Timo Poranen (ed.)

Software Projects 2008-2009

DEPARTMENT OF COMPUTER SCIENCES
UNIVERSITY OF TAMPERE

D-2009-6

TAMPERE 2009

UNIVERSITY OF TAMPERE
DEPARTMENT OF COMPUTER SCIENCES
SERIES OF PUBLICATIONS D – NET PUBLICATIONS
D-2009-6, AUGUST 2009

Timo Poranen (ed.)

Software Projects 2008-2009

DEPARTMENT OF COMPUTER SCIENCES
FIN-33014 UNIVERSITY OF TAMPERE

ISBN 978-951-44-7829-1
ISSN 1795-4274

Preface

This report contains project stories of 10 software development projects and one usability team project. The students came from Project Work and Software Project Management courses. The stories describe how the project went and what kind experiences the groups got during their project. In the end of each story there are statistics of the project.

Table 1 contains general course statistics (number of projects and usability teams, number of students in the courses and average project size in working hours) starting from year 2005.

The number of project work course students decreased from the previous years, and at the same time the number of project managers increased causing the ratio of managers and developers to be highly unrealistic when compared to work life. This can be seen also from the working hours of the projects, the average number of working hours was over six hundred higher than in the previous course.

Table 1: Course statistics 2005-2009.

Academic year	Projects	Usability teams	PW students	SPM students	Average project size
2005-6	19	1	98	8	1008 hours
2006-7	18	2	87	34	1089 hours
2007-8	14	1	70	29	997 hours
2008-9	10	1	60	39	1643 hours

Table 2 gives an overview of the projects of this year. The table lists project's name, project's type (WWW = a WWW application often containing a database, Mobile = a Mobile phone application), client (Dept. = Department of Computer Sciences, Univ. = other university units, Assoc. = association, Company = a commercial company), or used software development model (INC = Incremental, ITER = Iterative or SCRUM = agile Scrum development model), group size (Number of managers + size of the project team + usability experts), and total working hours of the project.

During the course project managers kept up the working hours of the project team. The working hours were divided into nine activities: project planning and management, requirements specification, software design, code (programming), integration and testing, reviews, repair (bug fixing), studying and other. All projects, except usability team, gave their working hours divided into these categories.

Table 2: General project statistics.

Project	Type	Client	Dev. Model	Group	Hours
Uteam	-	Dept.	All	3+7	1970
Mindwiki	WWW	Univ	Scrum	4+5+1	1319
EDP	WWW	Univ	ITER	4+5+1	1531
Majava	WWW	Dept.	INC	4+5+1	1100
Collikka	WWW	Assoc.	Scrum	4+6+1	2266
SMSRaptori	WWW	Company	INC	4+5	1185
Vixtory	WWW	Company	Scrum	3+5+1	1127
LiputON	WWW	Company	INC	3+5+1	1915
Green Quest	WWW	Company	Scrum	4+4+1	1556
Playful UI	Mobile	Company	Scrum	3+7+1	2852
Novel UI	Mobile	Company	Scrum	3+6+1	1576

The statistic sections of almost all projects contain tables for general project information, requirements and high-level design outcome, design outcome, projects documents, codelines, and productivity metrics.

All projects succefully produced working software, and it was hard to select a candidate for the award of the Information Processing Association of Tampere Region (Pirkanmaan Tietojenkäsittely-yhdistys ry. (PITKY, `http://www.pitky.fi`). After counting votes given by the students in the presentation days of the course, two teams had equal number of votes: LiputON and Usability Team. Usability team experiment started four years ago, and therefore, due to long history of the Usability Team, the department decided to propose them. Board of the PITKY ry gave the award to the Usability Team during the Project Management day 2009 (`http://www.cs.tut.fi/tapahtumat/projektinhallinta09/`) on 12th August.

Table 3: PITKY awards.

Academic year	Project
2004-5	Fysio-laadunhallintajärjestelmä
2005-6	HAT
2006-7	Mixel
2007-8	Aateliset
2008-9	Usability Team

In Table 3, there are listed awarded projects since 2004. More information

on these projects can be found from earlier course reports.

Course staff thank our clients, visiting lecturers, and students for great projects.

Tampere, August 2009
Timo Poranen

Contents

1 Usability Team

1.1 Overview

This year's usability team (Uteam) consisted of 7 usability experts and 3 project managers. Basic work was done by allocating UI-specialists to different projects and handling their usability related issues. The purpose of usability team spawns from the need of having one dedicated group or project to handle rather different and sometimes complex issues regarding usability. Group also serves as a learning-base for interactive technology students to incorporate different tools and methods they have learned. The first usability team was introduced in the 2005 - 2006 project work course. Since then the project has evolved each year to perform even better and to meet the real needs other projects regarding usability.

Uteam used a distributed organisational structure where usability experts were assigned to work with one or two separate project teams and reported to the usability project manager responsible for that specific customer project. This was done in order to ensure a good level of communication between ui-members and a solid dedication on each customer project. The overall managerial burden was also eased with this solution. The know-how left from the previous uteam's in the form of documents and webpages helped project managers greatly in getting the project up and running straight away. Work balance was monitored and tasks specified via email and on weekly meetings that were held every other week. Also a special workshop procedure were now introduced for the first time where ui-members were able to collectively discuss the usability jobs they were planing to do or had already done in their designated projects. This greatly improved the workload-issues and enriched the designs under development. Version control was used to give a common, centralized place for ui-related documents and to keep them safe from i.e. computer failures.

1.2 Organisation and management

Members of usability team were Nina Juuri, Ari Koivuniemi, Karoliina Käki, Aapo Laitinen, Janne Nyrhinen, Antero Salokangas and Aleksi Turpeinen. Each usability expert had one or more projects at their responsibility:

- Nina Juuri had Vixtory (formerly know as AgileTool) and LiputOn
- Ari Koivuniemi had Majava
- Karoliina Käki had Collikka

- Aapo Laitinen had MindWiki and Green Quest
- Janne Nyrhinen had NovelUI
- Antero Salokangas had PlayfulUI
- Aleksi Turpeinen had EDP

Figure 1: Project Managers: Pasi Paunu, Timo Ingalsuo and Arttu Tamminen.

Figure 2: Project Members: Nina Juuri, Ari Koivuniemi Karoliina Käki and Aapo Laitinen.

The most obvious usability tool for the Uteam was the usability lab itself. But since each project that the usability member worked for had different requirements, not all of them benefited from the usability lab. Another main tool used in the Uteam was the tools for communication. Email was the primary means of communication but since the team required more interaction, workshops and weekly meetings were held to meet this requirement.

Documents were written with OpenOffice and MS Office. Also a wide variety of other tools like Google sketchUp were used when usability members were required to work as a graphic artist.

Figure 3: Project Members: Janne Nyrhinen, Antero Salokangas and Aleksi Turpeinen.

1.3 Methods and tools

Uteam used the following usability tools and methods during the project:

Methods	Amount
User interface plan	7
Workshop	7
Usability analysis	4
Usability testing	4
Heuristic evaluation	3
Peer review	2
Videolog	1
User interface interview	1

Table 4: Used methods.

User interface plan was required from all projects, and most of the projects managed to provide them. For two of the projects there was some difficulties in producing UI plan, mainly because of lack of time. The required information was however produced in form of the prototypes and other desing products.

1.4 Project phases and development model

Uteam project followed customer service model throughout the project. That is, each individual project was considered as a customer, which used resources provided by Uteam. In organizing Uteam internal work, project followed traditional waterflow model when applicable, with the timelines set by the course management.

Milestone	Date	Phase
First meeting	10.10.2008	Planning and organizing
Preliminary analysis	29.10.2008	
Project plan	2.12.2008	Integration and work guidance
Project plan inspection	5.12.2008	
First workshop	18.12.2008	Workshop phase
Testing	23.03-06.04.2009	
Final report	28.04.2009	Project ending
Presentation	6.5.2009	
Final personal report	6.5.2009	

Table 5: Project phases.

Risk management was taken into account in the early phase of project planning. Generally all the major risks were indentified, and only some minor risks were realized. These were unforeseen difficulties in reservations of the usability laboratory, foreseen impact of one project manager trip to abroad, unforeseen issues in confidentiality and some issues in way the communication worked within the group. None of these risks endangered the Uteam continuum, also none of the major risks identified in risk management plan realized.

1.5 What to do better next time

Here are few schemes for future usability team

- The UI-member must be allocated to the designated group before the project worker lists are made public.
- Usability lab can cause timing issues, but with good preparation this can be minimized.
- An option to use TAUCHI personnel to lecture on usability if necessary.
- Maybe a joint project management with two other project managers from different projects, not a dedicated three project managers.

- Enchancing workshop procedures by utilizing e.g. small group working, usage of more sophisticated methods like brainstorming or prototyping and perhaps more unified working procedures.

1.6 Conclusions

Two of the biggest challenges were integration to the existing project groups due to the strong grouping effect, and the usability testing phase at the end of the project. The strong grouping phenomenon in projects generated again a situation where the usability team member were not first accepted as a full member of the project where he/she was allocated. A strong but decisive work aided in overcoming the problem and we hope future project courses take our suggestions into account when assembling the groups. The workload peak towards the end of the course is inevitable on usability team and the collision with KAME-course on the reservation times cannot be avoided in the current timeframe construction. With good planning the effects can only be minimized but not avoided.

Overall the usability team work was a great success. The group members got to use their knowledge and experience to their fullest and the team got to learn and use many tools and methods but also gained a valuable work experience with real projects and real clients. The Uteam members were very happy with the work and are very proud of it.

1.7 Statistics

Below is the total of number of hours each Uteam member spent in total, by classification and combined with Uteam and other projects.

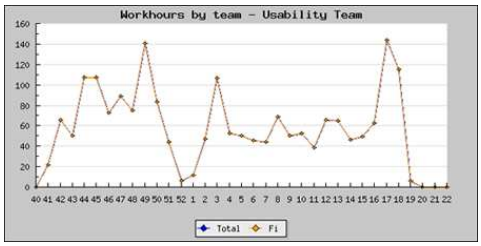

Figure 4: Total hours of Uteam.

The peek in figure 4 at week 49 can be explained by the making of project plan and technical failure with version control server.

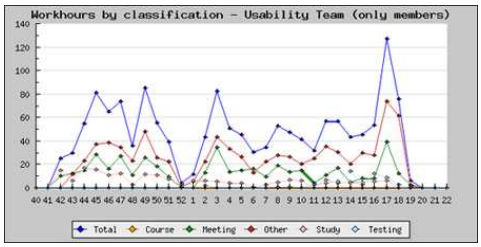

Figure 5: Workhours by classification.

As expected the Uteam has tremendous workload at the end of the course. This can also be seen in Figure 5.

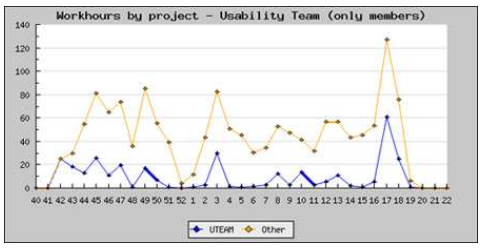

Figure 6: Combined stats.

Members	Starting date	Ending date	Days	Hours	Index
3+7	10.10.2008	06.05.2009	209	1970.4	0.94

Table 6: Overview of project (index is calculated by $(Hours/(Days * Member))$).

Document	Pages	Versions
Preliminary analysis	9	3
Project plan	34	11
Usability plans	73	7
Usability test reports	48	3
Ending report	36	5
UI-plan	162	5
Project story	4	2
Weekly reports	27	N/A
Total	391	

Table 7: Documentation.

2 MindWiki

2.1 Yleistä

Projekti tehtiin asiakkailta saadun tuoteidean pohjalta. Tuoteidea oli graafinen työpöytä, jossa yksittäiset wikidokumentit ovat kuin muistilappuja. Näitä muistilappuja voi lisätä tai poistaa pöydältä, niin järjestystä voi muuttaa ja niiden välille voi luoda nuolia riippuvaisuussuhteita kuvaamaan. Tuoteidea yhdisti wikiohjelmistojen yhteisöllisen tavan ylläpitää dokumentteja ja käsitekarttojen eli niin sanottujen Mind Map -kaavioiden tavan kuvata asioiden välisiä riippuvuussuhteita. Projektissa tuotteesta käytetty nimi MindWiki kuvastaa Mind Map ja Wiki - tekniikoista johdettuja ideoita.

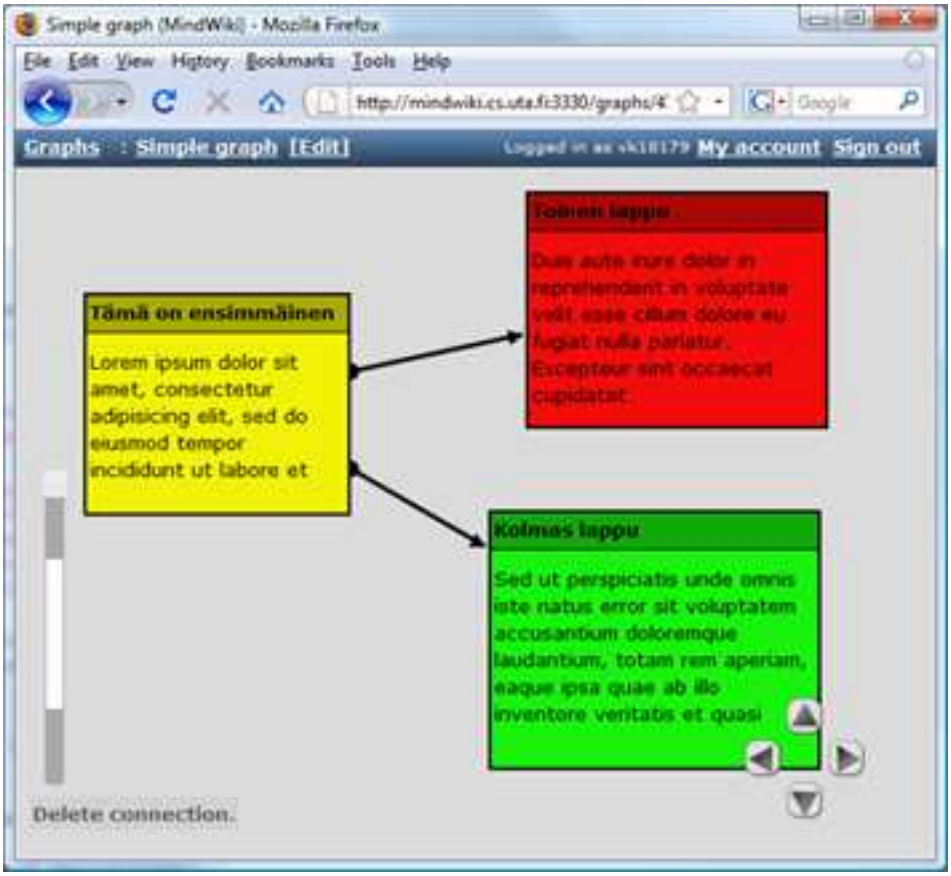

Figure 7: Ruudunkaappaus MindWiki ohjelmasta

MindWikin koodi on julkaistu MIT-lisenssillä, ja se on saatavissa jatkokehitystä varten osoitteesta `http://github.com/muhku/mindwiki/tree/master`

2.2 Organisaatio ja johtaminen

Projektiryhmä koostui neljästä projektipäälliköstä ja viidestä projektityöntekijästä. Lisäksi projektin käyttöön on nimetty yksi käytettävyysryhmän henkilö. Kaikki projektin henkilöt ovat Tampereen yliopiston opiskelijoita. Projektipäälliköt suorittivat ohjelmistoprojektin johtaminen -kurssia:

- Sami Blommendahl
- Ville Kivelä
- Anssi Männistö
- Lauri Renko

Työntekijät olivat projektityö -kurssilla olevia opiskelijoita.

- Mika Hannula
- Samu Ollila
- Jukka Peltomäki
- Aapo Tahkola
- Juhani Tamminen

Projektin käytettävyysasiantuntijana toimi Aapo Laitinen. Projektin toimeksiantajina olivat Matias Piipari Cambridgen yliopiston Sanger-instituutista ja Matias Muhonen Tampereen yliopiston tietojenkäsittelytieteiden laitokselta. Projektin ohjauksesta vastasi Timo Poranen.

2.3 Menetelmät ja työkalut

Sovelluksen runko toteutettiin Ruby-kieleen perustuvaa Ruby On Rails -ohjelmistokehystä käyttäen. Rubylla toteutetaan palvelimen puolelle muun muassa sovelluksen datan käsittely ja tietokantaoperaatiot. Varsinainen käyttöliittymä toteutettiin JavaScriptillä AJAX-tekniikkaa käyttäen. Tässä hyödynnettiin JavaScript-kirjastoa jQuery, joka saadaan käyttöön jRails- kirjaston avulla.

Ohjelmiston tietokantana on MySQL. Versionhallintajärjestelmänä toimii Subversion. Kun tehdyt muutokset päivitettiin versionhallintaan, uusi ajettava versio tuotettiin automatisoidusti käyttäen työkalua CruiseControl.rb Sovelluksen kehitystä ja testausta varten oli käytössä Tampereen yliopiston projektin käyttöön antama virtuaalipalvelin.

Figure 8: Projektin henkilöstö.

Projektinhallintajärjestelmänä käytetään Redminea, joka pystytettiin projektia varten asiakkaan toimesta. Projektin dokumentit julkaistiin projektin hallintajärjestelmän tarjoamassa wikissä. Lisäksi joitain tärkeimpiä dokumentteja laadittiin Microsoft Word-tekstinkäsittelyohjelmalla ja julkaistiin pdf-formaatissa.

Koska toinen asiakas asuu ja työskentelee ulkomailla, projektikokouksissa ja katselmoinneissa käytettiin Skypeä etäosallistumisen mahdollistamiseksi.

2.4 Projektin kehitysmalli ja vaiheet

Projektissa käytettiin ketterän ohjelmistokehitysmenetelmän periaatteita. Projektiryhmässä katsottiin, että jonkun tietyn kehitysmenetelmän käyttämistä sellaisenaan ei ole mahdollista, joten käytännössä projektissa sovelletaan jotain ketterää menetelmää, kuten Scrumia, vain niiltä osin kuin hyväksi nähdään. Projektissa edettiin joka tapauksessa ketterien menetelmien periaatteita noudattaen. Ketterä ohjelmistokehitys valittiin lähinnä asiakkaan toiveesta jo-

htuen. Ketteryys sopi varsin hyvin projektiin muun muassa siitä syystä, että tämän kaltaista järjestelmää ei aikaisemmin ole toteutettu. Siten sovelluksen kehittäminen lyhyissä iteraatioissa oli luontevaa. Kunkin iteraation pituus oli 3 viikkoa poikkeuksina iteraatiot 3 ja 6. Iteraatioiden katselmoinnit järjestettiin pääsääntöisesti kunkin iteraation loppupäivämäärää seuraavalla viikolla.

2.5 Johtopäätökset

Projektin aihe oli mielenkiintoinen, motivoiva ja riittävän haastava. Projektia voidaan pitää onnistuneena, koska se saavutti sille asetut tavoitteet, yksittäisiä vähäisiä puutteita lukuunottamatta. Vaikka projekti onnistuikin, se myös osoitti siihen osallistuneille henkilöille, että monta projektinhallinnan ja vaatimusten määrittelyn asiaa voisi tehdä paremminkin.

2.6 Tilastot

Tässä on esitetty tärkeimmät tilastot MindWiki-projektin ajankäytöstä ja tuotoksista

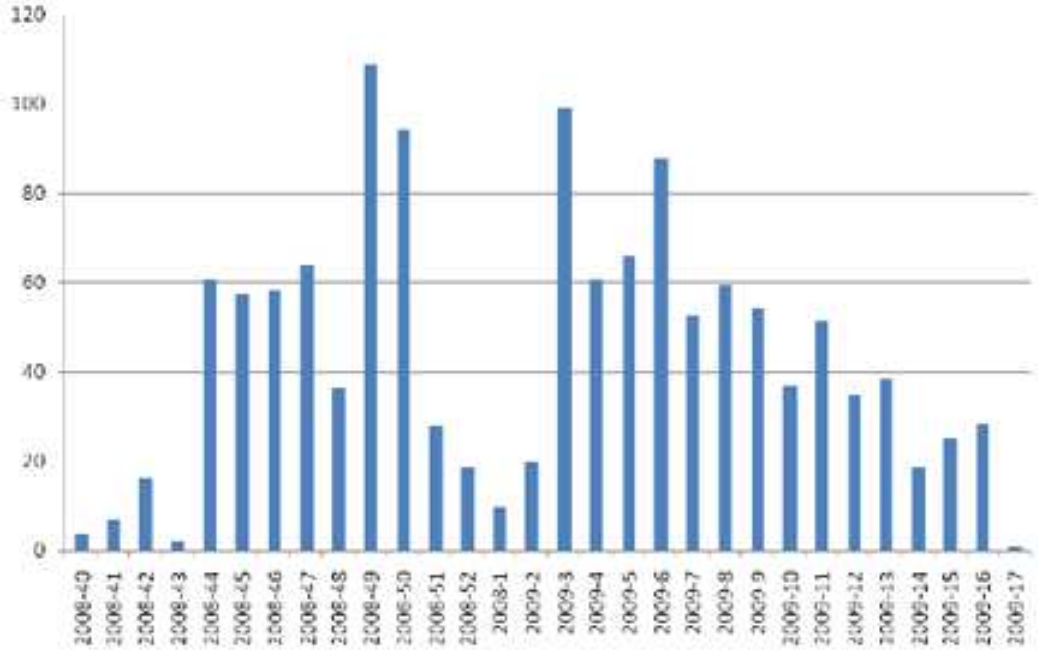

Figure 9: Projektin viikkotunnit.

Ryhmä koko	Kehitys malli	Aloitus pvm	Lopetus pvm	Päiviä	Tunteja	Tunteja/ (pv* ryh mäkoko)
5+4+1	Scrum	27.10.2008	28.4.2009	120	1319	1.10

Table 8: Projektin yhteenveto.

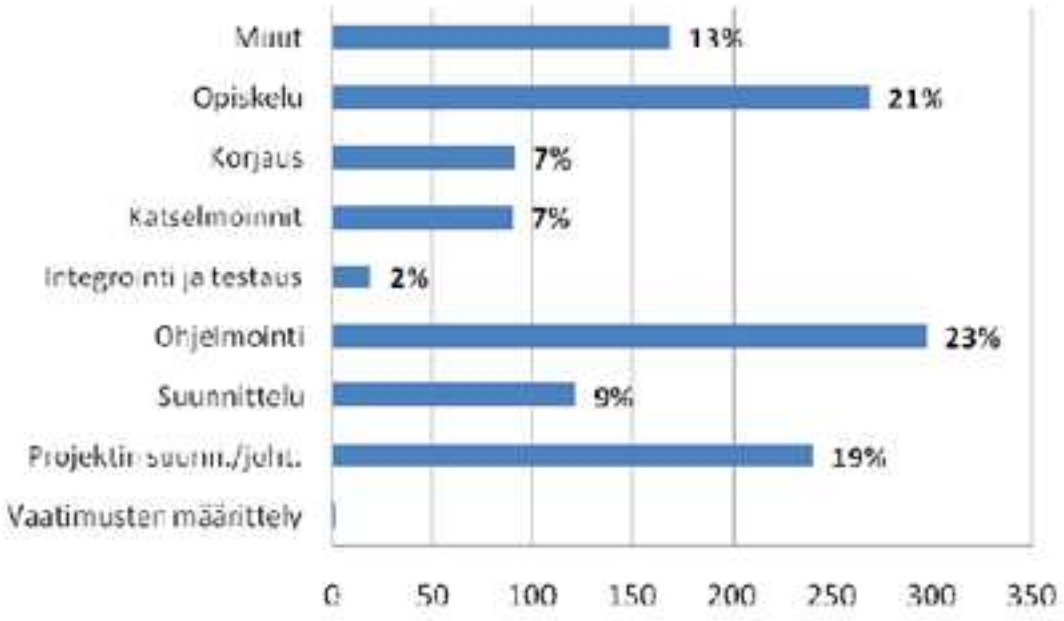

Figure 10: Projektityön jakautuminen.

LOC	3845
SLOC	2558

Table 9: Projektin koodirivit.

Dokumentti	Sivuja	versioita
Esitutkimus	1	8
Projektisuunnitelma	3	21
Vaatimusmäärittely	1	2
Testausraportti	1	4
Käytettävyystest. rap.	2	10
Loppuraportti	3	26
Projektitarina	1	4
Viikkoraportti	27	1
Katselmointipöytäkirjat	7	2
Yhteensä	46	68

Table 10: Projektin dokumentit.

3 EDP - Electronic Dictionary Project

3.1 Projektin kuvaus

Projektissa luotiin www-pohjainen työkalu kieli- ja käännöstieteiden laitoksen lehtori Juhani Norrin tutkimusmateriaalia sisältävään tietokantaan. Lehtori Norrin tietokanta sisältää noin 12 000 lääketieteellistä termiä, joiden esiintymistä on kartoitettu vuosilta 1375–1550 peräisin olevassa kirjallisuudessa. Termien ja niiden ilmiasujen(varianttien) lisäksi tietokanta sisältää lainauksia alkuperäislähteistä, joissa termejä on eri muodoissaan esiintynyt, kirjallisen kontekstin valossa muotoiltuja termien merkityksiä, sekä viitetietoja niiden termien osalta, joiden esiintymistä on aiemmissa julkaisuissa tutkittu. Käyttöliittymän toteutuksen lisäksi tietokantarakenne laitettiin osittain uusiksi, kuitenkin vanhat tiedot säilyttäen. Vanhaan tietokantarakenteeseen oli pala palalta tuotu uusia ominaisuuksia, joten kokonaisuus täytyi järkevöittää mm. tehokkuuden saavuttamiseksi.

3.2 Projektiorganisaatio

Projektiin osallistui neljä manageria, viisi varsinaista ryhmän jäsentä ja yksi käytettävyysasiantuntija. Projektimanagereina toimivat Ville-Veikko Kalkkila, Kimmo Rinteelä, Antti Toivanen ja Risto Välimäki. Antti toimi yleismanagerina ja hoiti yhteydet kurssin vetäjään, Kimmo johti suunnitteluryhmää ja hoiti yhteydet asiakkaaseen. Ville-Veikko ja Risto toimivat toteutusryhmän managereina.

Ryhmän jäsenet jaettiin kahteen ryhmään: Suunnittelu- ja testausryhmässä toimivat Jussi Oksanen, Piia Sajasalo sekä käytettävyysasiantuntija Aleksi Turpeinen. Toteutusryhmässä toimivat Matti Lassila, Anne Mikkonen ja Timo Sirainen. Suunnittelu- ja testausryhmä istui säännöllisesti palaverissa

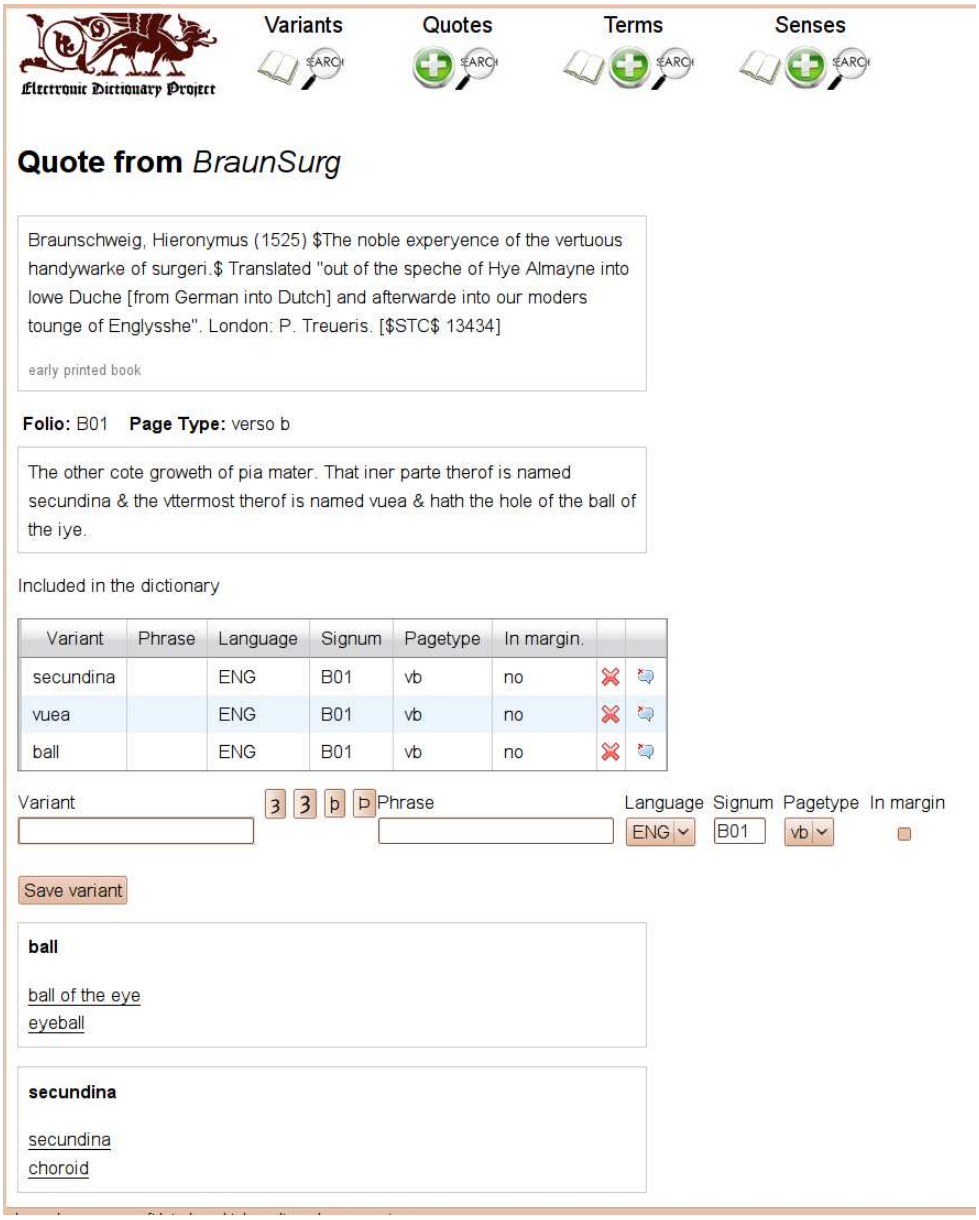

Figure 11: Esimerkkikuvankaappaus: Lainausnäyttö.

asiakkaan kanssa, ja keräsi vaatimusmäärittelyjä. Vaatimusmäärittelyt käännettiin käyttäjätarinoiksi ja niihin liittyviksi testeiksi. Tämä ryhmä hoiti myös pääosan testauksista. Toteutusryhmässä Matti ja Timo toimivat ohjelmoijina, ja Anne hoiti pääosin tietokantamuutoksia.

Projektiryhmän keskinäinen yhteydenpito tapahtui lähes jokaviikkoisten maanantaipalaverien, IRC-kanavan, Wikin ja sähköpostin kautta. Erityisesti

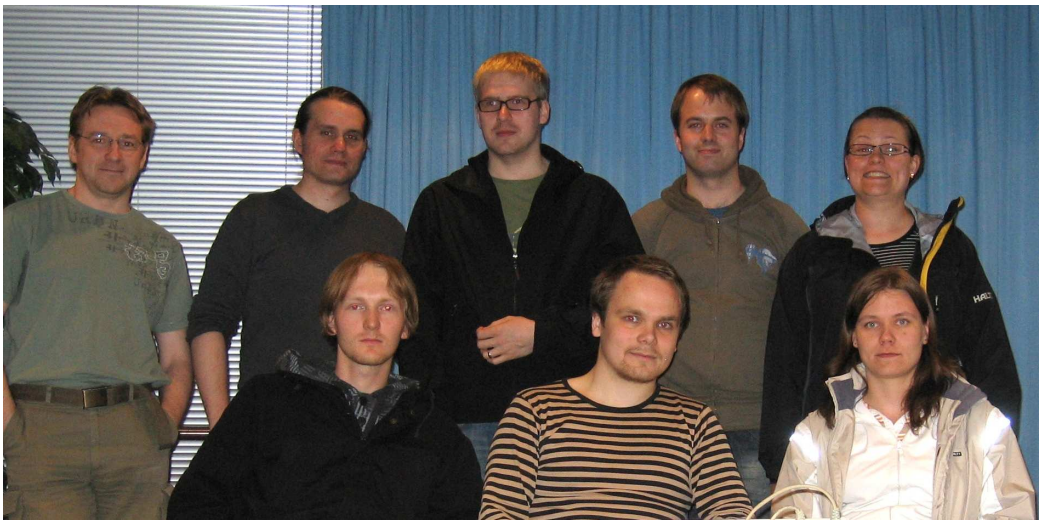

Figure 12: Ylärivissä: Kimmo, Ville-Veikko, Risto, Antti, Piia. Alarivissä: Aleksi, Matti, Anne. Kuvasta puuttuvat Jussi ja Timo.

toteutusryhmälle IRC muodostui erittäin tärkeäksi työkaluksi, varsinkin kun Joulun jälkeen toteutusryhmästä Timo lähti Amerikkaan, eikä näin voinut osallistua viikkotapaamisiin.

3.3 Menetelmät ja työkalut

Sovellus rakennettiin CakePHP-ohjelmakehyksen ympärille. CakePHP pitäytyy tiukasti MVC-mallissa ja on suunniteltu erityisesti ketterään ohjelmakehitykseen. Täysin ongelmatonta CakePHP:n toimintamallin omaksuminen ei ollut, mutta on selvää, että CakePHP:n käyttö selvästi nopeutti sovelluksemme kehitystä, ja loi samalla selkeät raamit ohjelmoinnille. Mikäli luomaamme työkalua jatkokehitetään, CakePHP-taitoinen pääsee hyvin nopeasti selville ohjelman rakenteesta, joten tässäkin mielessä CakePHP oli erinomainen valinta

Versionhallintaan käytettiin Subversionia. Jokainen versionhallinan päivitys vietiin automaattisesti kehityspalvelimelle ajoon. Projektin-, dokumenttien ja tehtävienhallintaan käytettiin Trac-järjestelmää, joka tarjoaa erinomaisen Wikin, tehtävienhallinnan ja versionhallinnan integraation. Tracin Wiki ja lähdekoodin seurantatyökalut olivat ahkerassa käytössä ja todettiin erinomaisiksi. Sen sijaan tehtävienhallinnan käytettävyydessä olisi mielestämme jonkin verran parannettavaa, vaikka se toki tällaisenaankin toi meille lisäarvoa.

Kehityspalvelimena meillä toimi laitoksen virtuaalipalvelin, jossa pyöriteltiin Tracia, Subversionia ja kehitysympäristöä, eli Apachea ja PostgreSQL-palvelinta. Virtuaalipalvelin toimi projektin alkuvaiheessa kohtuullisesti, mutta kurssin edetessä palvelinkatkokset ja hidastelu oli enemmän sääntö kuin poikkeus.

Ohjelmointityökaluina käytettiin lähinnä tekstieditoreja, mutta myös Eclipsen PHP-kehitysympäristöä (PDT).

3.4 Projektin vaiheet ja kehitysmalli

Projektin kehitysmallina käytettiin sovellettua Scrumia, jossa ohjelmankehitys jaettiin useisiin iteraatioihin. Varsinkin projektin alkuvaiheessa kehitysmalli oli hakusessa, mutta muotoutui talven mittaan kohtalaisen toimivaksi kokonaisuudeksi. Iterointia olisi voinut olla paljon enemmän, mutta toisaalta tällöin myös ohjelmointiresursseja olisi tarvittu enemmän. Tällöin olisi ollut mahdollista tuottaa enemmän toimivia prototyyppejä, joita olisi sitten asiakaspalaverien tuloksena muokattu, kun nyt käytettiin enemmän piirrettyjä prototyyppejä, ja varsinaisen ohjelman iteroiminen jäi vähemmälle.

Projekti toteutettiin kaikkiaan 7 vaiheessa, joista toteutusiteraatioita oli 5. Ennen toteutusta oli suunnittelujakso, jossa muodostettiin ryhmä, luotiin toteutuksen karkeat suuntaviivat ja valittiin toteutustekniikka. Viidennen toteutusiteraation jälkeen viimeisteltiin loppuraportointi, tehtiin järjestelmätestausta ja korjattiin bugeja. Toteutusiteraatioiden sisältö koostui kolmesta päätoiminnosta: Vaatimusten kerääminen, vaatimusten toteuttaminen ja testaus. Testauksen raportointi toteutuksen raportoinnin ja tuntiraportoinnin kanssa muodosti iteraation dokumentoinnin. Iteraatioiden jälkeen dokumentointi ja toteutetut sovelluksen osat katselmoitiin asiakkaan kanssa.

Iteraatioiden läpivienti perustuu hyvin pitkälle sanakirjatutkimustietokannan rakenteeseen. Rakenne pyrkii kuvaamaan sanakirjaan kuvattavia käsitekokonaisuuden osia, joten toteuttaminen tietokannan rakenteen mukaisessa järjestyksessä tarkoitti myös sanakirjatutkimuksen työnkulun osien mukaista ohjelmointitoteutusta. Valittu tapa tuki täten myös kommunikointia asiakkaan kanssa. Asiakkaalta saatiin kokemukseen perustuvaa palautetta työn tekemiseen vaikuttavista seikoista: Asiakas pystyi vertailemaan tehtävän sovelluksen toiminnallisuutta entisiin työtapoihinsa ja kykeni sitä kautta arvioimaan myös toteutuksesta saatavia hyötyjä käytännön tasolla. Projektin viisi toteutusiteraatiota (I-V) ajoittuivat päivien 10.11 ja 27.4. välille. Toteutuksen aikana pidettiin lomaa 22.12 – 5.1. Tuona aikana viikkotapaamisia ei järjestetty laisinkaan.

Vaihe	Kuvaus	Lopetus pvm
Iteraatio 0	Esitutkimus	10.11.08
Iteraatio I	Lainausosa (Quotes) ja tietokanta	10.12.08
Iteraatio II	Sanojen ilmiasut (Variants)	21.01.09
Iteraatio III	Termit ja viitetiedot (Terms, References)	25.02.09
Iteraatio IV	Merkitysosa (Senses)	08.04.09
Iteraatio V	Integraatio	27.04.09
Iteraatio VI	Raportointi ja korjaus	07.05.09

Table 11: Projektin vaiheet.

Table 12: General project information.

Team size	Dev. mod.	Start date	End date	Days	Hours	Hours / (Days * Team size)
4+5+1	Scrum	6.10.2008	7.5.2009	200	1531	0.77

3.5 Johtopäätökset

Projekti oli ryhmälle haaste, josta selvittiin hyvin. Suurta eripuraa ei ryhmän sisällä ollut missään vaiheessa, vaikka töiden jakaminen ei projektin johdolta aina sujunut niin tasaisesti kuin olisi voinut toivoa. Ryhmällä oli hyvä henki ja asioita hoidettiin tarmokkaasti silloin, kun tehtävät sitä vaativat.

Vaikka työt saatiinkin sujumaan ongelmitta, voisi monessa asiassa parantaa. Yksi tärkeimmistä on projektin kehitysmallin orjallinen noudattaminen. Joskus tämä saattaa tarkoittaa joustoista luopumista, mutta kehitysmallien avulla ryhmän sisäinen dynamiikka ja kommunikointi, toisin sanoen työpanoksen suuntaaminen onnistuu hallitummin ja varmemmin.

Toinen huomattava epäkohta on toteutuksen lipsuminen liian harvojen käsiin. Ohjelmointiympäristön ymmärtäminen ja sen tarjoamien mahdollisuuksien tarkasteleminen olisi tullut toteuttaa yhteisemmin. Kun toteutus eteni ja toteutettu sovellus monimutkaistui, tuli sen sisäistämisestä yhä haastavampaa. Vaikka päteviä tekijöitä olisikin ollut tarjolla, ei toteutuksesta perillä olleilla ollut enää tarmoa ja aikaa opastaa uusia ihmisiä avukseen projektin loppupuolella. Tilanne olisi voitu välttää kouluttamalla ryhmäläisiä laajemmin projektin alkupuolella.

3.6 Statistics

Table 13: Group effort by activity.

Activity	Plan. and man.	Req. spec.	Design	Code	Integ. and test	Reviews	Repair	Study	Other	Total
Hours	583	150	66	214	69	45	39	139	89	1394
%	42	11	5	15	5	3	3	10	6	100
Usability	16	15	50	0	8	8	6	18	16	131
Total	599	165	116	214	77	53	45	157	105	**1531**

Table 14: Requirements and high-level design outcome.

Requirements (User Stories)	Use-cases (Tests)	UI screens	Database diagrams	Database tables
49	82	40	1	23

Table 15: Project's documents.

Document	Pages	versions
Preliminary analysis	14	2
Project plan	26	5
Project's usability plan	42	1
Requirements specification	30	(Trac)
Test plan	40	(Trac)
User's guide	5	1
Installation guide	10	2
Test report	20	(Trac)
Usability test report	12	1
UI Heuristic evaluation report	5	
Final report	36	2
Final story	6	1
Weekly reports	24	
Inspection reports	4	
Total	264	

Table 16: Project's codelines.

Language	PHP	JavaScript	Views	SQL	TOTAL
LOC	5975	1171	3490	692	11328
Reused code	62766	128083	2404	171	193424
Code revisions	690				

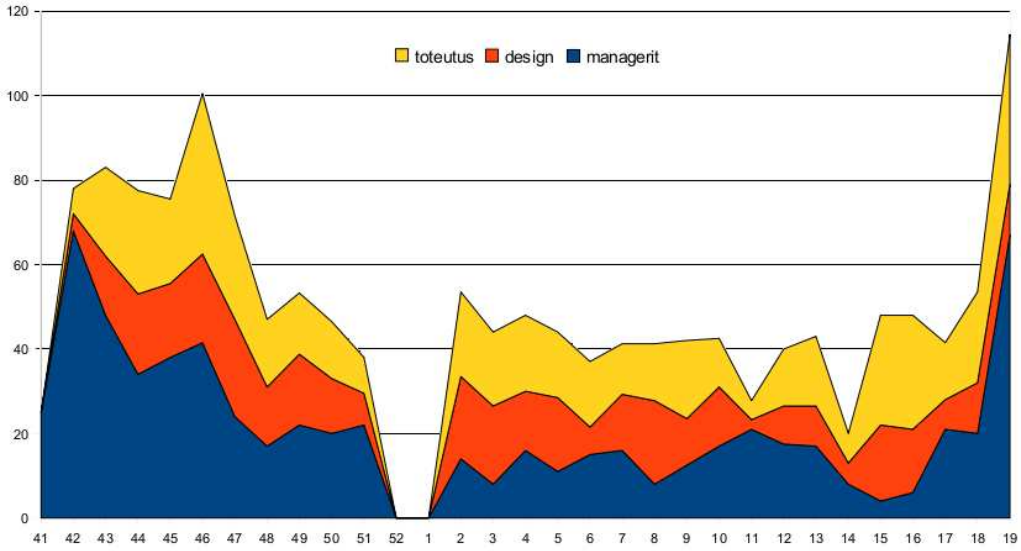

Figure 13: Projektin viikottainen työmäärä ryhmittäin.

Table 17: Productivity metrics.

PM	LOC / PM	LOC pages / PM	Total pages / PM
10,07	1125	22,5	48,7

4 Majava

4.1 Overview

Our team implemented a web based training system for the Beaver international informatics and computer literacy contest (for more information on the Beaver contest, please see `http://www.bebras.org/en/welcome`). The Majava system provides several sets of questions, divided into classes depending on the age group of the user. Majava contains question sets based on the Beaver competition questions from years 2007 and 2008. The user is presented with a set of questions based on his age and year choices. Majava gives feedback after the the session is over. An administrative interface is also included, making it possible to add and modify questions, years, agegroups and other data. The main view for answering questions is shown in Figure 14.

4.2 Organisation and management

Our customer was the Tampere University Computer Science Department through their representative and contact person Timo Poranen.

The team had four project managers: Iikku Mattila, Lauri Tuominen, Olli Ruotsalainen and Petri Molkkari. Project staff consisted of Kimmo Röppänen, Lauri Vilkki, Maiju Karhunen, Pasi Lampinen, Piia Naukkarinen, along with Ari Koivuniemi as our usability expert from the usability team. Figure 15 shows members of the team.

Work was divided roughly into five areas of development, testing, documentation, content creation and management. All team members participated in documentation activities. Project managers worked also on documentation and content creation tasks in addition to management. Project staff concentrated their efforts either on development or content creation, depending on their skills and areas of interest. There was no formal division of tasks however, so team members worked in different areas from time to time.

4.3 Methods and tools

Ruby on Rails was chosen as the application framework, thus Ruby was our development language. The architecture pattern used in Ruby on Rails is Model-View-Controller. The following tools were used:

- Programming language: Ruby

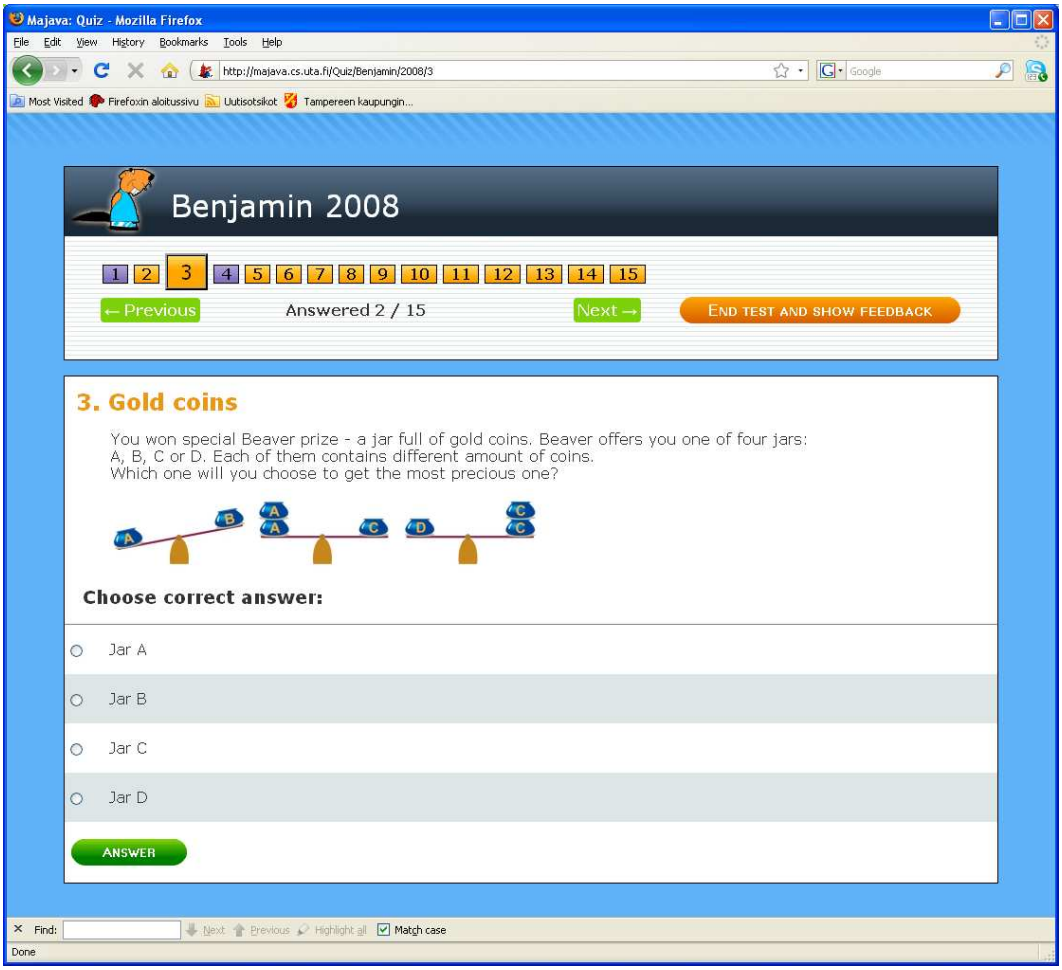

Figure 14: Majava quiz screen.

- Application framework: Ruby on Rails
- IDE: Eclipse
- Version control: SVN, Tortoise, Subclipse
- Documentation: Open Office Writer
- Project wiki, document distribution and storage: TikiWiki
- Testing environment: Firefox 3.0.4
- Chat: IRC

Figure 15: Majava team: Pasi Lampinen, Ari Koivuniemi, Piia Naukkarinen, Lauri Tuominen, Maiju Karhunen, Iikku Mattila, Petri Molkkari and Olli Ruotsalainen. Kimmo Röppänen and Lauri Vilkki are not shown.

- Notices and discussion: Mailing list

4.4 Project phases and development model

Majava was developed in increments. The first so-called zero increment consisted of gathering the team, requirements analysis with the customer and choice and setup of development tools. There were five development increments, the last of which did not include any new functionality, only minor changes and fixes.

The project proceeded according to plan. Some proposed features were dropped early in the project. The incremental approach helped us to complete tasks early, thus there was no huge crunch at the end and we could concentrate on fixing issues and making minor UI changes.

The scope of the project changed somewhat after the first working versions of the software were presented to the customer. In addition to developing the software, our team was tasked with content creation. This consisted

of translating and entering questions from various past Beaver contest into the Majava system. The team also had to make additions and corrections to some of the content.

4.5 Experiences

The course provided us with end-to-end experience on working in a software project. For many team members this was the first such experience. Typical software project challenges were encountered, such as changing and unclear requirements, regression issues, technical problems and difficulties in communication. The team also learned how to overcome these issues.

Team members had work and studying related time issues. Luckily our team was able to cope with these, as the issues were not encountered all at the same time. The project required a lot of work and constant attention. With a large team weekly meetings with everyone attending were also not trivial to organize.

We learned that tool setup is very important. Although technically our tools were chosen and set up during the zero increment, in practice all members did not have a working development environment at that point. It would have been well worth the extra effort to ensure that everyone has a working environment at the start of development work.

4.6 Statistics

Table 18: General project information.

Team size	Dev. mod.	Start date	End date	Days	Hours	Hours / (Days * Team size)
5+4+1	Incremental	6.10.2008	10.5.2009	216	1100	0.51

Table 19: Project's documents.

Document	Pages	versions
Preliminary analysis	13	2
Project plan	45	7
Requirements specification	42	11
Design plan	17	3
User interface document	19	6
Test plan	27	6
Usability test plan	15	3
Usability test report	19	2
Usability test guide	12	4
Final report	25	8
Weekly reports	30	
Total	264	

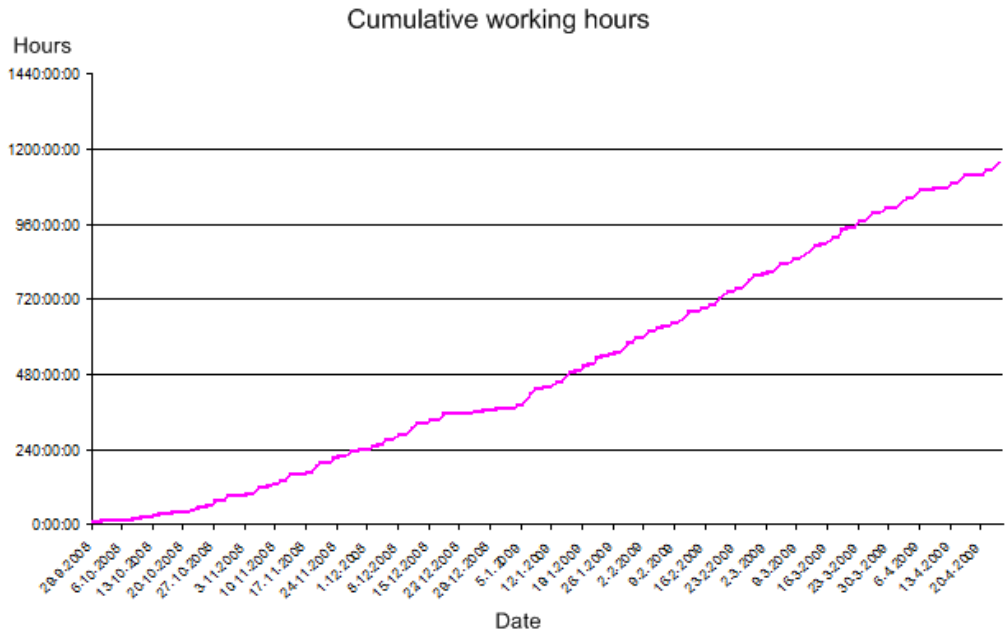

Figure 16: Weekly workload of the project.

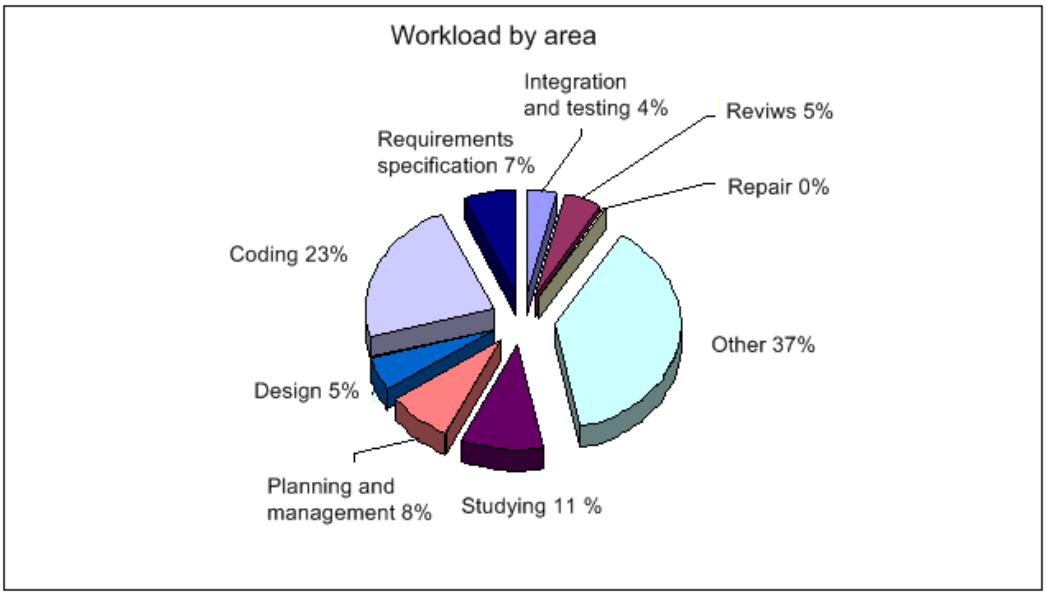

Figure 17: Workload of different aspects of the project.

5 Collikka

5.1 Yleistä

Collikka-projektin tehtävänä oli toteuttaa Suomen Collieyhdistys ry:n internetsivulle pitkä- ja lyhytkarvaisten collieiden terveystietokanta, joka sisältää tiedot jalostustoimikunnan tiedossa olevista sairaista tai lonkkavikaisista koirista. Collikka Collieiden terveystietokannan tarkoituksena on auttaa koirien kasvattajia ja muita koiraharrastajia rodun terveystilanteen ja -tietouden parantamisessa. Collikka on julkaistu 8.5.2009 osoitteessa:
`http://collikka.collieyhdistys.fi`.

Figure 18: Collikka collieiden terveystietokannan etusivu.

Suomen Collieyhdistys on tällä tavoin astunut kohti avoimempaa suhtautumista ja collieiden terveystiedot ovat kaikkien kasvattajien saatavilla.

Koiran omistajat voivat internetin kautta ilmoittaa Collikkaan oman koiransa sairaustiedot tai muuttaa koiran tietoja tai lisätä sinne uuden koiran. Jalostustoimikunta tarkistaa ilmoitetut tiedot ja sen jälkeen ne näkyvät julkisesti Collikassa. Collikkaan päivitetään myös automaattisesti tietyin väliajoin Suomen Kennelliiton taltioimat koirien perustiedot.

Projektin asiakkaana oli Suomen Collieyhdistys ry:n Jalostustoimikunta, jonka puheenjohtajana toimii Suvi Hirvonen.

5.2 Projektiorganisaatio

Projektiryhmään kuului 11 henkilöä. Projektipäälliköitä oli neljä ja projektityöntekijöitä oli seitsemän, josta yksi oli käytettävyysryhmän henkilö.

Projektipäällikköinä toimivat:

- Harri Heinisuo
- Pia Järvinen-Hiekkanen
- Leena Palovuori
- Matti Virtanen

Projektityöntekijöitä olivat:

- Jaakko Helenius
- Sari Kurimo
- Elvis Okemu
- Jonna Paananen
- Jaana Partanen
- Tiina Vainio

Käytettäyysryhmästä mukana oli:

- Karoliina Käki

Figure 19: Projektin henkilöstö.

Projektinjohtajista Harrin vastuualueena oli tekninen toteutus, Matin vastuualue oli scrum ja asiakasyhteydet, Pia vastasi dokumentoinnista, aikataulutuksesta ja kustannusarvioista ja Leena vastasi dokumentoinnista, viikkoraporteista ja yhteydenpidosta kurssin vastuuhenkilöön Timo Poraseen.

Projektilaisista Jaakon ja Jaanan vastuualueena oli koodaus ja Jaana vastasi lisäsi tietokannasta. Jonna Collieyhdistyksen jäsenenä tunsi parhaiten asiasisällön ja hänen vastuualueenaan oli myös käyttöiittymä. Sari ja Tiina vastasivat myös käyttöiittymästä ja Sari vastasi lisäksi järjestelmän testauksesta. Elvis laati käyttöhjeet ja oli mukana ulkoasusuunnittelussa. Karoliinan vastuualueena oli käytettävyystestaus.

5.3 Menetelmät ja käytetyt ohjelmistot

Projektiryhmä sai vapaasti valita käytettevät ohjelmistot ja tekniikat. Ohjelmoinnissa käytettiin Tietojenkäsittelytieteiden laitoksen ohjelmistoja:

- PostgreSQL 8.1.2
- PHP 5.2.0
- Javascript

Versionhallintana käytössä oli Tietojenkäsittelytieteiden laitoksen Subversion 1.4.6 -palvelin. Ohjelmina kätettiin TortoiseSVN ja Ubuntulle RapidSVN-ohjelmia. Projektin kotisivuna oli käytössä Mediawiki 1.13 ja Lockdown-laajennus. Selaimina oli käytössä Mozilla Firefox, Windows Internet Explorer, Google Crome ja Apple Safari.

Projektin dokumentit kirjoitettiin ja tallennettiin OpenOfficella sekä lopulliset versiot tallennettiin pdf-muotoon. Lisäksi käytettiin ohjelmia Macromedia Dreamweaver 8, Star UML, Adobe Photoshop, Visual Paradigm, Microsoft Office Visio ja Notepad++.

Kehitysmallina käytettiin Agile-menetelmiin kuuluvaa Scrumia. Scrumia pidettiin sopivimpana ja joustavimpana kehitysmallina. Projektin vaiheet jaettiin 6 sprinttiin. Muun opiskelun ja työssäkäyntien vuoksi emme voineet työskennellä päivittäin vaan enimmäkseen jokainen työskenteli kotona ja kerran viikossa kokoonnuimme yhteen keskustelemaan projektin vaiheista. Kotona työskennelless' yhteyttä pidettiin aluksi sähköpostilla. Se osoittautui kuitenkin vähän hankalaksi kommunikointi välineeksi. Projektin jäsenet sopivat keskenään yhteisistä työskentelyajoista ja silloin yhteyttä pidettiin ircillä. Käytössä oli myös Collikan keskustelufoorumi. Sprinttien lopussa suoritettiin katselmoinnit ja demotapaamiset asiakkaan kanssa.

5.4 Projektin eteneminen

Projektiryhmän jäsenten tehtävät jakaantuivat luontevasti kunkin opintojen maisteriohjelmien mukaisesti. Alkuvaiheessa projektin asiakas oli mukana useammassa palaverissa, jolloin saimme lisätietoja terveystietokannan vaatimuksista ja keskustelimme projektin kehitysvaiheista.

Projektiryhmä kokoontui viikoittain laitoksen projektityöhuoneessa lukuunottamatta joululomaa ja perioditaukoja. Palavereissa keskusteltiin ja käytiin läpi viikoittain projektin vaiheita ja suunniteltiin yhdessä projektin toteutusta. Viikoittain palavereissa yhdessä haettiin ratkaisuja esilläoleviin vaikeimpiin tilanteisiin.

Projektin tärkeimmät vaiheet:

- 08.10.2008: Projektin aloitus. Projektin johtajien ensimmäinen tapaaminen.
- 14.10.2008: Projektiryhmän ensimmäinen tapaaminen.
- 29.10.2008: Esitutkimuksen katselmointi.
- 05.11.2008: Ensimmäinen asiakastapaaminen.
- 19.11.2008: Projektisuunnitelman katselmointi.
- 26.11.2008: Ensimmäinen asiakasdemo ja 1.sprintin katselmointi.
- 10.12.2008: Projektin esittely ja asiakastapaaminen.
- 14.01.2009: Asiakasdemo ja 2.sprintin katselmointi.
- 11.02.2009: Asiakasdemo ja 3.sprintin katselmointi.
- 18.03.2009: Asiakasdemo ja 4.sprintin katselmointi.
- 22.04.2009: Loppukatselmointi ja asiakastapaaminen.
- 29.04.2009: Lopputapaaminen kurssin vetäjän kanssa.
- 06.05.2009: Projektin esittely ja projektiryhmän lopetustilaisuus asiakkaan kanssa.

5.5 Johtopäätökset

Monelle projektin jäsenelle tämä oli ensimmäinen oikea ohjelmistoalan projektityö. Projekti opetti monia uusia asioita ja antoi kokemusta tiimityöskentelyyn. Projektiryhmä tuli hyvin toimeen keskenään ja siinä oli koko ajan hyvä ja iloinen työilmapiiri. Projekti onnistui hyvin. Asiakas oli tyytyväinen lopputulokseen ja hän on ilmaissut tyytyväisyytensä moneen otteeseen.

Projektin kehitystä hidasti ja vaikeutti pitkälliset neuvottelut Suomen Kennelliiton kanssa. Tarkoituksena oli saada suora yhteys KoiraNetin tietokantaan. Siinä emme onnistuneet, mutta sovimme, että Collikka saa säännöllisesti päivitykset KoiraNetin tietokannasta.

Alkuperäisenä suunnitelmana oli tehdä myös uroskansio, jossa on tietoja jalostukseen käytettävistä koirista. Jouduimme kuitenkin luopumaan siitä aikapulan vuoksi. Projektin loppuvaiheessa tuli myös esille monia jatkokehitysehdotuksia, jotka on kirjattu loppuraporttiin.

5.6 Tilastot

Tärkeimpiä tilastoja Collikka-projektista.

Table 20: Projektin yhteenveto.

Ryhmän koko	Kehitys-malli.	Aloitus pvm	Lopetus pvm	Päivät	Tunnit	Tunnit / (Päiviä*Ryhmä)
4+6+1	Scrum	8.10.2008	6.5.2009	211	2093,5	0.90

Table 21: Tunnit aihealueittain

Kate-goria	Proj. suun.ja joht.	Vaa-tim määr.	Suun-nit-telu	Oh-jel-moin-ti	Integ-rointi ja tes-taus	Kat-sel-moin-nit	Korj.	Opisk.	Muut	Yht.
Tunnit	691.5	70	210.5	354	164.5	87	105.5	174	236.5	2093.5
%	33	3.3	10.1	16.9	7.9	4.2	5	8.3	11.3	100
Käytet.										172
Yht.										**2265.5**

Table 22: Vaatimukset ja korkean tason suunnitelmat.

Sivuja	Vaatimuksia	Käyttöliittymmän näkymiä	Tietokanta-kaavioita	Tietokanta-tauluja
30	28	25	1	8

Table 23: Katselmointilöydökset.

	Projektisuunnitelma
Sivuja	32
Valmistautumisaika	1155
Katselmointiaika	850
Löydöksiä	42
Käytetty aika/löydös	47,7

Table 24: Projektin dokumentit.

Dokumentti	Sivuja	Versioita
Esitutkimus	15	3
Projektisuunnitelma	34	9
Vaatimusmäärittely	30	13
Suunnitteludokumentti	7	4
Käytettävyystestaussuunnitelma	24	4
Käytettävyystestausraportti	27	5
Käyttöohjeet	28	1
Loppuraportti	41	12
Projektikertomus	6	1
Viikkoraportit	29	
Katselmointiraportit	14	7
Yhteensä	255	

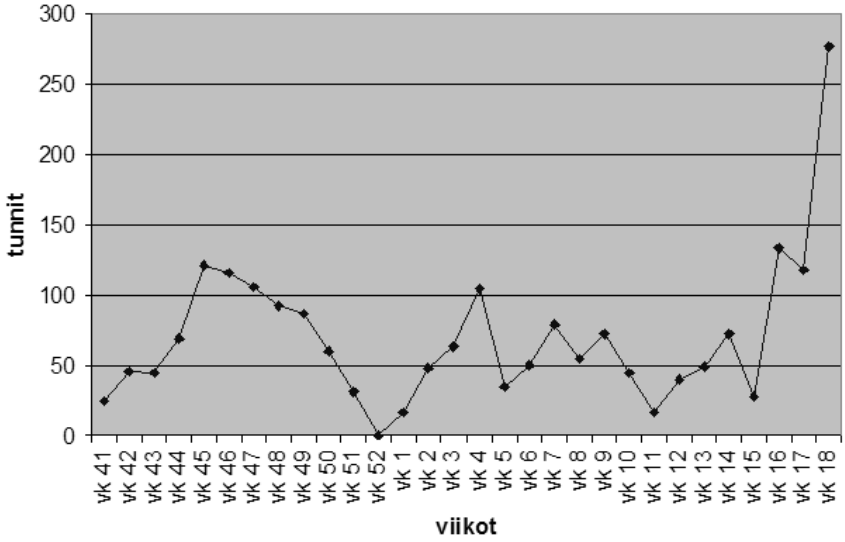

Figure 20: Projektin viikottaiset tunnit.

Table 25: Projektin koodit.

Kieli	PHP JavaScript HTML CSS
LOC	15691
SLOC	14400
Lainattua koodia	611
Lainattua ja muokattua koodia	1569
Tiedostoja	63
Funktioita	152

Table 26: Tuottavuus.

PK	LOC / PK	Tiedostot / PK	VM & S sivut / PK	LOC sivut / PK	Sivut yht. / PK
13,8	1162,3	4,6	2,7	22,7	41,2

6 SMSRaptori

6.1 Overview

The project's objective was to implement an SMS gateway system for Datapolis Oy. The system needs to relay SMS messages between mobile phones and business customers' systems. This enables SMS-based services, for example subscribing to a magazine by sending a text message. Another use case would be sending invitations, announcements etc. to a large number of people at once. All messages must be recorded in a database for billing.

In addition to the SMSRaptori software itself, an administration and billing tool was needed. SMSRaptori will replace a similar existing system. However, SMSRaptori was designed and implemented from scratch to be modular and extensible.

6.2 Organisation and management

The project managers were Hannu Lohtander, Mikko Viskari, Jon Sahlberg and Simo Tenhunen. Project crew included Antti Sand, Teemu Virta, Timo Korhonen, Tuomas Hietala and Vesa Alatalo.

Each member of the team worked mostly on their own. Meetings were held weekly where current issues with the project were discussed. Code and documentation were also reviewed during meetings. Occasionally, smaller groups would hold coding meetings. Other means of communication included an IRC channel, email and an internal wiki. Meeting summaries along with other bits of information were posted to the wiki. The managers sent out weekly reports by email.

6.3 Methods and tools

SMSRaptori was implemented in PHP, to be run on LAMP stack (Linux operating system, Apache web server, MySQL database and PHP programming language). The following sofware packages were used in development:

- Debian 4.0
- Apache 2.2.3 (Debian)
- MySQL 5.0.32
- PHP/5.2.8-0.dotdeb.1 with Suhosin-Patch
- Subversion 1.6.1

Figure 21: The project team. Hannu and Vesa are missing from the picture.

- OpenOffice.org 3.0
- Joomla! 1.5.10
- PHPUnit 3.3
- MediaWiki 1.14
- Timelet 0.2
- BOUML 4.12.3

Debian, Apache, MySQL and PHP were parts of the LAMP stack. Subversion was used for version control. Joomla! is a content management system which was used in the making of the administration interface. PHPUnit was utilized for unit testing.

Documentation was written on OpenOffice.org, with BOUML used for making UML diagrams. The project had an internal wiki powered by the MediaWiki software. Working hours were recorded using Timelet. All of the software used in development was found useful.

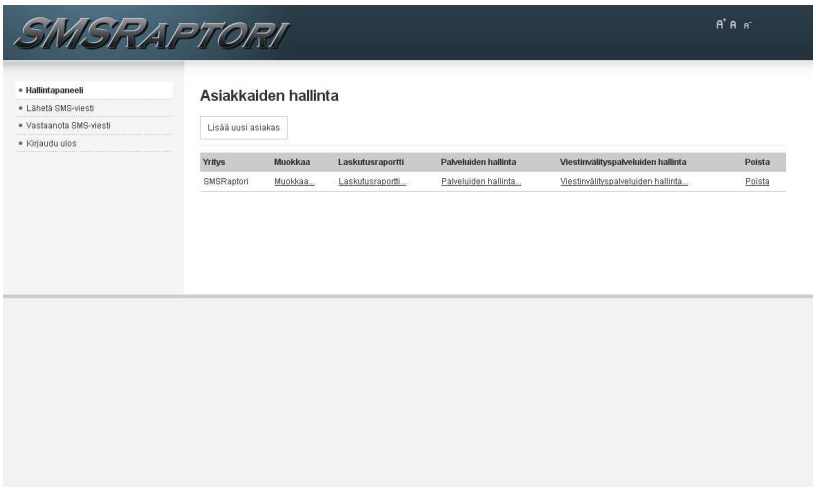

Figure 22: SMSRaptori administration interface

6.4 Project phases and development model

SMSRaptori utilized an incremental development model. The project was split into five increments, the first four of which involving design, implementation and documentation. The final increment was devoted to fixing any remaining bugs, finishing documentation and generally wrapping up the project.

- 1st increment: weeks 41 to 3
- 2nd increment: weeks 3 to 7
- 3rd increment: weeks 7 to 10
- 4th increment: weeks 11 to 15
- 5th increment: weeks 15 to 18

There were five inspections of documents during the course:

- 2008-10-30: Preliminary analysis
- 2008-12-11: Requirements specification
- 2009-01-29: Test plan
- 2009-02-27: Design plan
- 2009-04-28: Final report

6.5 Experiences

Producing object oriented PHP code while also making use of design patterns turned out to be more difficult than anticipated. Some of the team members had not programmed in PHP previously, which for them meant picking up the language on the go. Writing unit tests was also seen as difficult, and extensive test coverage was not achieved. Nevertheless, during the course of the project the PHP beginners learned the basics of PHPUnit testing in addition to the basics of the language.

The incremental development model was considered useful, allowing the project to be split into smaller pieces which could be worked on independently of each other. As the software was a redesign and reimplementation of an existing system, specifying the requirements went smoothly. The project's Subversion repository saw heavy use and was soon found to be indispensable.

Neither the project crew nor the managers were very experienced in their roles. There were some communication issues, as well as bouts of low motivation. A few of the original requirements were not met. In spite of issues, the project produced a working, modular piece of software, easily extensible should the need arise. Also, the administration and reporting facilities were significantly improved with regard to speed and usability.

6.6 Statistics

This section includes various statistics on the project.

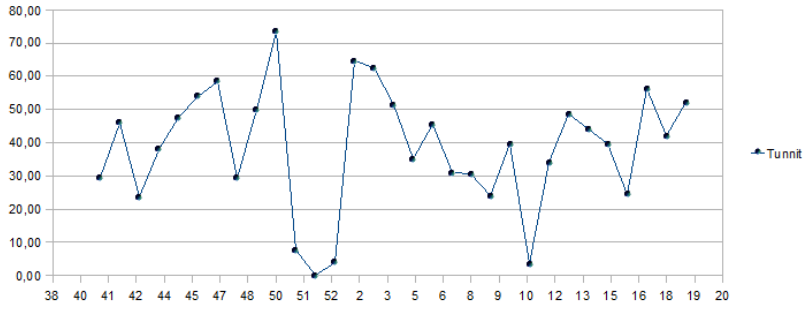

Figure 23: Weekly workload of the project

Table 27: General project information.

Team size	Dev. mod.	Start date	End date	Days	Hours	Hours / (Days * Team size)
5+4+0	Incremental	1.10.2008	6.5.2008	217	1185	0.61

Table 28: Group effort by activity.

Activity	Plan. and man.	Req. spec.	Design	Code	Integ. and test	Reviews	Repair	Study	Other	Total
Hours	41.0	3.8	4.0	23.4	1.6	3.1	1.6	8.1	13.2	
%										
Usability										0
Total	486.05	45.2	48.5	277.5	18.5	36.5	19.5	96	157	**1185.25**

Table 29: Requirements and high-level design outcome.

Pages	Requirements	Use-cases	UI screens	Database diagrams	Database tables
25	15		8	1	13

Table 30: Design outcome.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
26	1	3	1	1	3

Table 31: Inspection findings.

	Preliminary analysis	Requirements spec.	Test plan	Design plan	Final report
Pages and/or screens	14	25	24	26	29
Preparation time	6	7	5	6	6
Inspection time	7	12.5	6	5	7
Findings	7	24	8	8	1
Used time / findings	1.0	0.5	0.8	0.6	7.0

Table 32: Project's documents.

Document	Pages	versions
Preliminary analysis	14	1
Project plan	29	2
Requirements specification	25	6
Design plan	26	5
Test plan	24	3
Final report	29	3
Final story	6	1
Weekly reports	17	
Inspection reports	3	
Total	173	

Table 33: Project's codelines.

Language	PHP
SLOC	11 542
Files	148
Code revisions	403

Table 34: Productivity metrics.

PM	SLOC / PM	Files / PM	RS & DES pages / PM	SLOC pages / PM	Total pages / PM
7,8	1480	19.0	6.54	29.6	48.3

7 Vixtory

7.1 Yleistä

Projektin tarkoituksena oli jatkokehittää edellisen projektityökurssin (2007-2008) kehittämään AgileTool-työkalua. Projektin aikana työkalun nimi vaihtui AgileToolista Vixtoryksi. Vixtory on www-pohjainen työkalu vaatimusten ylläpitämiseen ja hallitsemiseen ketterässä verkkoprojektissa. Ohjelmisto mahdollistaa projektin vaatimusten liittämisen suoraan kehitettävän sovelluksen toimiviin näkymiin.

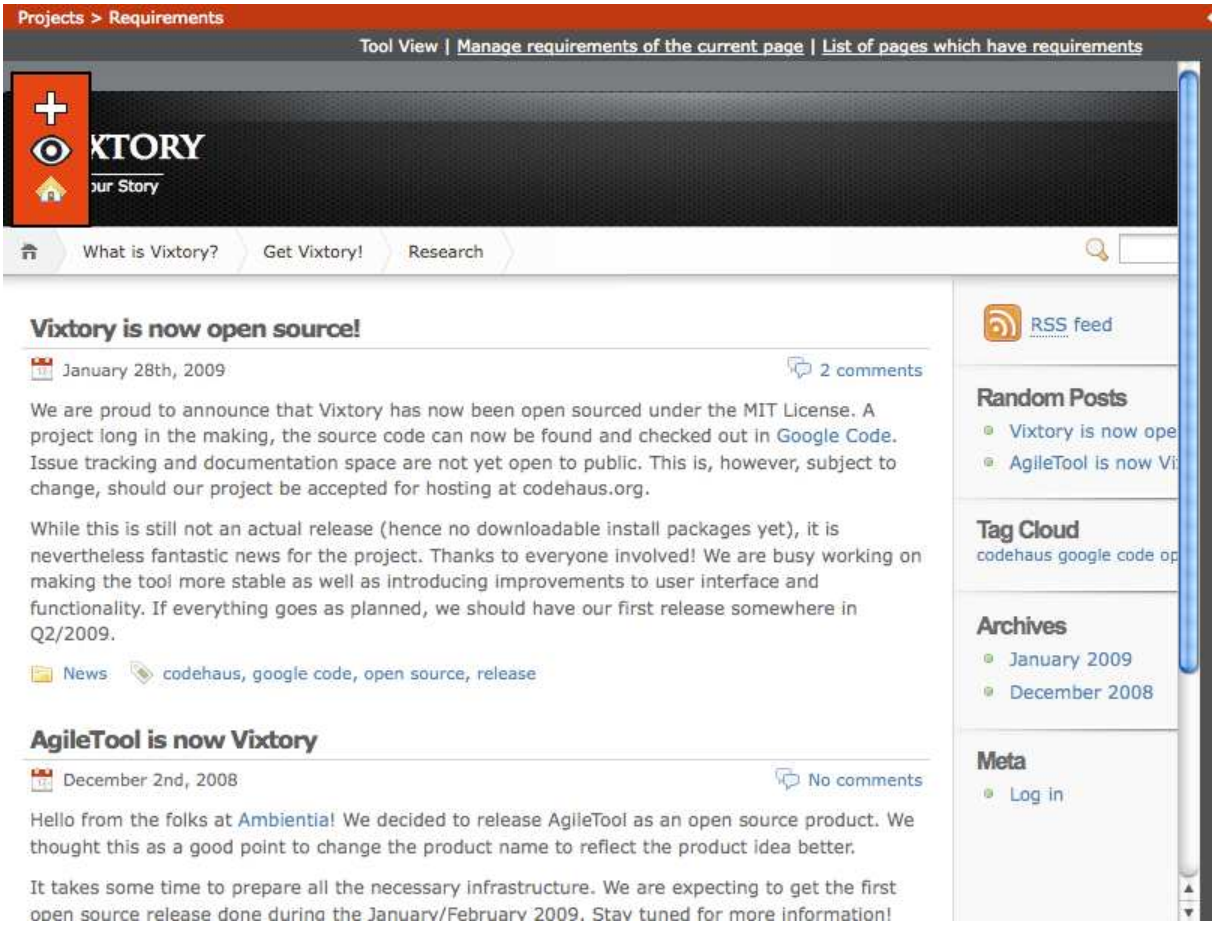

Figure 24: Kuvakaappaus vaatimustenhallintanäkymästä

7.2 Projektiorganisaatio

Projektissa työskenteli yhteensä 12 ihmistä. Projektiryhmä koostui viidestä kehittäjästä (Jami Lehtovirta, Antti Loponen, Niina Majaranta, Niina Ojala, Lasse Varjus), yhdestä käytettävyysasiantuntijasta (Nina Juuri), kolmesta projektipäälliköstä (Jukka Hell, Antti Järvinen, Juuso Kosonen) sekä kolmesta asiakkaan edustajasta (Mike Arvela, Matias Muhonen, Tero Tielinen).

Figure 25: Projektin henkilöstö.

7.3 Työskentelymetodit ja -työkalut

Projektissa käytettiin seuraavia työkaluja:

- Grails sovelluskehys ja Groovy ohjelmointikieli
- Subversion-versionhallintaohjelmisto
- Confluence-wikisovellus

Table 35: General project information.

Team size	Dev. mod.	Start date	End date	Days	Hours	Hours / (Days * Team size)
3+5+1	Scrum	1.10.2008	5.4.2009	216	1127	0.58

- JIRA-tehtävienhallintasovellus
- Skype
- Eclipse, NetBeans ja IntelliJ IDEA integroidut sovelluskehitysympäristöt.

7.4 Projektin vaiheet ja kehitysmalli

Projektimallina käytettiin opiskelijaprojektiin mukautettua Scrum-mallia.

Kehitys oli jaettu 8 sprinttiin, jotka olivat kestoltaan noin 3 viikkoa. Myös projekti piti taukoa yliopiston perioditaukojen ajan.

7.5 Statistics

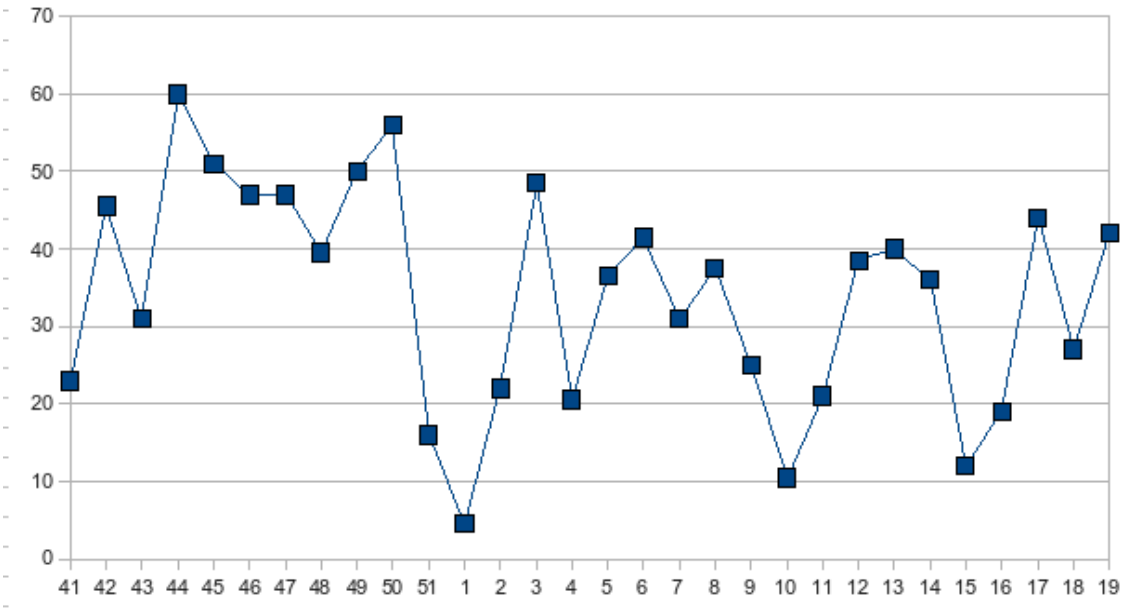

Figure 26: Tunnit viikoittain.

Table 36: Group effort by activity.

Activity	Plan. and man.	Req. spec.	Design	Code	Integ. and test	Reviews	Repair	Study	Other	Total
Hours %	43.9%	0.2%	1.4%	22.6%	0.9%	1.0%	10.3%	8.6%	11.0%	100.0%
Usability										103
Total										**1127**

Table 37: Requirements and high-level design outcome.

Pages	Requirements	Use-cases	UI screens	Database diagrams	Database tables
	25	20	7	N/A	6

Table 38: Design outcome.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
13	N/A	N/A	N/A	N/A	N/A

Table 39: Inspection findings.

	Project plan
Pages and/or screens	14
Preparation time	200 min
Inspection time	60 min
Findings	28
Used time / findings	9.28 min / finding

Table 40: Project's documents.

Document	Pages	versions
Preliminary analysis	6	3
Project plan	15	3
Project's usability plan	4	1
User interface document	6	2
Final report	13	3
Final story	6	1
Weekly reports	30	
Inspection reports	1	
Total	81	

Table 41: Project's codelines.

Language	Groovy	GSP	Javascript	Total
LOC	2420	1845	1125	5390
Files	31	24	11	66
Code revisions	84+			

Table 42: Productivity metrics.

PM	LOC / PM	Files / PM	RS & DES pages / PM	LOC pages / PM	Total pages / PM
	727	8.9	4.2	14.54	25.51

8 LiputON

8.1 Overview

The topic of the LiputON project was to develop a mobile ticketing system for TeliaSonera. The core concept of the system was that tickets could be represented by 2D barcode graphics instead of regular paper printouts. The customers could order these 2D codes to their own cellphones and use them to gain access to different kinds of events. The reading and verification of the code would be done at the event location by an employee who has a camera phone with the Java-based verification client installed. Additionally, a website with both end-user and administrative functionalities was implemented to facilitate ticket sales and event management.

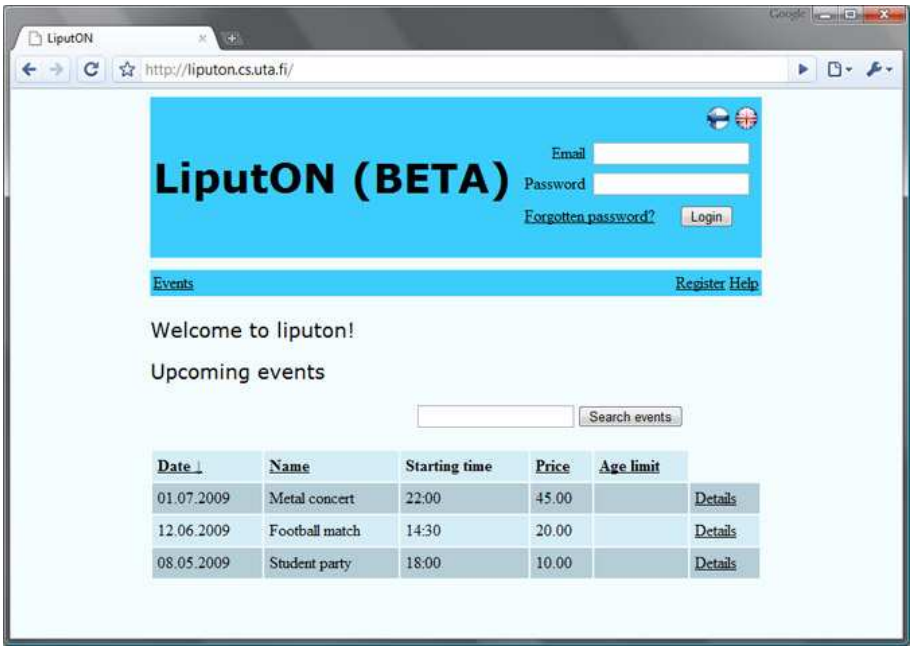

Figure 27: The main page of the website shows some upcoming events the user can book tickets for.

8.2 Organisation and management

The LiputON project group consisted of nine members. Piia Perälä, Timo Talvitie and Tero Tapio worked as project managers, and Sari Itäluoma, Nina Juuri, Maisa Lampinen, Julia Lielahti, Tuomas Kujala and Toni Miettinen as project members.

The responsibility areas were mainly distributed so that Sari, Toni and Tuomas handled the programming, Maisa and Julia the testing and creation of use cases, and Nina as the Uteam representative the UI design and usability issues. The three managers handled the day-to-day management issues, and additionally Piia worked on the concept design and was the contact person between the client and the group, Timo worked on the technical administration issues and concept design, and Tero handled the project's risk management issues and schedules.

Figure 28: Members of the group. From top-left Tuomas Kujala, Toni Miettinen, Timo Talvitie, Piia Perälä, Tero Tapio, Julia Lielahti, Maisa Lampinen, Sari Itäluoma.

8.3 Methods and tools

The client gave the group the option to quite freely choose the most convenient development tools and methods for the project. For the web development we chose PHP, and for the mobile verification client Java ME. The reasons for choosing these were the group members' previous experience regarding them, and also the availability of open source components for generating and reading 2D barcodes. The database of choice was PostgreSQL, although the use of a database abstraction layer in the implementation facilitates the use of other database types as well.

Other important tools of the project were mostly related to communication and knowledge sharing. The project had its own active IRC channel,

and additionally MSN Messenger was used very much in addition to weekly meetings and e-mail communication. Mediawiki and Subversion were used within the group to store and share various documents and code.

8.4 Project phases and development model

The LiputON project was implemented using an incremental software development model. It worked reasonably well with this particular project because even though the requirements specification did change throughout the project, there were still components that could form logical increments, be isolated from each other, and thus also be developed separately from each other.

In retrospect it could be argued that the use of some agile development model might have worked even better. The incremental model required quite a bit of planning and documentation to work. On the other hand without a thorough and approved requirements specification the implementation might have gone to a completely wrong direction, since the application area was new to most of the people in the group. The incremental development model was probably a decent compromise when trying to balance between the requirements from the client, the requirements of the course, and the tight schedules of the people in the group.

The implementation and testing of the five increments were done as follows:

- 1st increment - from November 22nd to December 10th
- 2nd increment - from December 11th to January 6th
- 3rd increment - from January 7th to January 19th
- 4th increment - from January 20th to February 1st
- 5th increment - from February 2nd to February 14th

After the fifth increment quite a lot of integration-related testing and bug fixing was done to the system to assure the quality of the deliverables. Other milestones of the project included the following document reviews:

- Review of the preliminary analysis - October 27th
- Review of the project plan - November 18th
- Review of the requirements specification - December 12th
- Review of the implementation plan and test plan (increments 1-3) - January 23rd

8.5 Experiences

The general opinion within the group was that the project was useful because it provided hands-on experience regarding what software projects are like. The project was also considered very laborous, and at least a part of the group felt that there was some stress and a sense of hurry involved. This was emphasized by the fact that all three managers worked throughout the course, as did most of the group members as well. This combined with the technically challenging nature of the project and having to focus on other university courses simultaneously caused some issues with the project schedules.

All in all the LiputON project was still definitely a success. Everybody participated in the project and provided an important contribution to the end result - a working mobile ticketing system that the client was also satisfied with. The modular nature and good documentation of the product allows the client to now develop the system further to suit their own specific needs.

8.6 Statistics

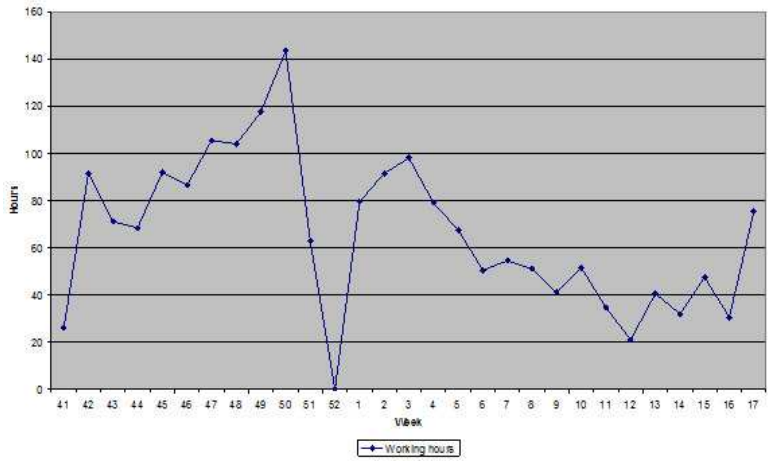

Figure 29: Working hours for each week.

Table 43: General project information.

Team size	Dev. mod.	Start date	End date	Days	Hours	Hours / (Days * Team size)
5+3+1	INC	7.10.2008	27.4.2009	203	1914,5	1.05

Table 44: Requirements and high-level design outcome.

Pages	Requirements	Use-cases	UI screens	Database diagrams	Database tables
110	44	29	32	1	10

Table 45: Design outcome.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
177	3	2	32	0	0

Table 46: Project's documents.

Document	Pages	versions
Weekly reports	21	21
Preliminary analysis	16	5
Project plan	44	5
Requirements specification	110	6
Light usability analysis	12	5
User interface plan	65	7
Implementation plan	100	8
Test plan	109	11
Test report	15	5
Manual	23	2
Final report	41	2
Project story	5	1
Maintenance document	17	2
Heuristic evaluation report	6	1
Inspection reports	4	
Total	588	

Task	Hours	Percentage
Project planning and management	601	31,4 %
Requirements specification	252	13,1 %
Design	229	12 %
Code	273,5	14,2 %
Integration and testing	148,5	7,8 %
Review	49	2,6 %
Repair	101	5,3 %
Studying	88	4,6 %
Other	172,5	9 %
TOTALS:	1914,5	100 %

Figure 30: Group effort by activity.

Programming language	Files	Program units	LOC	SLOC	Reused code (%)
PHP	44	18 classes	12179	8332	5,6 %
SQL	1	11 tables	151	140	0 %
JAVA	114	123 classes	14425	8601	91 %
TOTALS:	159	152	26755	17073	48,5 %

Figure 31: Information about the code.

Table 47: Productivity metrics.

PM	LOC / PM	Classes / PM	RS & DES pages / PM	LOC pages / PM	Total pages / PM
12,6	2123	11,2	22,8	42,5	49,9

9 Green Quest

9.1 Overview

The project subject was to create a new environmental game. The main idea was simple but quite challenging: the game should motivate people to live more environmentally friendly. Facebook was chosen as a platform for the game since it enabled the players to challenge friends and compete with them. Players can get points by answering environmental quizzes or by completing quests.

Figure 32: Green Quest - Level 2.

These quests range from simple "Recycle glass" to "Invent a fusion reactor". Game also has a mobile client which enables several options: player can take new quests, send pictures to the game and mark the quests as 'completed'. If a quest is marked as completed, it is then possible to vote for it and if there are enough positive votes, the player is rewarded with points.

9.2 Organisation and management

Project managers

- Timo Klemetti
- Markus Laurila
- Jouni Mutanen
- Reetu Mönkkönen

Software developers

- Olli Alatalo
- Lauri Hahne
- Janne Pihlajaniemi
- Mikko Tillikainen

Usability team representative

- Aapo Laitinen

Figure 33: Project Team (Timo is missing)

Project group used IRC, mailing list and phone as communication tools. At first we met only once a week, but in early February we started to meet twice a week. Meetings were kept at the Demola premises.

Responsibility areas were given as soon as each group member's talents were found. Managers responsibility areas were: Timo was our risk manager, Markus took care of the communication and weekly reporting, Jouni was our main designer and graphic artist and Reetu took care of the technical stuff (webpages, tools etc). Everyone of our staff members took part to the coding but Olli also helped Jouni to design the game. Aapo was responsible for UI design and the usability of our game.

9.3 Methods and tools

- MediaWiki
- Google Groups
- Google Docs
- IRC
- Eclipse
- Django
- WidSets

9.4 Project phases and development model

We used Scrum as well as it can be used in an university project. As you may guess, we didn't meet on a daily basis - we used IRC instead. Scrum was the perfect choice for us, allowing rather flexible requirements and documentation. We divided the project in 6 sprints but added one more sprint later on since we felt better to divided the workload even more.

Sprints

- 14.10.2008 - 2.12.2008: Projekti alkuun
- 2.12.2008 - 19.12.2008: Prototyyppi
- 19.12.2008 - 16.1.2009: Jouluinen kevytsprintti
- 16.1.2009 - 30.1.2009: Ohjelman rakenne
- 30.1.2009 - 27.2.2009: Rankkaa toteutusta

- 27.2.2009 - 9.4.: Loppujen ominaisuuksien toteutus
- 9.4.2009 - 1.5.2009: Viimeistely

Milestones

- 14.10.2008 - First meeting of project managers
- 16.10.2008 - First group meeting
- 22.10.2008 - Meeting the client
- 31.10.2008 - Preliminary analysis reviewed
- 21.11.2008 - Project plan reviewed
- 3.12.2008 - Internal release of Requirements specification
- 19.12.2008 - Prototype reviewed
- 3.2.2009 - Game design plan published
- 27.2.2009 - Sprint review
- 3.4.2009 - Test plan published
- 15.4.2009 - Test plan reviewed
- 21.4.2009 - Requirements specification published
- 21.4.2009 - Design plan published
- 25.4.2009 - Test summary published
- 29.4.2009 - Final report published
- 30.4.2009 - Final meeting
- 6.5.2009 - Project presentation
- 6.5.2009 - Final meeting with the client

We used a lot of time to study and design the game. It could be said that the design phase took too much time and we should have began the coding earlier with several prototypes. Although this is true, we are quite happy for the thorough design and planning phase; it made the implementing a whole easier phase and we did manage to fulfill the client's request for a funny environmental game. Only regret is that we didn't really use a lot of time for testing and finding bugs.

9.5 Experiences

Everyone of us feels that the project was fun and gave most of us a first glimpse to the real software development. Other university courses don't offer this kind of chance since most of them are mostly theoretical. The course didn't really feel too easy and the project tools offered a challenge since they weren't familiar to the most of us. All in all the course taught us a lot and we didn't encounter any obstacles we couldn't manage so the outcome is very positive. We give two advices to the following project groups: "Plan the project well but don't stick in plans too stubbornly if they don't work" and "Socialize within the group and discover eachother's talents early in the project".

9.6 Statistics

Table 48: General project information.

Team size	Dev. mod.	Start date	End date	Days	Hours	Hours / (Days * Team size)
4+4+1	SCRUM	14.10.2008	6.5.2009	204	1555.5	0.85

Table 49: Group effort by activity.

Activity	Plan. and man.	Req. spec.	Design	Code	Integ. and test	Reviews	Repair	Study	Other	Total
Hours	506.5	21.5	251.5	167.5	61	43	39.5	171.5	219.5	1481.5
%	34.2	1.5	16.9	11.3	4.1	2.9	2.7	11.6	14.8	
Usability										74
Total										**1555.5**

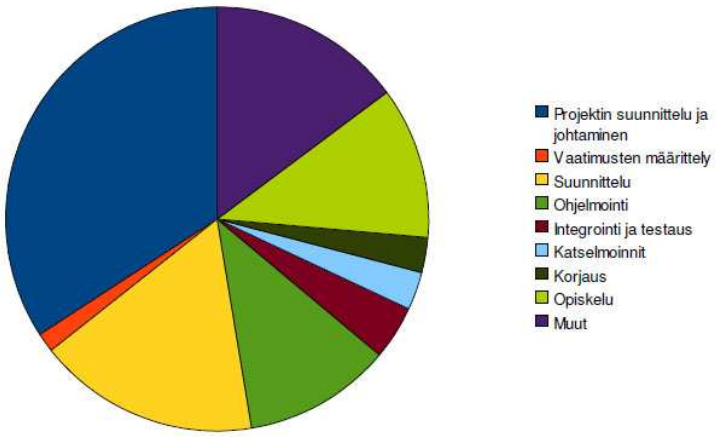

Figure 34: Work shared by categories.

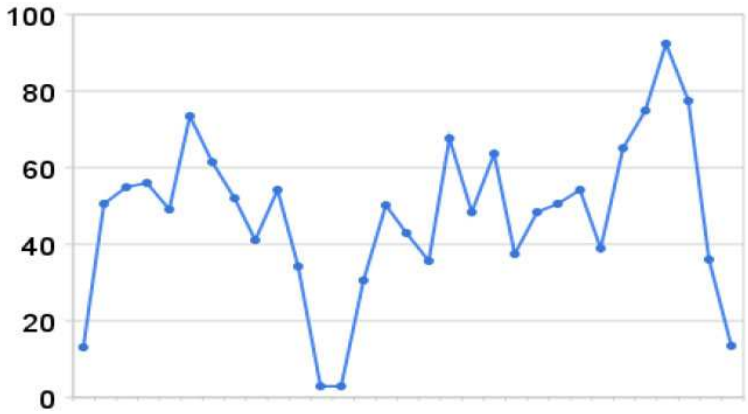

Figure 35: Weekly workload of the project.

Table 50: Requirements and high-level design outcome.

Pages	Requirements	Use-cases	UI screens	Database diagrams	Database tables
	13	12	19	3	12

Table 51: Design outcome.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
	0	1	0	0	3

Table 52: Project's documents.

Document	Pages	versions
Preliminary analysis	14	2
Project plan	33	4
Requirements specification	21	3
Design plan	14	3
Test plan	28	6
Test report	4	1
Final report	19	1
Final story	6	1
Weekly reports	28	
Inspection reports	4	
Total	171	

Table 53: Project's codelines.

Python	1713
FBML	1541
HTML	1335
CSS	2090
JS	764
Total	7443
Code revisions	1573

Table 54: Productivity metrics.

PM	LOC / PM	Classes / PM	RS & DES pages / PM	LOC pages / PM	Total pages / PM
10.2	729.7	1.2	3.7	14.6	607

10 Playful UI

10.1 Overview

The goal of the Playful UI project was to innovate and implement fun and playful user interface and interactivity features which are familiar from video games for the standard cell phone calendar of Nokia cell phones. These features include sounds, graphics and animations for the calendar, new interactions for basic functions (such as mini-games for deleting entries) and any possible new ideas that the team comes up with. The calendar application was developed for Symbian S60 3rd edition platform and is implemented using Python for Symbian. The target phones are Nokia N95 and 6210 Navigator with 320x240 screen resolution.

As part of the project we also performed a user study of the application with twenty participants. The goal of the study was to evaluate the selected playful features that had been implemented during the project. The objective was to determine whether they enhance the user experience or not. The user's expectations and satisfaction about playfulness were also evaluated. The most important conclusion of the study was that the playful content features were ranked the highest.

10.2 Organisation and management

The project team consists of project managers taking Software Project management course and team members including one usability team member taking the Project Work course. The project managers are:

- Joonas Mäkinen (joonas.makinen@cs.uta.fi)
- Ville Rahikainen (ville.rahikainen@uta.fi)
- Juuso Raitala (juuso.raitala@uta.fi)

Other team members are:

- Gururaj Mahajan (gururaj.mahajan@uta.fi)
- Mikko Arminen (mikko.arminen@uta.fi)
- M.Orkun Oẍen (orkun.ozen@uta.fi)
- Jean Fairlie (jean.fairlie@uta.fi)
- Piotrek Lewandowski (piotrek.lewandowski@gmail.com)

- Pan Pan (pan.pan@uta.fi)
- Stanislav Radomskiy (stanislav.radomskiy@uta.fi)

Piotrek Lewandowskis was primarily available for the project until December. There is also a usability team (U-Team) member assigned for the project:

- Antero Salokangas (antero.salokangas@uta.fi)

Figure 36: Project team

10.3 Tools

The application will run on mobile phones which have Series 60 3rd edition Feature Pack 2 or 3 and it will use Python for Symbian v. 1.4.4. No single programming tool had been found, thus all the members used which ever programming tool they feel most comfortable with. Subversion was used for centralized version control.

Communication tools used by the project were IRC, Skype and e-mail (including a mailing list). IRC was initially pushed to common use, but was not adopted and remained to be used mainly by the project managers. Skype was successfully used to communicate when meeting in real life was

not feasible. Email and mailing lists were also essential in keeping touch both within the team and with the client and the course staff.

The project has a wiki installed for internal communication, collaborative editing of documents and as a project homepage for external communication. The wiki software used (TikiWiki) proved to be too difficult and troublesome to use for making good-looking documents, so much of the documentation was made using traditional office software such as Microsoft Office or OpenOffice.org. As a project home page and sprint backlog a wiki proved fine, as it allowed every team member to make quick changes easily.

10.4 Development model and project phases

Our development model was Scrum with six sprints. Sprint length varied from two to four weeks depending on current situation and impending milestones such as the user study. First sprints were used to duplicate basic calendar functionality and the later one increasingly to innovate and implement playful features. By the beginning of each sprint sprint-specific requirements were pushed to the current sprint backlog. In the end of each sprint the sprint outcome was evaluated and used in helping to plan the following sprint. Instead of daily scrum meetings we held similar meetings once or twice a week depending on current situation.

As we did not have a strict requirements specification provided to us in the beginning of the project, using an agile software development methodology was deemed the only realistic possibility. Requirements were defined and refined throughout the project, especially when the next sprint was about to begin.

Inspections and reviews were organized with the project course staff when mandated. The inspections did not affect the progress of the project in any notable way, but they improved the quality of the documentation under inspection. No in-team inspections and reviews were organized.

10.5 Actualized risks

The following risks materialized throughout the project:

- Platform and programming language difficultness and insufficient programming skills slow down the project. Only a few team members actually took part in the implementation due to skill or other issues.
- Team doesn't have needed special skills. Graphic and sound expertise was missing.

- Team members being busy with other studies and work. We were quite sure this was going to take place, and it did. However, it's difficult to assess how much it actually delayed our progress.

- Allotted hours exceeded. Some team members greatly exceeded the estimation (circa 200 hours) allocated for each person. Worth noting is that hours were very varyingly divided within the project team. This was due to fact that only part of the team actively participated in the implementation work which proved to be the bottleneck.

- Estimation failures. Providing accurate estimations proved to be difficult throughout the project. However, the estimations became more accurate as project progressed. Also the initial COCOMO estimations were not accurate, but this was expected due to agile software development methodology used and the feature list not being complete when the estimation was performed.

- Project too big for the group. The project proved to be more labourious than initially seen.

- Uneven distribution of hours. Some team members possessed special skills crucial to the project, thus they became overemployed.

- Integrating of team members into programming. We failed to familiarize some team members to the platform, the programming language and especially the current state of the application. Thus not all were able to catch up the implementation work as others progressed.

The last two risks above were not foreseen in the initial risk assessment.

10.6 Experiences

We regard the project at least a partial success. The application has much features and the feature goals agreed with the client in the beginning of the project have been mostly met. Our end product may not be the most polished and the product quality could be better, but the application by now acts as a successful demonstration of a functional playful application. We liked the possibility to get programming experience the project provided. The project showed us real problems and issues probably also faced in commercial real-life software development and software projects. We got much time management expertise, both personal and for organizing the time management issues for the whole team. Also the project showed ourselves our personal level of expertise.

Figure 37: Screenshot of the application.

As we were probably the most multicultural project this year, we had our share of communication problems due to cultural and language differences. Being a cross-cultural team also proved to be an advantage. It forced us to improve our understanding of different people and to share our cultural values with each other, thus the team work taught us more than otherwise possible. The project helped us to improve our interpersonal skills. In the future we will be better off when working in cross-cultural projects. Also in brainstorming new ideas the different backgrounds we had proved to be an advantage.

Overall this was not an easy project due to project size and challenges in cross-cultural communication. Nevertheless we strived to and succeeded in keeping up high team spirit in spite of difficulties.

10.7 Statistics

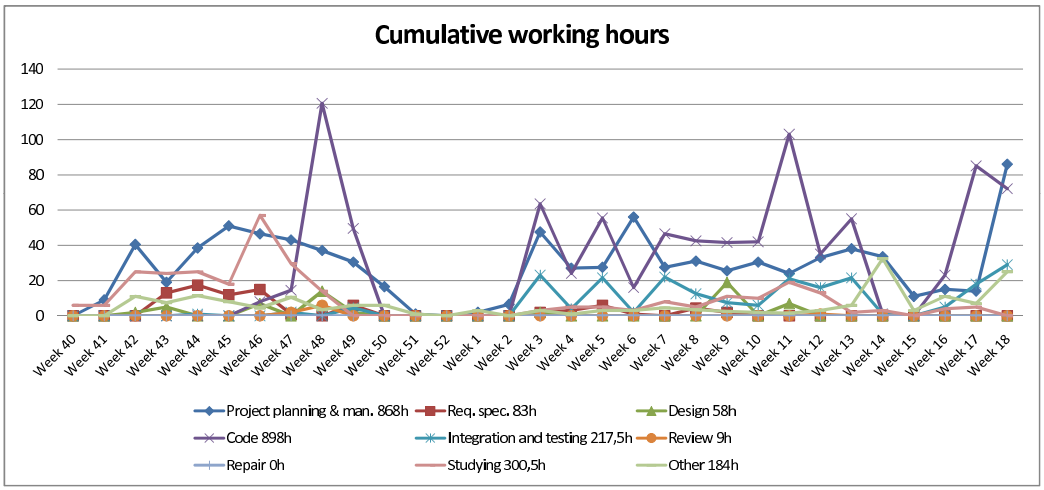

Figure 38: Weekly workload of the project.

Table 55: General project information.

Team size	Dev. mod.	Start date	End date	Days	Hours	Hours / (Days * Team size)
3+7+1	Scrum	8.10.2009	6.5.2009	210	2858	1.24

Table 56: Group effort by activity.

Activity	Plan. and man.	Req. spec.	Design	Code	Integ. and test	Reviews	Repair	Study	Other	Total
Hours	868	83	58	898	217.5	9	0	300.5	184	2618
%	33.2%	3.2%	2.2%	34.3%	8.3%	0.3%	0%	11.5%	7%	100%
Usability										240
Total										**2858**

Table 57: Inspection findings.

	Project plan Inspection 1	Project plan inspection 2
Pages and/or screens	22	22
Preparation time	3h 10min	10h 40min
Inspection time	1h 50min	2h 45min
Findings	N/A	80
Used time / findings	N/A	10min

Table 58: Project's documents.

Document	Pages	versions
Preliminary analysis	8	4
Project plan	23	14
Project's usability plan	9	3
Requirements specification	21	7
Design plan	16	4
User interface document	14	5
Test plan	18	10
User's guide	N/A	N/A
Test report	N/A	N/A
Usability test report	N/A	N/A
UI Heuristic evaluation report	N/A	N/A
Final report	25	8
Final story	6	5
Weekly reports	28	
Inspection reports	4	
Total	172	

Table 59: Project's codelines.

Language	Python
LOC	10050
SLOC	6800
Reused code	N/A
Reused and modified code	N/A
Files	38
Functions	654
Code revisions	250

Table 60: Productivity metrics.

PM	LOC / PM	Files / PM	RS & DES pages / PM	LOC pages / PM	Total pages / PM
	535	2.0	N/A	10.7	16.3

11 NovelUI

11.1 Overview

This is the project story for the NovelUI project. The project was run under the project work course at the Computer Science department at the University of Tampere. The purpose of the project was to create an interactive 3D world which represents users' digital music collection in a mobile phone.

Because of some problems with using sound together with 3D models, the final product is a demo that works on PC, but not on mobile.

In the demo, the user can move around in a 3D city, which has 5 different music districts. These districts differ from each other by their looks and by the characters that roam there. The user can talk with the characters to listen to music and collect the songs to their playlist.

11.2 Organisation and management

The project group consisted of 3 managers and 6 developers, as well as one usability specialist from the usability team. At the beginning of the project there were 7 developers, but one developer had to leave in December.

Kirsikka was responsible for risk management, weekly reports, schedule and resource management. Hannu took care of quality and testing, and Maria was responsible for usability and design.

The project group had weekly meetings first at the university and later at the Demola premises. In addition to the meetings, communication was handled through a wiki, e-mails and some people even found their way to the project's IRC channel.

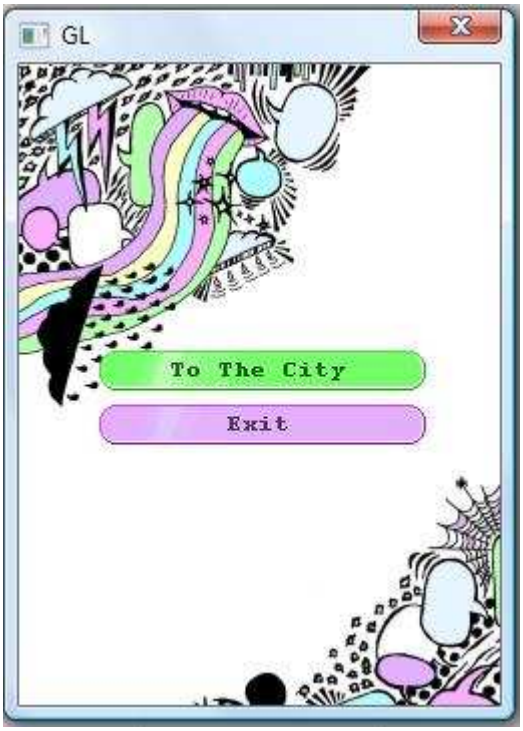

Figure 39: Menu screen.

11.3 Methods and tools

Carbide.c++ 2.0 was used for coding the software, with ActivePerl 5.6.1 library, S60 SDK and Open C/C++ Plug-ins for mobile development. Nokia PC Suite was used for transferring files and application between PC and mobile phone. Google Sketchup Pro was used for modelling houses and exporting models into COLLADA format.

The wiki software used in the project was MediaWiki and it was installed on the virtual server provided by the computer science department. Subversion was used for version control.

11.4 Project phases and development model

The development model used in the project was Scrum. Its principles had to be modified because it was not possible to arrange daily meetings for the group. Instead, a wiki page was used to emulate daily meetings. There were a total of 7 sprints, with each sprint lasting for 2-5 weeks.

In the beginning of the project, new technology was identified as the greatest risk, since only one developer had prior experience with 3D devel-

Figure 40: Character in the city.

Table 61: Project phases

Date	Event
07.10.08	Initial information about the project
14.10.08	First manager meeting
17.10.08	First meeting with the client and managers
21.10.08	First project group meeting
31.10.08	Preliminary analysis review with course supervisor
03.11.08	Client meeting with the whole project group
17.11.08-19.12.08	Sprint I
20.11.08	Project plan inspection
10.12.08	Course presentation about the project
16.12.08	Sprint I review with the client
05.01.09-25.01.09	Sprint II
26.01.09-15.02.09	Sprint III
27.01.09	Sprint II review with the client
16.02.09-08.03.09	Sprint IV
09.03.09-26.03.09	Sprint V
27.03.09	Sprint V review and testing session with the client
27.03.09-09.04.09	Sprint VI
10.04.09-30.04.09	Sprint VII
28.04.09	Final meeting with course supervisor
30.04.09	Official end of the project
06.05.09	Final project presentation
07.05.09	Final meeting with the client

Figure 41: Character dialog.

opment and mobile was also unfamiliar territory for most. This turned out to be quite true. There were some major problems with using music and 3D models on the mobile the way that was planned.

The project group was quite large, which made effective communication and coordination somewhat challenging. This could have been partially solved by arranging meetings more often, but finding even one time a week that suited everyone was problematic.

Although the group knew that one developer would leave the project in December, there were some unforeseen problems with that.

11.5 Experiences

Although the project was very challenging, and we did not succeed in implementing the originally envisioned idea, the general feeling is still that the project succeeded. The idea for the project was very ambitious and some of the technologial problems could not be foreseen. However, the final product is still a decent demo of the original idea and, in our opinion, shows what the concept could be capable of.

Next time it would be a good idea to spend more time specifying the

Figure 42: Playlist.

Table 62: General project information.

Team size	Dev. mod.	Start date	End date	Days	Hours	Hours / (Days * Team size)
3+6+1	Scrum	7.10.2008	7.5.2009	213	1576	0.74

requirements for the project, as well as to have some sort of code inspections with the group during the project. Testing was also something that was overlooked in the project, because most of the time there just wasn't much to test. More time should have been put into making sure that the development and the UI specification were in sync.

11.6 Statistics

Table 63: Group effort by activity.

Activity	Plan. and man.	Req. spec.	Design	Code	Integ. and test	Reviews	Repair	Study	Other	Total
Hours	526	42.5	79	275.5	36.5	73.5	51.5	213	137.5	1435
%	36.6	3.0	5.5	19.2	2.5	5.1	3.6	14.8	9.6	100
Usability										141
Total										**1576**

Table 64: Requirements and high-level design outcome.

Pages	Requirements	Use-cases	UI screens	Database diagrams	Database tables
	23	2	6	0	0

Table 65: Design outcome.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
13 (+ appendixes)	2	0	1	0	2

Table 66: Inspection findings.

	Project plan	Sprint II review	Sprint III review
Pages and/or screens	22	N/A	N/A
Preparation time	6h 35min	N/A	N/A
Inspection time	10h 50min	10h	6h 45min
Findings	31	N/A	N/A
Used time / findings	34min	N/A	N/A

Table 67: Project's documents.

Document	Pages	versions
Preliminary analysis	10	8
Project plan	25	11
Usability analysis	14	3
Requirements specification	21	2
Design plan	13(+appendixes)	8
User interface plan	29	9
Test plan	13	5
User + maintenance guide	8	1
Test report	4	1
Final report	23	7
Final story	9	1
Weekly reports	29	
Inspection reports	6	
Total	204	

Table 68: Project's codelines.

Language	C/C++
LOC	27872
SLOC	12784
Reused code	50%
Files	31
Functions	72
Code revisions	315

Table 69: Productivity metrics.

PM	LOC / PM	Files / PM	RS & DES pages / PM	LOC pages / PM	Total pages / PM
9.4	2965	3.3	3.6	59	21.7

Figure 43: A view of the city.

Figure 44: Project managers: Hannu, Kirsikka and Maria.

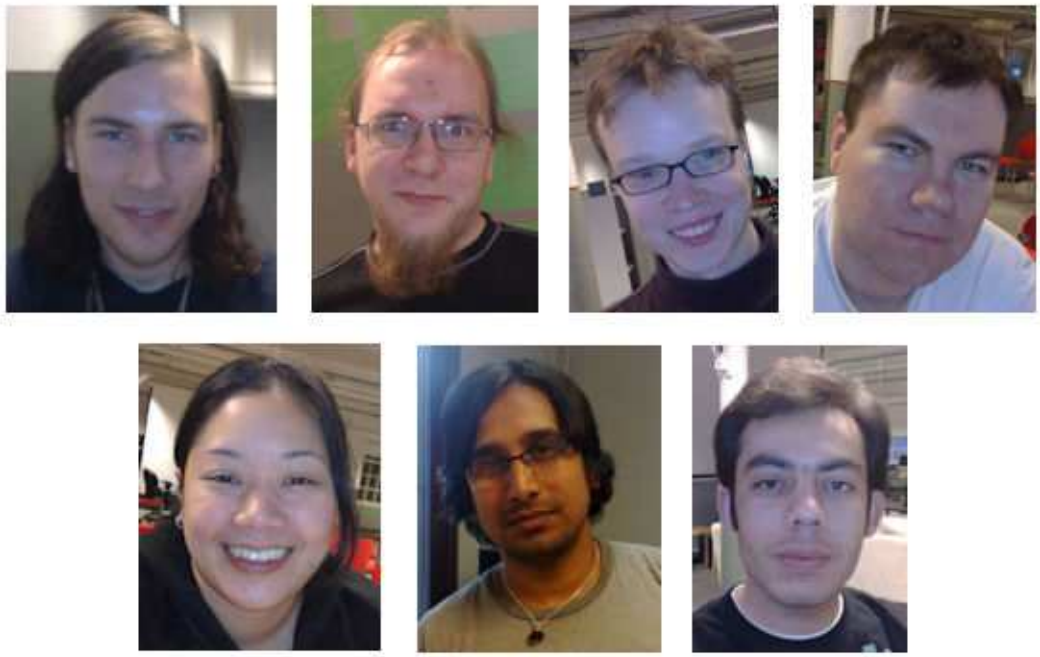

Figure 45: Usability person and developers: Janne, Jefim, Matias, Mika, Misada, Rajib and Tausif.

Figure 46: Weekly workload of the project.