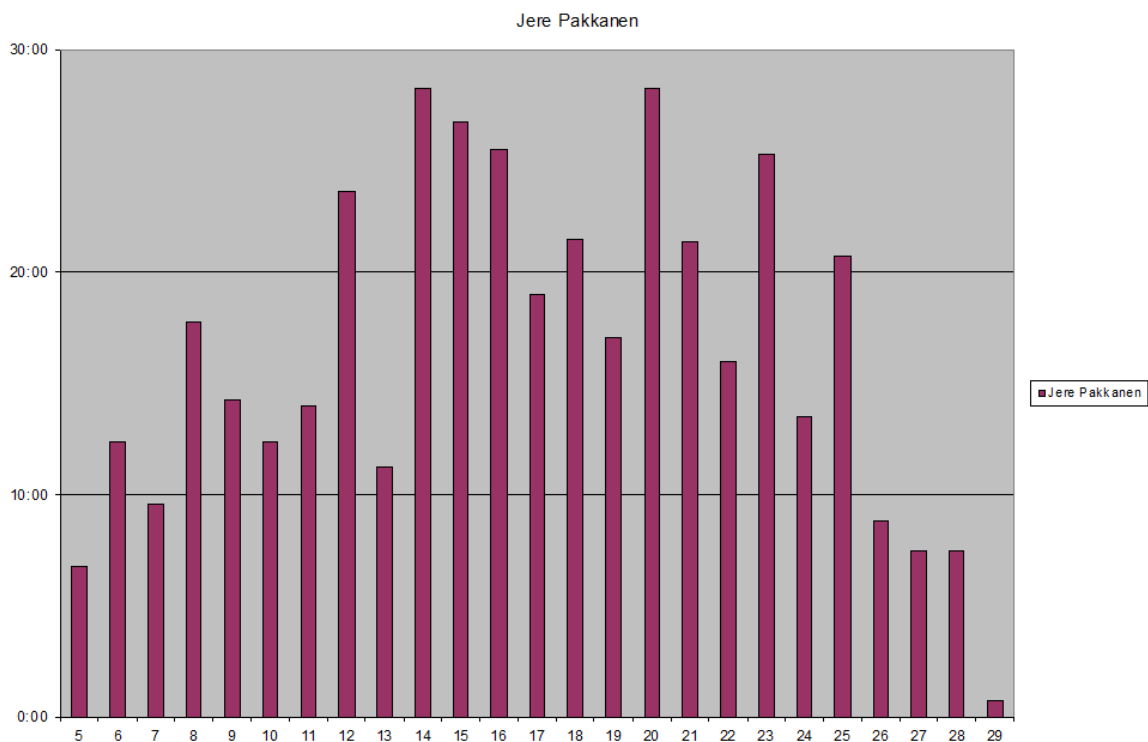
Linna ei osallistunut sovelluksen ohjelmointiin toteutusvaiheessa, joka sijoittui viikoille 11–24. Väliesittelyn jälkeen Linna toteutti tilakartan prototyypin ja ensimmäisen järjestelmätäuskerran, muuten hän keskittyi projektinhallintaan ja suunnitteludokumenttien laatimiseen. Testauksen jälkeinen kuukausi pyhitettiin projektiraportin tavoitteelliseen kirjoittamiseen.



Kuva 7.4: Linnan työtunnit viikoittain.

7.7 Pakkasen työtunnit viikoittain

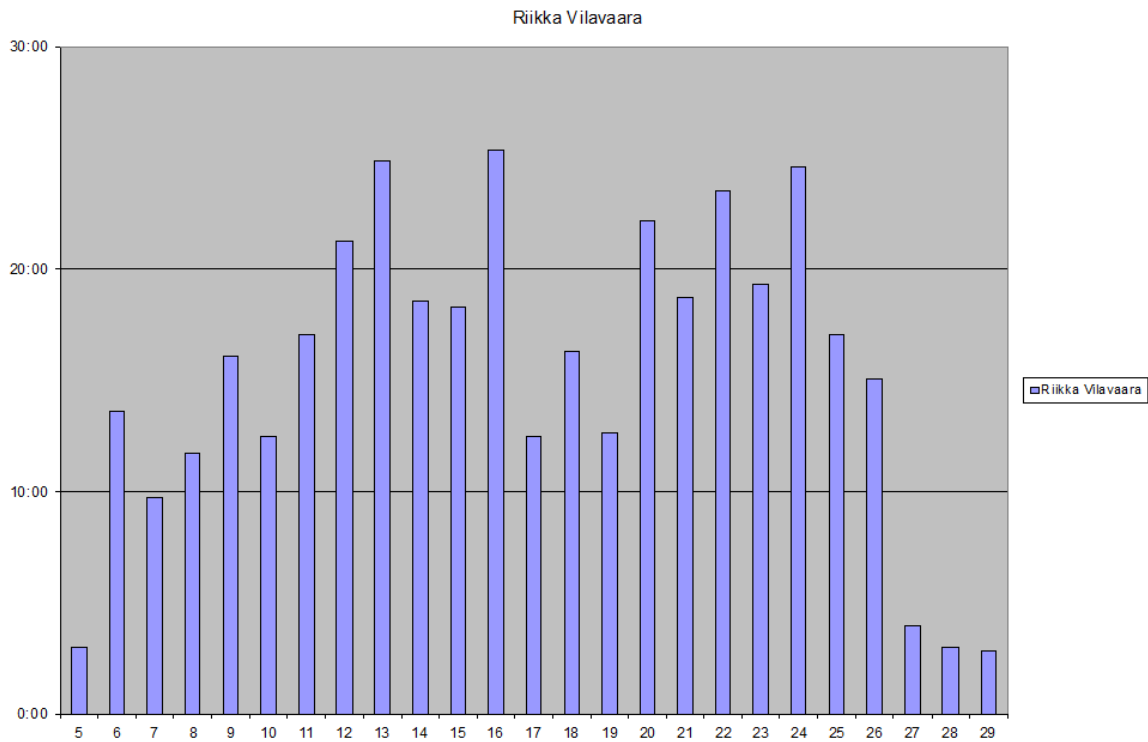
Pakkasen viikoittaiset työtuntimäärät vaihtelivat suuresti. Suurempien viikkojen ansiosta Pakkanen pystyi ottamaan hieman rennommin muilla viikoilla työtuntien suhteen. Vaatimusmäärittelyä päivitettiin usein samalla viikoilla kuin palavereja pidettiin, mikä näkyy poikkeuksellisen korkeina tuntimäärinä. Pakkanen osallistui muita enemmän sovelluksen vaativan tiedoston- ja virheenkäsittelyn toteuttamiseen, mutta jäi alle tuntitavoitteen projektin muissa vaiheissa.



Kuva 7.5: Pakkasen työtunnit viikoittain.

7.8 Vilavaaran työtunnit viikoittain

Vilavaaran viikoittaiset työtuntimäärät vaihtelivat myös suuresti. Suurin osa työtunneista sijoittui toteutusvaiheeseen, jossa hän toteutti suurelta osin käyttöliittymäpuolen ohjelmaosion. Testausvaiheen jälkeen Vilavaara kertoi sairastuneensa, mistä johtuen sovellusraportin kirjoittaminen siirtyi viimeistelyvaiheen alusta myöhempään ajankohtaan.



Kuva 7.6: Vilavaaran työtunnit viikoittain.

8 Riskit ja niiden hallinta

Luvussa kuvataan projektin ennakoitujen riskien toteutumista sekä niiden vaikutusta projektin läpivientiin ja tuloksiin. Lisäksi esitellään toimenpiteet riskin ennaltaehkäisyyn ja toipumissuunnitelmat riskien toteutumisen varalta. Riskien haittavaikutuksiin varauduttiin varaamalla työvaiheisiin arvioitua enemmän aikaa, mutta aikaresurssien ehtyessä sovelluksesta jouduttiin rajaamaan tärkeitä toimintoja jatkokehitykseen.

8.1 Riskien todennäköisyydet ja vakavuudet

Groundhog-projektin läpivientiin liittyviä riskejä arvioitiin nelikenttämallilla, jossa riskin toteutumisen todennäköisyys ja seurausten vakavuus luokiteltiin suureksi, kohtalaiseksi tai pieneksi. Riskit järjestettiin ensisijaisesti vakavuuden tai toissijaisesti todennäköisyyden perusteella. Jos joko riskin vaikutus tai todennäköisyys oli pieniä, kokonaisriski oli myös pieni eikä se vaatinut välittömiä toimenpiteitä. Jos sekä riskin vaikutus että todennäköisyys olivat suuria, myös kokonaisriski oli suuri ja toimenpiteet riskin ehkäisemiseksi ja toipumissuunnitelman laatimiseksi käynnistettiin välittömästi. Kaikissa muissa tapauksissa riski oli keskinkertainen ja edellytti suunnitelman laatimista riskin ehkäisyksi. Oheinen riskimatriisi kuvaa projektin alussa tunnistetut riskit ja toteutuneiden riskien osalta toteutuneen vakavuuden. Projektin aikana uusia riskejä ei tunnistettu.

Riskin kuvaus	Arvioitu todennäköisyys	Arvioitu vakavuus	Toteutunut vakavuus
Aikaresurssien rajallisuus	Suuri	Suuri	Kohtalainen
Tilojen ja laitteiden puute	Suuri	Kohtalainen	Kohtalainen
Ryhmän tietotaitojen puute	Kohtalainen	Kohtalainen	Kohtalainen
Ryhmän jäsenten muut sitoumukset	Kohtalainen	Kohtalainen	-
Ominaisuuksien toteuttamisen vaikeus	Pieni	Kohtalainen	-

Taulukko 8.1: Riskien arviointi.

Riskienhallintaa voidaan projektissa pitää onnistuneena. Riskit tunnistettiin jo projektin alussa ja toimenpiteet riskien todennäköisyyksien ja vaikutusten pienentämiseksi olivat pääsääntöisesti onnistuneita. Sovellus valmistui sovellukseen toteutettavien ominaisuuksien rajausten ja tilaajan esittämien muutosten jälkeen ajallaan. Voidaan kuitenkin arvioida, että joidenkin riskien kohdalla toimenpiteiden käynnistäminen vieläkin aikaisemmassa vaiheessa projektia olisi pienentänyt riskin vaikutusta entisestään. Yhden riskin (ryhmän tietotaitojen

puute) osalta arvioitua todennäköisyyttä ei seurattu riittävästi ja riski toteutui ilman, että siihen olisi varauduttu laatimalla toipumissuunnitelma. Suuremmassa projektissa riskienhallinnan systemaattisuuteen tulee kiinnittää enemmän huomiota kuten myös käytettäviin hallintavälineisiin.

8.2 Aikaresurssien rajallisuus

Toteuma: Riski toteutui osittain. Kaikkia suunniteltuja toimenpiteitä ei pystytty toteuttamaan ja osa toteutettiin ajallisesti liian myöhään, mutta vaikutus onnistuttiin rajaamaan hyväksyttävälle tasolle.

Kuvaus: Aikaresurssien rajallisuus liittyy ohjelmiston laatutavoitteisiin. Hyvä ohjelmisto vaatii perusteellista suunnittelua, järjestelmällistä testaamista sekä runsaasti iteraatioita ohjelmakoodin refaktoroinnissa. Projekti toteutetaan annettujen aikaresurssien puitteissa.

Vaikutus: Projektia toteutettiin takaisinkytketyn vesiputousmallin mukaisesti. Riski konkretisoitui toteutusvaiheessa, jolloin osa suunnitellusta toteutuksesta siirrettiin jatkokehityksessä toteutettavaksi.

Toimenpiteet riskin ennaltaehkäisyyn:

- Vaatimusmäärittelyssä priorisoitiin kriittisimmät toiminnallisuudet toteutettavaksi ensimmäisinä (toteutettu).
- Sovellusta kehitettiin inkrementaalisesti lisäämällä uusia ominaisuuksia muokaten iteraatiivisesti olemassa olevaa takaisinkytketyn vesiputousmallin mukaisesti. Toimiva sovellus saatiin kuitenkin koottua yhteen varsin myöhäisessä vaiheessa toteutusta, tässä Scrum-prosessi oli saattanut toimia paremmin (toteutui osittain).
- Arkkitehtuuria ja luokkarakennetta suunniteltaessa hyödynnettiin C#-kielen standardikirjastoja sekä geneerisiä tyyppejä rajapinnoissa. Tämä mahdollisti sovelluksen joustavan muokkaamisen (toteutettu, osoittautui kuitenkin joissain tilanteissa vääräksi valinnaksi).
- Käyttöliittymä suunniteltiin käyttävän ikkunointia, joka helpotti moduulien lisäämisen ja korvaamisen (toteutettu).
- Suunnittelu- ja toteutusvaihe aloitettiin mahdollisimman varhaisessa vaiheessa ja laadukas prototyyppi ennätettiin toteuttaa (toteutettu).
- Tavoitteena oli, että yksikkötestaus aloitetaan samaan aikaan kuin toteutus, jotta niitä voisi suorittaa samanaikaisesti. Tähän ei kuitenkaan päästy ja sovellusta ei yksikkötestattu kehityksen aikana (ei toteutettu).

Toipumissuunnitelma riskin toteutumisen varalta:

- Projektin aikana tilaajan kanssa sovittiin tietyt toiminnallisuudet siirrettäväksi jatkokehitykseen. Projektin päättyessä voitiin toimittaa toimiva prototyyppi vaikkakin toiminnallisuudeltaan suunniteltua suppeampana (toteutettu).
- Projektiryhmä laati vaatimusmäärittelyn, sovellusraportin, käyttöohjeen ja testausdokumentit. Tavoitteena oli, että projektiryhmä olisi dokumentoinut myös jatkokehitykseen siirretyn toiminnallisuuden sillä tasolla, että toteutusta olisi näiltä osin voitu aikanaan suoraan jatkaa. Tähän ei kuitenkaan päästy (ei toteutettu).

8.3 Tilojen ja laitteiden puute

Toteuma: Riski toteutui jo ennen projektin käynnistymistä.

Kuvaus: Projektin käynnistysvaiheessa tavoitteena oli tehdä töitä yhteisissä tiloissa ja projektin tarjoamilla laitteilla. Covid19-pandemien aiheuttamien yliopiston tilojen käyttörajoitusten takia, ryhmä ei saanut huonetta käyttöönsä ja projektin jäsenten tuli työskennellä henkilökohtaisilla laitteilla.

Vaikutus: Sovelluksen tuli vaatimusmäärittelyn mukaan toimia millä tahansa perustason tietokoneella, joten laitteiden vaikutus projektiin ei ollut suuri. Kaikilla projektilaisilla ei kuitenkaan ollut samanlaista laitekokoonpanoa, mikä aiheutti pieniä viiveitä kehityksessä ja vaikeutti työnjakoa. Yhteisten tilojen puuttuminen vaikutti ryhmätyöskentelyyn, jolle löydettiin korvaavia muotoja.

Toimenpiteet riskin ennaltaehkäisyyn:

- Riski toteutui jo ennen töiden käynnistymistä, joten ennaltaehkäisyä ei kyetty suunnittelemaan.

Toipumissuunnitelman riskin toteutumisen varalta:

- Projektiryhmä työskenteli kokonaan etänä henkilökohtaisilla tietokoneillaan. Tietotekniikan opiskelijoilla oli riittävät työvälineet työpöytäsovelluksen kehittämiseen (toteutettu).
- Otettiin käyttöön etätyöskentelyä tehostava Discord-kanava, johon kuului ryhmän jäsenten lisäksi tekninen ohjaaja. Ryhmä kokoontui puheyhteyden ylitse neuvottelemaan kolmesti viikossa (maanantaisin, keskiviikkoisin ja perjantaisin) ajankohtaisista toimenpiteistä (toteutettu).

8.4 Ryhmän tietotaitojen puute

Toteuma: Riski toteutui. Toimenpiteiden ansiosta vaikutus onnistuttiin rajaamaan hyväksyttävälle tasolle siirtämällä tilaajan kanssa sovitusti osa toiminnallisuuksista jatkokehitykseen.

Kuvaus: Projektiryhmän kokemus kyseessä olevan kokoluokan sovellusprojektista sekä siihen liittyvästä projektityöskentelystä oli vähäistä.

Vaikutus: Resurssien käytön tehottomuuden seurauksena osa sovelluksen ominaisuuksista jouduttiin rajaamaan toteutuksen ulkopuolelle.

Toimenpiteet riskin ennaltaehkäisyyn:

- Tilaajan edustajilla oli runsaasti tietotaitoa aihealueesta, joten projektiryhmä sai on saanut erinomaista ohjeistusta toteutusratkaisuihin (toteutettu).
- Projektiryhmä valitsi ohjelmointikielen ja kehitysympäristön sekä sovittujen lisenssien puitteissa myös ulkoiset kirjastot, mutta kokemuksen puute näkyi erityisesti tehdyissä valinnoissa (toteutettu).
- Teknisellä ohjaajalla oli runsaasti kokemusta ohjelmistokehitysprojekteista, ohjelmoinnista ja kuvankäsittelyn ohjelman kehittämisestä ja tätä hyödynnettiin (toteutettu).

Toipumissuunnitelma riskin toteutumisen varalta:

- Riski toteutui ilman toipumissuunnitelmaa. Toipumisessa hyödynnettiin toimintojen siirtämistä jatkokehitykseen, joka sovittiin tilaajan kanssa.

8.5 Ryhmän jäsenten muut sitoumukset

Toteuma: Riski onnistuttiin suunnitelluilla toimenpiteillä välttämään. Jäsenille ei tullut enakoimattomia sitoumuksia.

Kuvaus: Projektiryhmän jäsenten muut sitoumukset olisivat saattaneet viivästyttää projektin läpivientiä. Kaksi ryhmän jäsenistä kirjoitti pro gradu -tutkielman samaan aikaan, mikä esti täysipäiväisen työskentelyn. Tilaajan edustajat eivät vaatineet projektin valmistuvan tietyssä aikataulussa, mutta projektisopimuksessa määritettiin ehdoton takaraja projektin valmistumiselle.

Vaikutus: Projekti viivästyi, mutta ei tämän riskin vuoksi.

Toimenpiteet riskin ennaltaehkäisyyn:

- Ryhmä asetti itselleen viikoittaisen 100 tunnin tavoitteen, johon sovelluksen edistymistä peilattiin viikoittaisessa tilakatsauksessa. Viikkotavoite pienennettiin projektin aikana 70 tuntiin (toteutettu).
- Poissaoloista aiheutuneet vaikutukset ehkäistiin siten, että muista sitoumuksista ilmoitettiin ryhmälle etukäteen. Tällöin ryhmän muut jäsenet sopivat, ketkä suorittavat poissaolevan tehtävät (toteutettu).

Toipumissuunnitelma riskin toteutumisen varalta:

- Kokonaisriski ei noussut eikä toipumissuunnitelmaa laadittu.

8.6 Ominaisuuksien toteuttamisen vaikeus

Toteuma: Riski onnistuttiin suunnitelluilla toimenpiteillä välttämään.

Kuvaus: Toteutettavat ominaisuudet oli määritetty selkeästi. Tästä huolimatta toteutus on osoittautunut haastavaksi versiohallinnan lukuisista *fix*-alkuisista komennoista päätellen. Käsiteltävät tiedostot olivat suuria, joten niitä ei välttämättä voinut lukea muistiin kerralla. Myös käsiteltäviä tiedostotyyppisiä oli erilaisia, mikä lisäsi toteutuksen vaikeutta.

Vaikutus: Projekti viivästy ja osa toiminnallisuuksista siirrettiin jatkokehitykseen. Syynä tähän ei kuitenkaan ollut ominaisuuksien toteuttamisen vaikeus.

Toimenpiteet riskin ennaltaehkäisyyn:

- Haastavien toimintojen osalta toteuttajat ottivat yhteyttä tekniseen ohjaajaan tai tilaajan edustajiin (toteutettu).
- Ryhmäpalaverissa tarkennettiin toteutusratkaisuja (toteutettu).
- Selvitettiin mahdollisuudet valmiskomponenttien käyttöön esimerkiksi datan lukemisen osalta (toteutettu).

Toipumissuunnitelma riskin toteutumisen varalta:

- Kokonaisriski ei noussut eikä toipumissuunnitelmaa laadittu.

9 Jäsenten kokemuksia ja oppimaa

Luvussa kuvataan projektiryhmän jäsenten kokemuksia ja oppimaa sovellusprojektista. Projekti oli kaikille ryhmän jäsenille ensimmäinen oikea ohjelmistokehitysprojekti, joka sisälsi projektin kaikki kehitysvaiheet aina vaatimusmäärittelystä projektin tulosten luovutukseen asti. Ryhmän jäsenet kokevat oppimistavoitteidensa toteutuneen hyvin ja ovat olleet tyytyväisiä päästyään mukaan projektiin. Ryhmän jäsenet saivat kattavan kuvan projektinhallinnasta ja tilaajien kanssa toimimisesta, vaikka projektin tulokset toimitettiin vasta sopimuksen määräajan päättymisen jälkeen. Luvuissa 9.1–9.4 ryhmän jäsenet kuvaavat kokemuksiinsa ja oppimiaan asioita.

9.1 Iivanaisen kokemuksia ja oppimaa

Iivanainen käsitti projektin tilaisuutena haastaa omia ohjelmointitaitojaan projektin laajuudessa ohjelmointityössä. Hänellä oli päälimmäisenä ajatuksena riittävätkö taidot tämän laajuiseen ohjelmistokehitykseen. Iivanainen arveli, että etäaikana voisi ottaa palaverien osalta hieman rennommin, mutta päinvastoin yllättyikin erilaisten ohjeiden määrästä.

Kirjoittamisesta ja ohjelmoinnista opittiin suhteellisen paljon sekä projektin hallinnasta tarvittavista niin sanotuista pehmeistä taidoista. Iivanainen kertoo näiden taitojen parantuneen käytännön kokemuksen kautta enemmän kuin niihin tarkoitetuilla kursseilla aikaisemmin. Toisten ohjelmakoodiin ei tarvinnut juurikaan puuttua, koska ryhmällä oli selkeät vastualueet. Ryhmätyöskentelyssä taas enempi kommunikoinnin määrä oli vähempää parempi. Projekti koettiin rentona suorittamisena, vaikka vastuuta ja painetta oli viedä sitä eteenpäin.

Palaverien pöytäkirjat ovat suurelta osin Iivanaisen käsialaa. Virallisempina vastuualueena olivat järjestelmätestauksen dokumentit, joiden laatimisessa otettiin mallia aiempien ryhmien valmiista dokumenteista. Hän huomaa hyödyntävänsä nykyisin enemmän hahmola-keja sekä muita tekstin luettavuutta parantavia asioita. Myös tekstin muotoiluun kiinnittää enemmän huomiota. Sovelluksen liitedatojen käsittelyyn ja käyttöjärjestelmään liittyvien ohjelmaosien toteuttamisen myötä Iivanainen hahmottaa paremmin ohjelmaosien ja vastuualueiden rajoja. Hänen mukaansa ohjelmakoodin parantaminen tehostuu ja tarvittaessa voi kohdistaa kysymyksiä vastuuhenkilölle, kun kehittäjillä on selkeät vastualueet.

9.2 Linnan kokemuksia ja oppimaa

Odotin kurssilta tilaisuutta osallistua projektiin, joka olisi sisältänyt enemmän käytännön ryhmätöitä ja vähemmän itsenäisiä kirjoitustehtäviä. Aihekuvausta lukiessani saatoinkin olla tyytyväinen ryhmävalintaan, vaikka en tuntenut muita ryhmän jäseniä entuudestaan. Ensimmäisessä valmistauduin ohjelmoijan rooliin, mutta ryhdyin sitten oma-aloitteisesti projektipäälliköksi.

Projektia sitoivat yliopiston mallit ja odotukset, joita noudattamalla olisi varmaankin pääsyt vähemmällä. Palaverija lukuun ottamatta projekti koostui omalla kohdallani itsenäisesti suoritettavista kirjoitustehtävistä. Ryhmätööt menevät monesti etänä sellaiseksi, että sovietaan yhdessä tehtävienjaosta, mutta varsinainen työnteko tehdään itsenäisesti.

Projektipäällikön vastuualue koostui projektinhallinnan tehtäväkokonaisuudesta. Kokemukseni mukaan hyvällä projektihallinnalla voidaan tunnistaa muutoksia vaativia toimenpiteitä, mutta projektipäälliköltä puuttuu todellinen vaikutusvalta opiskelijaprojekteissa. Projekti alkaa olla sen verran myöhässä, ettei projektinhallinta ole enää kovin mieluisaa.

9.3 Pakkasen kokemuksia ja oppimaa

Ennen projektin alkamista Pakkanen tiesi odottaa suurta työmäärää, sillä hän oli tutustunut aiempien vuosien projekteihin ja projektiohjeeseen. Muita odotuksia liittyi palaverien virallisuuteen ja pöytäkirjojen laatimiseen. Aluksi työmäärä arvelutti, mutta projektin edetessä ennakkoluulot hälvenivät havainnon myötä, että uuden opettelua ei ollut kauheasti. Suuresta työmäärästä huolimatta projektissa ei ollut yhtä paljon uuden opettelua kuin luentokursseilla mihin oltiin tottuneita.

Etätyöskentely on ollut Pakkasesta toimivaa, vaikka etänä tekeminen vaikuttikin projektin taustalla. Työskentely olisi varmasti ollut erilaista, nopeampaa työtahtia kenties, jos oltaisiin oltu samaan aikaan kaikki paikalla. Ryhmätyöskentely on toiminut paremmin kuin muilla kursseilla satunnaisesti muodostetuissa ryhmissä, joiden ongelmia ei ilmennyt niinkään paljon. Toteutettavia asioita oli paljon, mutta jokainen teki oman osuutensa. Palaveritkin olivat sopivan muodollisia, ettei niiden virallisuutta tarvinnut jännittää.

Varaprojektipäällikön tittelistä ei tullut ylimääräistä työtä, sillä projektipäällikkö ei pitänyt lomaa. Kerran tapahtui läheltä piti -tilanne. Palaverien puheenjohtajana Pakkanen onnistui mielestään paremmin projektin alkuvaiheissa. Hänestä pieni esiintymisjännitys edesauttaa kommunikaatiotaitojen kehittymisessä. Pakkanen aloitti vaatimusmäärittelyn laatimisen heti projektin alussa, joten kerralla tehtävä kirjoitusmäärä pidettiin kohtuullisena. Jos

Pakkanen tekisi kuvankäsittelyn alusta, niin hän tekisi muutamia asioita eri tavalla. Kuvankäsittely saatiin kuitenkin pääosin toimivaksi.

9.4 Vilavaaran kokemuksia ja oppimaa

Projektin suuri työmäärä oli Vilavaaran tiedossa hyvissä ajoin ennen projektin alkamista, mutta häntä mietitytti tuleeko työmäärä lopulta vastaamaan ennakko-odotuksia. Rajoitusten takia kurssin suorittaminen ylipäänsä ei ollut itsestänselvyys, esimerkiksi siinä miten viralliset palaverit tulisivat vaikuttamaan tilaajan ja ohjaajien kanssa toimimiseen. Poikkeusjärjestelyt vaatisivat varmasti muutakin kuin luentojen siirtäminen etäopetukseen. Vilavaara tiedosti, ettei projekti tulisi olemaan pelkkää koodausta, vaan siihen sisältyisi myös runsaasti kirjoittamista.

Vilavaara kertoi yliarvioineensa raportoinnin vaativuuden, sillä dokumenttien laatiminen ei vastannut muiden opiskelijoiden puheista saatua mielikuvaa. Palavereista ei ollut aiempaa kokemusta eivätkä ennakko-odotukset vastanneet siten juurikaan hänen projektikokemuksiaan. Vilavaara korostaa vuorovaikutuksen merkitystä tilaajan edustajien ja ohjaajien kanssa toimimisessa, missä on ensiarvoisen tärkeää pitää usein yhteyttä kuin jättää yhteydenpito vähemmälle. Sovelluksen osalta tekemisen määrä ei yllättänyt vaan ohjelmointia jopa odotettiin.

Dokumenttien kirjoittaminen on harjaannuttanut virallisen tekstin tuottamisessa. Palautteen perusteella on saanut huomioita siitä, missä on voinut vielä parannettavaa. Käyttöliittymän toteuttamisessa Vilavaara tunnistaa vielä parantamisen varaa. Ensimmäinen idea ei välttämättä ole se paras, sillä vasta kokeillessa saattaa huomata siitä heikkouksia. Projektissa opittiin paljon niissä tilanteissa, joissa aikaisemman osaamisen puolesta ei ollut apua. Vilavaara huomaa kirjoittavansa nykyään valmiimpaa koodia, joka tarvitsee jälkeenpäin vähemmän refaktorointia.

10 Yhteenveto

Groundhog-projekti kehitti kevään 2021 Sovellusprojekti-kurssilla Geologian tutkimuskeskukselle multimodaalisen tutkimusdatan hallintaohjelmiston tukemaan tomografialaboratorion näytteiden analyysiä. Prototyypin helpottaa erityyppisten tutkimusdatojen yhdistämistä, hallintaa ja visualisointia, mutta sovelluksessa riittää vielä runsaasti kehitettävää ennen sen käyttöönottoa.

Projektissa tärkeimmät sovellukselle asetetut tavoitteet toteutuivat suurelta osalta. Kaikkia vaatimusmäärittelyssä suunniteltuja toimintoja ei ehditty toteuttaa, mutta sovellukseen kehitettävistä toiminnoista sovittiin tilaajan kanssa. Sovelluksen kehittäminen jatkuu projektin jälkeen avoimen lähdekoodin ohjelmistona tilaajan toimesta.

Projektissa saavutettiin tavoitteen mukainen enimmäistyömäärä ja ryhmän jäsenet sitoutuivat projektiin aikataulun pidentämisestä huolimatta. Tulokset luovutettiin sopimuksen vastaisesti myöhässä. Projektin päättyessä projektiryhmän jäsenet tuntevat sovellusprojektin keskeiset projektinhallinnan menetelmät ja osaavat soveltaa niitä käytännön projektityössä.

Vastaavan ohjaajan mukaan toteutunutta työmäärää ei pystytty luotettavasti vertaamaan suunniteltuun projektisuunnitelman työmäärien arvioinnin puutteista ja työtuntien kirjausten virheistä johtuen. Projektisuunnitelma hyväksyttiin vasta kesäkuun puolessa välissä ja sen viivästyessä projektin jäsenillä ei todennäköisesti ollut tiedossa projektin läpiviennin käytänteet, mikä osin saattoi johtaa epäyhtenäiseen toimintaan – ainakin kirjauskäytänteiden osalta.

Lähteet

- [1] Jarkko Aalto, Timo Konu, Samuli Kärkkäinen, Samuli Rahkonen ja Miika Raunio. 2013. "Potku-sovellusprojekti, Projektiraportti". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [2] Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. 2021. "Groundhog-sovellusprojekti, Arkkitehtuurisuunnitelma". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [3] Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. 2021. "Groundhog-sovellusprojekti, Ajankäytönseurantatyökaluja". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [4] Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. 2021. "Groundhog-sovellusprojekti, Käyttöliittymäsuunnitelma". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [5] Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. 2021. "Groundhog-sovellusprojekti, Luokkasuunnitelma". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [6] Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. 2021. "Groundhog-sovellusprojekti, Projektisuunnitelma". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [7] Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. 2021. "Groundhog-sovellusprojekti, Sovellusraportti". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [8] Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. 2021. "Groundhog-sovellusprojekti, Testausraportti 1–2". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [9] Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. 2021. "Groundhog-sovellusprojekti, Testaussuunnitelma". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [10] Iiro Iivanainen, Harri Linna, Jere Pakkanen ja Riikka Vilavaara. 2021. "Groundhog-sovellusprojekti, Vaatimusmäärittely". Jyväskylän yliopisto, informaatioteknologian tiedekunta.

-
- [11] Mari Kasanen, Leevi Liimatainen, Marina Mustonen, Juhani Sundell ja Arttu Ylä-Sahra. 2019. "Peltihamsteri-sovellusprojekti, Projektiraportti". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [12] Joel Kivelä, Erkki Koskenkorva, Mika Lehtinen, Oskari Leppäaho ja Petri Partanen. 2014. "Liikkuva-sovellusprojekti, Projektiraportti". Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [13] Microsoft Corporation. 2008. "Capitalization Conventions". Saatavilla WWW-muodossa, <http://msdn.microsoft.com/en-us/library/ms229043.aspx>, viitattu 21.7.2021.
- [14] Santanen, Jukka-Pekka. 2017. "Tietotekniikan Sovellusprojektien ohje". Jyväskylän yliopisto, informaatioteknologian tiedekunta.