

Kuinka toteuttaa helppokäyttöinen karttapalvelu

Ville Hytönen

Opinnäytetyö
Syyskuu 2009

Tietojenkäsittely



JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) HYTÖNEN, Ville	Julkaisun laji Opinnäytetyö	Päivämäärä 28.08.2009
	Sivumäärä 45	Julkaisun kieli Suomi
	Luottamuksellisuus () saakka	Verkojulkaisulupa myönnetty (X)
Työn nimi KUINKA TOTEUTTAA HELPPOKÄYTTÖINEN KARTTAPALVELU		
Koulutusohjelma Tietojenkäsittely		
Työn ohjaaja(t) BISTER, Timo		
Toimeksiantaja(t) Codewise Oy		
<p>Tiivistelmä</p> <p>Opinnäytetyön tavoitteena oli tutkia, kuinka toteuttaa helppokäyttöinen karttapalvelu. Lisäksi tutkittiin muita Web 2.0 -teknologioita, Web 2.0 -tekniikoita ja Web 2.0 -ilmiöitä. Varsinkin Web 2.0 -teknologioita ja Web 2.0 -tekniikoita hyödynnettiin työssä. Tutkimuksen lähtökohtana oli toteuttaa järjestelmä käyttäen karttamashup-rajapintaa, jota hyödyntäen saataisiin toteutettua toimiva karttajärjestelmä. Tutkimustyössä olen myös hieman tutkinut käytettävyyttä, joka on yksi tärkeimpiä asioita järjestelmien kehittämisessä.</p> <p>Tutkimus oli kehittämisprojekti ja tutkimuksen aikana kehitettiin ja toteutettiin toimiva karttajärjestelmä toimeksiantajalle, mistä koostuikin suurin osuus tutkimustyöstä. Web-järjestelmässä yhdisteltiin Web 2.0 -teknologioita, Web 2.0 -tekniikoita ja erityisesti karttamashup-rajapinta-osa-aluetta, joka oli koko tutkimustyön lähtökohtana. Teoriaosuudessa tutkittiin Web 2.0 -ilmiötä ja selvitettiin, mitä se tarkoittaa, koska termi on vielä epäselvä niin internetin käyttäjille kuin Web-kehittäjillekin. Työssä käytiin läpi termin historiaa ja sitä, mistä se on tullut ja miksi sitä nykypäivänä käytetään päivittäin useissa järjestelmissä ja palveluissa. Lisäksi käytiin läpi, miten Web 2.0 -idologiaa on hyödynnetty nykypäivänä muuallakin kuin web-järjestelmien kehityksessä.</p> <p>Tutkimustyön tuloksena saatiin toimiva karttajärjestelmä, joka yhdistää Google Maps -karttamashup-rajapintaa järjestelmään ja on käyttöön otettavissa nopeasti uusille asiakkaille suunnittelunsa ansiosta.</p>		
Avainsanat (asiasanat) Kehitysprojekti, web 2.0, karttamashup		
Muut tiedot		



Author(s) HYTÖNEN, Ville	Type of publication Bachelor's Thesis	Date 28082009
	Pages 45	Language Finnish
	Confidential () Until	Permission for web publication (X)
Title HOW TO IMPLEMENT EASY-TO-USE MAPSERVICE		
Degree Programme Business Information Systems		
Tutor(s) Bister, Timo		
Assigned by Codewise Oy		
<p>Abstract</p> <p>In Bachelor's Thesis was studied how to implement an easy-to-use mapservice. Web 2.0 technologies, Web 2.0 methods and Web 2.0 phenomenon were studied within the scope of the research topic. Especially Web 2.0 technologies and Web 2.0 methods were used in the research. The baseline of the research was to execute a system using a map mashup interface, and benefiting from this a functional map system could be created. In the research usability was also studied, one of the most important parts of developing systems. Usability was studied only in connection with Internet web pages.</p> <p>The research method in the project was development where nothing in particular was researched generally but a fully functional web service was developed for the assigner Web 2.0 technologies, Web 2.0 methods and specially mashup interface –part were mixed in the research with the web system, this mashup being the whole baseline of the research. In theory part Web 2.0 phenomenon was studied and it was discussed what it means, because to a basic user of the Internet and web system developer it is still a term and issue which is unfamiliar. Term history was focused on, where it originates from and why it is used nowadays in several systems and services on an every day basis.</p> <p>The result of this study was a working map system that uses Google Maps map interface to web system and which is quick to deployment for new customers because of its design and implementation in the project. In literary research Web 2.0 possibilities were studied in development of web systems and what different technologies and methods there are around Web 2.0 term.</p>		
Keywords Development project, web 2.0, map mashup		
Miscellaneous		

SISÄLTÖ

1	JOHDANTO.....	6
2	TUTKIMUSASETELMA	7
2.1	Työn lähtökohdat	7
2.2	Tutkimuskysymykset.....	8
2.3	Tutkimusmenetelmät	9
3	WEB 2.0 MASHUP -SOVELLUKSEN KEHITYS	10
3.1	Web 2.0.....	10
3.2	Web 2.0 -teknologia	13
3.3	Web 2.0 - ja Mashup -teknologia	15
3.4	Käytettävyys	19
4	TUTKIMUKSEN TOTEUTUS	21
4.1	Työn aloittaminen.....	21
4.2	Esitutkimus	21
4.3	Ratkaisuvaihtoehtojen pohdinta ja perustelut.....	23
4.4	Vaatusmäärittely.....	24
4.5	Suunnittelu.....	25
4.6	Tekniikat ja käytänteet.....	26
4.7	Ensimmäisen prototyypin toteutus	27

	5
4.8 Järjestelmän kehittyminen	28
4.9 Ylläpito	28
4.10 Testaus	29
4.11 Karttasovelluksen haasteet.....	29
5 TULOKSET JA JOHTOPÄÄTÖKSET	31
5.1 Järjestelmän toiminnallisuuden esittely.....	31
5.2 Saavutukset	33
5.3 Vastaukset tutkimuskysymyksiin	34
5.4 Ohjelmistoprojektin arviointi	38
6 POHDINTA	41
LÄHTEET	43

KUVIOT

KUVIO 1. Järjestelmän arkkitehtuuri	32
KUVIO 2. Google Mapsin plussat ja miinukset	37
KUVIO 3. Microsoft Live Mapsin plussat ja miinukset	38

1 JOHDANTO

Tutkimuksen lähtökohtana on tilanne, jossa on tutkittu, kuinka pystytään toteuttamaan karttajärjestelmä mahdollisimman helposti kuntien käyttöön. Karttajärjestelmällä tarkoitetaan tässä työssä internet-järjestelmää, joka toimii Web-selaimessa ja sisältää interaktiivisen karttaelementin. Tätä hyödyntäen on tuotu kartalle muita objekteja ja karttaelementtejä. Tarve tämänlaiselle karttajärjestelmälle on ollut jo jonkin aikaa olemassa Suomen kunnilla. On todella yleistä, että ihmiset hakevat paikkoja ja palveluita olemassa olevista karttapohjaisista palveluista. Tämän vuoksi työssä on tuttu, kuinka on mahdollista toteuttaa mahdollisimman vaivattomasti helppokäyttöinen karttajärjestelmä Uuraisten kunnan käyttöön.

Tutkimustyön voi jakaa kahteen osa-alueeseen: teknisenä työnä tutkimustyössä on toteutettu toimiva karttajärjestelmä, joka yhdistelee Web 2.0 -tekniikoita ja erityisesti karttamashup-tekniikkaa, johon myös tutkimus painottuu. Tämän lisäksi teoriaosuudessa on tutkittu Web 2.0 -teknologiaa, Web 2.0 -tekniikkaa ja itse Web 2.0 -ilmiötä. Tämä on ollut tärkeää tutkimuksessa, koska Web 2.0 on niin kehittäjille kuin peruskäyttäjillekin vielä osaltaan outo sekä tuntematon, ja kaikki eivät välttämättä tiedä, mitä se tarkoittaa nykypäivän internetin käyttämisessä. Web 2.0:sta on tutkittu myös sen tuomia ilmiöitä, internetin kehittymistä Web 2.0:n ansiosta ja sitä, millä tapaa se on vaikuttanut nykyisen internet-infrastruktuurin kehitykseen.

Toimivan karttajärjestelmän kehittämisen yhteydessä on tutkimuskohteiksi otettu järjestelmässä käytettävät tekniikat ja teknologiat. Näistä työn lähtökohtana oli käyttää järjestelmässä karttamashup-rajapintaa. Mashup-rajapinnan lisäksi järjestelmässä on käytetty muita hyväksi havaittuja tekniikoita ja teknologioita, jotka erityisesti helpottavat järjestelmän käytettävyyttä loppukäyttäjän näkökulmasta. Järjestelmän kehitystä on seurattu projektinhallinnoinnin näkökulmasta ja kehityksessä on käytetty olemassa olevaa toimivaa ohjelmistokehitysmallia. Tähän liittyvien vaiheitten mukaisesti projekti on viety läpi yhdessä työn toimeksiantajan kanssa.

2 TUTKIMUSASETELMA

Tässä luvussa esitellään työn lähtökohdat, työn rajausta, tutkintakysymykset ja tutkimusmenetelmät.

2.1 Työn lähtökohdat

Codewise Oy on vuonna 2003 perustettu pieni ohjelmistoyritys, joka on erikoistunut sisällönhallintaan ja webjärjestelmien tuottamiseen. Yritys työllistää tällä hetkellä 3 työntekijää. Yrityksen päätuotteena on Block-ohjelmistoperhe, jonka tarkoitus on suodattaa haitallista sisältöä internet-sivustoilta. Toimeksiantajalla oli tarvetta kehittää mashupeja yhdistelevä karttapalvelujärjestelmä ja tutkia, kuinka tämän tapainen järjestelmä voitaisiin toteuttaa mahdollisimman helposti. Toimeksiantaja huomasi tarpeen syntyvän tämän tapaiselle järjestelmälle ja sen kehitykselle, kun kunnan asiakkaat tarvitsivat karttajärjestelmää kunnan palveluista tiedottamiseen ja markkinoimiseen.

Kehitysprojektin aikana tutkittiin karttamashup -rajapintojen yhdistelemistä nykyisiin Web 2.0 -tekniikoihin ja käyttöliittymän käytettävyyttä suunnittelun ja loppukäyttäjän näkökulmasta. Myös työn lähtökohtana on suunnitella ja toteuttaa järjestelmä, joka käyttää karttamashup-rajapintaa. Tavoitteena tutkimustyöllä on toteuttaa toimiva karttajärjestelmä, joka on käytettävyydeltään selkeä ja mahdollisimman helposti uudelleen toteutettavissa oleva järjestelmä uuden asiakkaan käyttöön.

Aihe työssä on rajattu karttapalvelun suunnitteluun ja toteutukseen. Tämän yhteydessä tutkitaan toteutettavan järjestelmän suunnittelua ja toteutusvaiheen tarpeita ja niitä tekniikoita, joita tämänlaisen järjestelmään suunnittelussa ja toteutuksessa tarvitaan. Lisäksi tutkitaan käytettävyyttä sen verran, kuin se vaikuttaa järjestelmän käyttöliittymään. Missään muussa muodossa ei tutkimuksen aikana tutkita käytettävyyttä, koska se ei tutkimustyöhön muulla tapaa liity. Näillä rajoituksilla pyritään pitämään oma tutkimustyön mahdollisimman selkeänä ja tiiviinä, jolloin työ pysyy selkeänä ja myös hyvin rajattuna.

2.2 Tutkimuskysymykset

Opinnäytetyölle on määritetty seuraavat tutkimuskysymykset:

1. Mihin asioihin suunnittelussa pitää kiinnittää huomioida mashup-tekniikkaan perustuvaa järjestelmää toteutettaessa?

Tässä kohdassa paneudutaan enemmän mashup-tekniikoihin ja asioihin, joihin pitäisi kiinnittää huomiota, kun suunnitellaan mashup-teknologialla toteutettavaa Web-järjestelmää. Lisäksi tuodaan esille hyvät puolet ja huonot puolet, suunnittelun ongelmakohdat sekä haasteet.

2. Miten mashup-tekniikkaan perustuvan palvelun käyttöliittymästä saadaan käytettävyydeltään mahdollisimman selkeä?

Tämän kysymyksen tarkastelussa tutkitaan, kuinka mashup-tekniikkaan perustuvan järjestelmän käyttöliittymästä saadaan käytettävyydeltään mahdollisimman selkeä käyttäjän kannalta. Tämä oli työn suunnittelussa yksi tärkeimmistä vaatimuksista, koska järjestelmän käyttäjäkunta oli ikäjakaumaltaan laaja. Tämän vuoksi järjestelmästä piti saada niin yksinkertainen ja helppo, että järjestelmän käyttö olisi helppoa.

3. Kuinka ilmaiset mashup-rajapinnat eroavat tosistaan?

Lisäksi tutkimuskysymyksissä selvitetään hieman mashup-rajapintojen eroja. Tarkastelun kohteena ovat lähinnä karttapalveluihin soveltuvat rajapinnat, mutta lisäksi tuodaan hieman esille, mitä muita ilmaisia mashup-rajapintoja on olemassa. Karttarajapintojen välillä selvitetään niiden eroavaisuuksia, eri karttarajapintojen hyviä ja huonoja puolia, mahdollisia riskejä sekä sitä, mitä haasteita rajapinnat tuovat kehittäjälle toimittaessa mashup-rajapintojen kanssa.

Nämä kysymykset mielestäni kuvaavat opinnäytetyön tutkimusprosessia ja auttavat kuvaamaan työn tulosta paremmin. Näihin kysymyksiin vastataan työn kuluessa tarkemmin.

2.3 Tutkimusmenetelmät

Tutkimus on kehittämistutkimus, jonka tarkoituksena oli suunnitella ja toteuttaa mashupeja käyttävä internet-järjestelmä toimeksiantajalle. Tiedon hankinta ja kerääminen tutkimuksen aikana on toteutettu asiakashaastattelulla, jossa saatiin selville, mitä pilottiprojektiin saadun asiakkaan vaatimukset järjestelmälle olivat. Nämä vaatimukset on avattu tarkemmin Tutkimuksen toteutus -luvun vaatimusmääritelmä-kohdassa. Tiedonkeruussa on hyödynnetty projektin ohjelmistototeutusvaiheessa tapahtunutta havainnointia ja myös projektin sisäisiä palavereja ja katselmointeja. Näistä on ollut suuri hyöty tiedon hankinnassa ja keräämisessä.

Omasta mielestäni tärkeimpänä tiedonkeruumenetelmänä kehittämistutkimuksen aikana on ollut ongelmanratkaisutapa, jota hyödyntäen olen pystynyt keräämään tietoa tutkimuksen aikana hyvin ja hyödyntämään alan ja aiheen kirjallisuutta. Näitä tapoja hyödyntäen on onnistuttu löytämään Codewise Oy:lle toimiva ratkaisu toteuttaa internet- järjestelmä, joka hyödyntää kartta mashup-rajapintaa.

3 WEB 2.0 MASHUP -SOVELLUKSEN KEHITYS

3.1 Web 2.0

Web 2.0 -termillä tarkoitetaan yleisellä tasolla Internetin 2.0 -versiota. Tätä voidaan kuvastaa ainoastaan yleisellä tasolla sen takia, koska Internet 2.0:aa ei ole missään vaiheessa julkistettu, vaan nykyinen internet on muovaantunut Web 2.0 -ilmiöiden vuoksi sellaiseksi, mitä se nykypäivänä on. Alun alkaen kyseinen termi on kehitetty O'Reilly and Associatesin sekä MediaLive Internationalin järjestelmän konferenssin markkinointiin, ja tämän jälkeen kyseinen termi on levinnyt laajempaan käyttöön kuvaamaan internetin seuraavaa versiota.

Termillä viitataan internetin kehittymiseen ja siirtymiseen toiminnallisempiin WWW-pohjaisiin sovelluksiin. Web 2.0:lla tarkoitetaan sosiaalisempaa lähestymistapaa sisällön tuottamiseen ja jakeluun. Yleisesti korostetaan myös avoimempaa keskustelua, päätösvallan hajauttamista, tiedon vapaata jakamista sekä tiedon uudelleen käyttämistä. Ennen tämä ei ollut mahdollista, mikä pääasiassa johtuu tekniikan rajallisesta mahdollisuudesta toteuttaa vapaata tuottamista, mistä syystä sitä ei ole pystytty toteuttamaan. Toinen syy on ollut murrosvaihe internetiin, joka on vasta 2000-luvun taitteessa muuttunut huomattavasti siitä, mitä se joskus on ollut. Nykyistä internetiä ei ole alun perin suunniteltu siihen, mitä se nykyään on. Web 2.0:n merkityksestä ei ole kuitenkaan vielä täyttä yksimielisyyttä, koska epäilijät väittävät koko termin olevan turha, koska sitä käytetään liian paljon, miten halutaan. Osaksi tämä voi johtua siitä, että termin vaatimuksia ei ole määritelty tarpeeksi hyvin, minkä vuoksi melkein jokaisen WWW-sovelluksen, joka käyttää hyväkseen jotain dynaamista ja uutta tekniikkaa tai tapaa, voidaan määritellä olevan Web 2.0 -sovellus. (Paul Graham - Web 2.0, Web 2.0 - johdatus Internetin uusiin liiketoimintamahdollisuuksiin 2007, 8 - 13.)

Monia käsitteitä ja teknologioita on yhdistetty Web 2.0 -ilmiöön, kuten esimerkiksi blogit, podcastit, RSS-syötteet ja erilaiset yhteisölliset WWW-sovellukset, kuten Wikipedia ja Flickr. Yksi laajimmista Web 2.0 -ilmiöistä on ehdottomasti Web-standardit ja AJAX. Näitä ei normaali käyttäjä välttämättä huomaa, mutta ne ovat toiminnassa nyky-internetissä jokapäiväisillä sivustoilla ja palveluissa, joissa käyttäjät

viettävät aikaansa. Selaimet käyttävät Web-standardia, joka määrittelee, miten elementit sivulla näkyvät ja toimivat käyttäjälle.

AJAX-tekniikka taas hyödyntää Javascriptiä ja XML:ää, jota hyödyntäen voidaan luoda sivustolle hienoja, visuaalisia ja näyttäviä graafisia elementtejä. Yhtenä osana Web 2.0 -ilmiöitä ovat WWW-rajapinnat ja API-kirjastot, joihin tässä työssä on enemmän keskitytty. WWW-rajapinnat ja API-kirjastot ovat rajapintoja ja apukirjastoja. Ne tuovat web-järjestelmään jotain omaa erityistä. Esimerkiksi karttarajapinnat mahdollistavat karttaelementin piirtämisen järjestelmään ja näin ollen kartan näyttämisen käyttäjälle web-selaimen kautta.

Jotkut API-kirjastot monipuolistavat olemassa olevien ohjelmointikielien toimintoja tai vain yksinkertaisesti helpottavat niiden käyttämistä, ja tämä taas nopeuttaa toimimista kyseisellä ohjelmointikielellä tai tekniikalla. Tällaisia ovat esimerkiksi erityisen suositut AJAX-kirjastot, joista suosituimpia ovat JQuery- ja MooTools-kirjastot. Käytettävät rajapinnat mahdollistavat täysin uudenlaisten WWW-pohjaisten palveluiden kehittämisen. Syynä siihen, että avoimia rajapintoja ja API-kirjastoja hyödynnetään järjestelmän kehityksessä, on se, että ne nopeuttavat järjestelmän kehitystä moninkertaisesti verrattuna siihen, että monet yksinkertaiset toiminnot täytyisi kehittäjien itse toteuttaa omaan järjestelmäänsä. Tästä syystä järjestelmän kehitys voi nopeutua moninkertaisesti, mikä taas säästää järjestelmien kehittämisessä rahaa. Omassa kehitystyössäni olen hyödyntänyt useita erilaisia rajapintoja sekä API-kirjastoja, jotka olen nähnyt tarpeellisiksi niin järjestelmän toimivuudessa kuin multimedia ominaisuuksien hyödyntämisessäkin.

(Yee 2008, 77 - 105, RSS tutorial n.d.)

Web 2.0:n tekniset mahdollisuudet ovat todella laajat, minkä vuoksi monet harrastajat ja ohjelmistokehittäjät ovat kehittäneet monia innovatiivisia Web-palveluita, jotka helpottavat ihmisten elämää tai sosiaalista verkostoitumista ja yhteydenpitoa. Web 2.0 mahdollistaa todella monimutkaisten ja monipuolisten järjestelmien toteuttamisen, joissa käyttäjillä on laajat mahdollisuudet toteuttaa itseään ja saada erilaisia elämyksiä. Parhaimpana esimerkkinä voidaan mainita 2000-luvun suurin ilmiö, ”sosiaaliset palvelut”, joista käytetään termiä ”sosiaalinen media”. Näistä esimerkkeinä mainittakoon räjähdysmäisesti suosiota saaneet Facebook, MySpace ja suomalainen IRC-Galleria. Mainitut järjestelmät ovat erinomaisen hyviä esimerkkejä

sosiaalisen median tuomista eduista palveluissa. Idea on pääosin siinä, että tehdään järjestelmä, jota ei ole ennen ollut ja joka tarjoaa ihmisille elämyksen ja palvelun, jossa jokainen käyttäjä voi vaikuttaa järjestelmän tai palvelun kehitykseen itse. Näin ollen jokainen käyttäjä on järjestelmässä sisällöntuottajana.

Tätä samaa ideaa on hyödynnetty muuallakin teollisuudessa. Nykyään esimerkiksi peliteollisuudessa on käytetty samaa Web 2.0:n tuomaa ilmiötä, jossa käyttäjät voivat itse tuottaa sisältöä järjestelmiin ja peleihin. Web 2.0 -ilmiö ei ole ainoastaan vaikuttanut käyttäjien käyttäytymiseen verkossa tai järjestelmien laajuuteen, vaan myös markkinointi ja liiketoimintamallit ovat saaneet oman osansa internetin muuntautumisesta. Tämän vuoksi markkinointi ja liiketoiminta ovat entistä enemmän siirtyneet internetiin ja avanneet ovia uudentilaisille markkinoinneille ja liiketoimintamalleille.

On syntynyt suuria IT-yhtiöitä, kuten Google, jonka liikeidea pääosin on lähtenyt mainoksien markkinoinnista, joita he myyvät. Näin mainokset saavat enemmän näyttötilaa heidän omassa hakukoneessaan. Tämä liikeidea ei olisi toimiva, ellei Googlen kehittämä hakukone olisi niin suosittu, kuin se nykyään on. Hakusanoille voidaan myydä mainostilaa ja näin ollen mainostilan ostava yritys saa hyvää markkinointia hakujen ohessa. Lisäksi tietty haku pystytään nostamaan ylemmäksi, jolloin tuotteen tai markkinoitavan mainoksen näkyvyyttä saadaan parannettua. Tämän vuoksi internet-markkinoinnista on tullut todella suosittua. Yhtenä uutena liiketoimintamallina on tullut virtuaalitavaroiden myyminen ja siihen liittyvä virtuaaliraha.

Tämä lähti liikkeelle Habbo Hotellin innovatiivisena ideana, jossa luotiin interaktiivisempi paikka, jossa ihmiset pystyivät keskustelemaan sekä viettämään vapaa aikaansa. Tuottoa järjestelmään tuli siitä, että järjestelmästä käyttäjät pystyivät ostamaan hahmoilleen uusia tavaroita, joista he olivat valmiit maksamaan. Tämä on laajentunut moniin palveluihin ja internet-järjestelmiin, joissa voi ostaa kolmansien osapuolien tekemiä asioita ja tavaroita, joilla voi nostaa omaa imagoa. Yleisesti nämä liittyvät suoraan sosiaalisen median palveluihin tai peliteollisuuteen. (Blogger's Manifesto: Free Speech and Censorship in the Age of the Internet, Broughton 2008. Googlen mainokset ja muita sosiaalisen median liiketoimintamalleja).

3.2 Web 2.0 -teknologia

Teknologiana Web 2.0 on vielä kehittyvä ja monitahoinen. Teknologia sisältää palvelinpuolen ohjelmistoja, sisällön syndikointia ja erilaisiin viestintäprotokolliin ja standardeihin pohjautuvia selaimia. Nämä kaikki yhdessä mahdollistivat sivustoilta sellaisia ominaisuuksia, joita ennen ei ole osattu edes odottaa sivustoilta ja jotka nyt on mahdollisia ja nähtävissä useilla sivustoilla. Yhtenä tunnetuimpana tekniikkana näistä on AJAX. AJAX-teknologia itsessään ei ole mitään, vaan se on yhdistelmä monia jo olemassa olleita tekniikoita (JavaScript, XML, XHTML, DOM).

AJAXin suurimpana hyötynä web-järjestelmien kehittämisessä on se, että se mahdollistaa tietojen lataamisen selaimessa ilman uutta sivulatausta. Tämä antaa aivan erilaisen mahdollisuuden tehdä paljon sulavampia sivustoja ilman, että sivua pitäisi ladata uudestaan esim. tehtäessä jonkinlaista hakua tai vaikka lajiteltaessa hakutulosta uudestaan. Tämän tyyppinen AJAX-toteutus järjestelmään voidaan toteuttaa oikeastaan kahdella tavalla. On mahdollista rakentaa toimintakohtaisesti AJAX-toteutus suoraan järjestelmään tai tehdä palvelimen ja selaimessa näkyvän sivun välille ns. AJAX-kerros, joka hoitaa AJAX-toteutukseen liittyvän kommunikaation ja toimintalogiikan näiden kahden osapuolen välillä. (Samcom artikkeli – Mikä on Web 2.0, What is Web 2.0 - O'Reilly Network 2005.)

Suurimpana etuna käyttäessä AJAX-teknologiaa järjestelmän toteutuksessa ovat sivujen nopeammat ja huomaamattomat latausajat sekä monipuolisten käyttöliittymätoimintojen toteuttaminen selainkäyttöliittymään. Näin pystytään pilkkomaan haettavaa tietoa palvelimelta, jolloin tiedon haettava koko on paljon pienempi, mikä tietenkin vaikuttaa sivuston ja järjestelmän latausaikoihin dramaattisesti, minkä myös käyttäjä huomaa. Asiasta tekee huomaamattomampaa se, että tietoa haetaan taustalla, jolloin käyttäjä ei oikeastaan edes huomaa asiaa.

AJAX-toteutuksen rinnalle on viime aikoina myös tullut muita teknologioita, jotka mahdollistavat monipuoliset käyttöliittymät ja tuovat erilaisia lähestymistapoja monipuolisen verkkopalvelun suunnitteluun ja toteuttamiseen. Yksi suosiota nostonut tapa on RIA-teknologia (Rich Internet Application). Tässä teknologiassa periaatteena on se, että käyttäjälle tuodaan käyttöliittymä, ja hänen toimintojensa ja tekojensa perusteella näytetään erilaista tietoa käyttöliittymässä. Adoben Flex on yksi

suosituimmista RIA-sovelluskehitysratkaisuista, mutta samanlaista sovelluskehitystä voidaan tehdä myös AJAX-toteutuksessa yhdistämällä palvelin ohjelmointikieleen. Jälkimmäinen vaihtoehto ei välttämättä ole niin hyvin optimoitu ja tehokas kuin valmis ympäristö.

Adoben Flex on täysin suunniteltua tähän sovellusratkaisuun, minkä vuoksi se on paljon tehokkaammin optimoitu kuin AJAX-toteutusta yhdistelevä sovellusratkaisu. Lisäksi hyöty käytettävässä valmiiksi suunniteltuja RIA-sovelluskehitysratkaisuissa on yleensä rahallinen, mistä kehittäjät ja palvelun toteuttajat ovat valmiit maksamaan. Yleisesti kysymyksessä on palvelun nopeuteen ja optimointiin liittyvä asia. Nämä erot ovat yleisiä ilmaista ja maksullisten järjestelmien välillä.

AJAX-sovelluskehitysratkaisulla on täysin mahdollista toteuttaa kyseisiä sovellusratkaisuja ja hyötynä on avoimuus. Varmasti löytyy monella tapaa avoimia kehitysalustoja, jotka ovat usein ilmaisia ja joissakin tapauksissa myös parempia kuin maksulliset alustat. Toinen kilpaileva tekijä on hinta, joka on pienempi, tai ratkaisu on täysin ilmainen verrattuna Adoben Flexiin. Tämä onkin yksi syy, minkä vuoksi avoimella lähdekoodilla on suuri kannattaja- ja kehittäjäyhteisö, sekä miksi ihmiset jaksavat ja haluavat kehittää ohjelmistokehitysratkaisuja ilmaiseksi.

Web 2.0 -maailmassa on myös tuleva trendi, joka on ns. ”pilvipalvelut” jotka lähinnä tarkoittavat, että nykyinen käyttöjärjestelmä siirtyy yhä enemmän internetiin ja näin ollen web-palvelimille, joista pystytään jopa lataamaan kokonainen käyttöjärjestelmä tietokoneelle ajettavaksi. Tätä kautta on mahdollista ajaa käytettäviä ohjelmia tai palveluita internet-yhteyden kautta, jotka normaalisti toimisivat paikallisesti omalla koneella. Hyötynä on se, että ohjelmia ja palveluita voidaan käyttää missä tahansa, eikä vain ainoastaan omalta kotikoneelta. Tarvitaan ainoastaan toimiva internet-yhteys, jota hyödyntämällä käyttöjärjestelmä voidaan ladata suoraan internetistä toimivalle tietokoneelle. Nykypäivänä tämän on mahdollistanut se, että internet-yhteyksien nopeudet ovat kasvaneet niin paljon, että tiedonsiirto on riittävän nopeaa toteuttamaan tällaisia ratkaisuja.

Yksi tunnetuimmista pilvipalveluratkaisuista on Googlen Docs-palvelu. Palvelun idea ja pääperiaate on mahdollistaa tekstinkäsittely missä ja milloin tahansa ilman, että käyttäjän tarvitsee asentaa koneelleen mitään. Lisäksi tämä kyseinen palvelu on täysin

ilmainen, ja se kilpailee tämän vuoksi johtavan tekstinkäsittelyohjelmiston Microsoftin Office-pakettia vastaan. Tulevaisuudessa on nähtävissä pilvipalveluiden lisääntyminen niin, että koneisiin ei tarvitsisi asentaa enää ollenkaan paikallisesti toimivia ohjelmistoja, vaan kaikki ohjelmistot on ajettavissa internetistä joko selaimen kautta ilman, että käyttäjän tarvitsee näitä ohjelmia asentaa koneelleen. Tämä taas helpottaa tietokoneen käyttöä entisestään. (How cloud computing works, Web 2.0 and Cloud Computing 2009.)

3.3 Web 2.0 - ja Mashup -teknologia

Mashup-termi tarkoittaa Web-järjestelmien ohjelmistokehityksessä WWW-järjestelmää, jossa yhdistetään tietoa useammasta kuin yhdestä paikasta ja yhdistetään ne yhdeksi integroiduksi järjestelmäksi. Hyvänä esimerkkinä tästä on opinnäytetyön teknisessä osuudessa tehty Karttapalvelu-järjestelmä. Karttapalvelu käyttää olemassa olevista Mashup-rajapinnoista vain yhtä ainoaa rajapintaa, mutta todellisuudessa tähän palveluun olisi voitu liittää useampiakin rajapintoja yhdistämään erilaisia toimintoja.

Ohjelmistokehityksessä ei ole ennen otettu huomioon, että voitaisiin näin vapaasti yhdistellä olemassa olevia API-kirjastoja ja muita rajapintoja yhdeksi järjestelmäksi, joka mahdollistaisi aivan erilaisten järjestelmien toteuttamisen suurimmaksi osaksi ilmaisilla työkaluilla. Web 2.0:n tuomien mahdollisuuksien turvin tämä on tullut mahdolliseksi toteuttaa. Mashupin käytön suosio on Web 2.0 -tekniikan tuleminen jälkeen noussut valtavasti. Tämä voi osittain johtua esimerkiksi Google-yhtiön tapaisista palvelun tarjoajista, jotka ovat kehittäneet ilmaiseen käyttöön työkaluja ja rajapintoja, jolloin jokainen voi tehdä omia sovelluksia yhdistelemällä näitä rajapintoja.

Näiden lisäksi on olemassa valtavasti ilmaisia API-kirjastoja, joita ennen Web 2.0:aa ei ole ollut, vaan suurin osa on pitänyt tehdä itse tai ostaa ulkopuoliselta tekijältä. Tällä hetkellä nykyiset Web 2.0 -standardit ja olemassa olevat tekniikat mahdollistavat mashupien, API-kirjastojen, erilaisten tekniikoiden ja standardien sekoittamisen yhteen järjestelmään ja näiden toteuttamisen todella hyvin, mutta tulevaisuudessa on tulossa paljon suurempia ja laajentuvaisempia tekniikoita, joita hyödyntämällä voidaan toteuttaa helposti ja vaivatta organisaation kokoisia palveluita

tai sovelluksia. Näitä tekniikoita hyödyntämällä voidaan rakentaa vain tiettyjä kerroksia järjestelmästä entistä helpommin ja tehdä web-sovelluskehityksestä enemmän komponentti-mallista, jolloin voidaan vain kehittää jokin uusi komponentti järjestelmään. Mielestäni Web-kehityksessä ei ole vielä täysin huomioitu Mashup-palveluiden ja rajapintojen hyödyllisyyttä. Ohjelmistokehittäjät ovat heränneet ns. Web 2.0 -aikakauteen, jotta olemassa olevia tekniikoita ja tapoja pystyttäisiin täysin hyödyntämään. (Google Maps API Documentation, ProgrammableWeb: How to make your own web mashup 2009.)

Ilmaiset mashup -karttarajapinnat

Kehittämisprojektin alkaessa ensimmäisenä asiana oli kartoittaa kaikki ilmaiset karttamashup-rajapinnat. Tarkoituksena oli selvittää, minkälaisia esimerkkiprojekteja kyseisillä rajapinnoilla on ennestään tehty, kuinka helppoa niillä oli kehittää web-järjestelmää ja kuinka ne yleisesti toimivat. Tarkoituksena oli kerätä kaikki oleellinen, jota tarvittiin kehittämisprojektin etenemiseen niin tutkimustyön näkökulmasta kuin ohjelmistokehittämisprojektin näkökulmastakin valitsemaan oikea työkalu tälle projektille.

Ilmaiset karttamashup-rajapinnat ovat Googlen Maps, Microsoftin Maps Live ja Yahoo Maps. Lisäksi otettiin muodon vuoksi Map Server -projekti, jonka tarkoituksena on toteuttaa samanlainen rajapinta, jota Google, Microsoft ja Yahoo tarjoavat. Näistä kyseisistä karttamashup-rajapinnoista voitiin valita ainoastaan kaksi tarjoajaa ehdokkaiksi. Nämä olivat Google Maps ja Microsoftin Live Maps. Syyksi yksinkertaisesti voitiin laskea se, että näissä oli kehityspanoksia eniten ja että nämä kaksi olivat ylivoimaisesti monipuolisemmat tarjolla olevat rajapinnat.

Näistä laajin ja kauemmin kehityksen alla ollut oli Googlen Maps-rajapinta. Tämä johtui siitä, että kyseinen palvelu oli julkistettu jo helmikuussa 2005, jonka vuoksi kyseisen rajapinnan kehitykseen oli käytetty vuosia enemmän resursseja sekä käyttäjien mielipiteitä siitä, mihin suuntaan rajapintaa pitäisi kehittää. Tämä on ehkä tehnytkin rajapinnasta niin loistokkaan ja monipuolisen. . Kyseinen rajapinta on todella laaja ja sisältää paljon monipuolisia toimintoja, jotka yksinkertaisesti helpottavat kehittämistä kyseisellä rajapinnalla. Se on lisäksi tuonut rajapinnalle sellaisia piirteitä, joita kehittäjät ovat halunneet karttarajapinnalta saatavan.

Erityisenä helpottavana piirteenä rajapinnalle oli tuttu kommunikointikieli, jonka vuoksi sen kehittäminen oli äärimmäisen helppoa harrastajille ja web-kehittäjille. Lisäpiirteenä oli, että tätä pystyttiin erittäin helposti yhdistelemään muihin Web 2.0 -tekniikoihin, teknologioihin ja jo tuttuihin asioihin. Uutena mahdollisuutena keväällä 2008 Google julkaisi Maps- rajapinnalle myös mahdollisuuden käyttää kyseistä rajapintaa Flash multimedia -tekniikan kautta. Tämä mahdollisti multimediapainotteisemman karttasovelluksien tekemisen. Sovellukset olivat näyttäviä ja muutkin kuin ohjelmistokehittäjät pystyivät tekemään niitä. Tärkeintä oogle Maps kehityksen kannalta oli AJAX -tekniikka, joka mahdollisti synkronisia sivun latautumisia ja elementtien tuomista kartan rajapinnalle yksinkertaisen helposti ja nopeasti. Näin ollen kartalle pystyttiin tuomaan suuria ja raskaita elementtejä, mutta myös niin nopeasti ja huomaamattomasti, että tämä ei järjestelmien käytettävyyttä millään tapaa haitannut. (Google Maps announcement on Google Blog, Google Maps Api Documentation 2009.)

Google Maps -rajapinnalle on olemassa paljon yhteisön tuottamia apusivustoja eli ohjesivustoja, jotka helpottavat aloittelijan työtä kehittämisessä. Monet harrastajat ovat tehneet sivustoja ja ns. wiki-sivustoja, joista saa paljon apua työn aloittamiseen. Näitä on olemassa enemmänkin, mutta ne, jotka on tutkimuksen aikana onnistuttu paikallistamaan, on otettu mukaan tähän tutkimukseen ja listattu lähteissä. Näiden lisäksi on Googlen oma keskustelupaikka, jossa on oma osasto kyseiselle rajapinnalle, jossa käyttäjät voivat ehdottaa uusia kehitettäviä asioita, kysyä apua ongelmiinsa ja auttaa muita käyttäjiä.

Haastajana tällä sektorilla Microsoft oli suurella panostuksella saanut oman karttarajapinnan näkymään julkisuuteen, mikä selvästi toi kamppailua rajapintojen osalta kehittäjille. Microsoftilla oli ehkä suurimpana hankaluutena saada oma rajapinta jollain tapaa eroamaan Googlen kyseisestä rajapinnasta. Tähän selvästi oli panostettu erilaisella lisenssisopimuksella, joka pohjautui "karttatiilien" latausmääriin. Myös Microsoft oli selvästi panostanut karttapuolen käyttöliittymän kehittämiseen, koska tästä oli selvästi tarkoituksella tehty hieman hienompi verrattuna Google Mapsiin. Selvin eroavaisuus oli mahdollisuus selata karttaa 3D-tilassa suoraan selaimella. Tämä oli täysin uutta selainkäyttöliittymissä. Google sai jälkeempään omansa versionsa kehitettyä, mutta tämä oli selvin keino, jolla Microsoft yritti kiihdyttää Googlen saamaa etumatkaa kiinni webjärjestelmien maailmassa. Googlella oli jo aikaisemmin ollut

Google Earth -palvelu, joka mahdollisti 3D-selaamisen, mutta tämä oli erillinen sovellus eikä web-selaimella toimiva palvelu.

Ominaisuuksien osilta Microsoftin Live Maps on alusta lähtien ollut vähintään yhtä laadukas karttamashup-rajapinta. Tottunut käyttäjäkunta on Google Mapsilla paljon suurempi, minkä vuoksi näillä strategisilla vedoilla Microsoft sai näkyvyyttä ja julkisuutta vielä jokseenkin tuntemattomalle web-palvelulleen. Ainoana ongelmana Microsoftin Live Maps-rajapinnalla on sen ohjelmointikielen riippuvuus.

Ohjelmointikielen pakkollisuus ei ole välttämättömyys, mutta Microsoft on antanut omille rajapinnoilleen ohjeita ainoastaan .NET framework -pohjaisille ohjelmointikielille. Tämä tarkoittaisi järjestelmän toteuttamista .NET framework tekniikalla. Ainoana ongelmana ja riskinä tässä on, että tällä hetkellä PHP on johtava Web-kieli, jonka vuoksi harrastelijat eivät lähde välttämättä opettelemaan uutta kieltä vain karttarajapinnan takia. Tosin ammattitaitoiset osaavat tehdä samat ominaisuudet ilman ohjeita PHP-ohjelmointikieltä hyödyntäen. (ProgrammableWeb: How to make your own web mashup, Google Maps Api Documentation 2009.)

Muut ilmaiset mashup-rajapinnat

Karttarajapintojen lisäksi on tietysti olemassa muitakin mashup-rajapintoja, joita voidaan hyödyntää Web 2.0 -järjestelmien kehittämisessä. Silti mashup-rajapinnoista käytetään eniten karttarajapintoja laajasti niiden käytettävyyden ja suosionsakin vuoksi. Työssä mainitaan muutama merkittävä ilmainen mashup-rajapinta, jotka on hyvä tietää. Ne on rajattu ainoastaan sen perusteella, mikä on niiden näkyvyys internet-maailmassa ja mitkä ovat niiden mahdollisuudet kehittää nykyistä internet-sivujen monimutkaisuutta, infrastruktuuria ja sisältöä. (ProgrammableWeb: How to make your own web mashup, Web 2.0, mashups and social networking - what is it all about? 2009.)

Youtube-videopalvelu-sivustolla on myös Google Mapsin tapaan olemassa oma rajapintansa kehittäjille, jota voidaan hyödyntää. Rajapinta mahdollistaa tällä hetkellä erilaisia toimintoja videoleikkeisiin kuin käyttäjätietojen hakemiseen. Youtube-palvelussa käyttäjätiedot ovat julkisia ja tämän vuoksi niitä voi kuka vain hakea käyttäen tätä rajapintaa. Youtubea on mahdollista myös soveltaa vaikka Google Mapsin päälle, jolloin voidaan tiettyyn koordinaattiin asettaa videoleike, jota käyttäjän

on mahdollista katsoa. Tämä on tietysti vain yksi mahdollisuus, johon tätä voi käyttää. Googlen palveluista oikeastaan todella monet palvelut mahdollistavat julkisen rajapinnan kautta toimivuuden. Toinen mainitsemisen arvoinen on Picasa-kuvapankkipalvelu. Tämän mahdollistaa monenlaisien yleisien toimintojen käyttämisen rajapinnan kautta, jota normaalisti voitaisiin hallita käyttöliittymä tason kautta. Toinen kuvien julkaisemiseen liittyvä tunnettu palvelu Flickr mahdollistaa myös mashupien sekoittamisen muihin järjestelmiin. Se on melkein verrattavissa suoraan Picasa-kuvapalveluun, joskin se on ehkä vielä suositumpi kuin Picasa. (ProgrammableWeb: How to make your own web mashup, Youtube APIs and Tools 2009.)

3.4 Käytettävyys

Käytettävyyteen painottamista järjestelmien kehittämisessä ei painoteta koskaan liikaa. Tämä on yksi tärkeimpiä asioita kehittämisprojektia aloitettaessa. Monelta osalta tämä vaikuttaa käyttäjien psykologiseen haluun sekä omakohtaisiin tuntemuksiin siitä, onko järjestelmää mukava käyttää. Tämän vuoksi tämä asia on otettava huomioon järjestelmien kehityksessä, koska jos järjestelmä on hankala käyttää ja jokin asia ei käyttäjän mielestä tunnu luonnolliselta, niin monesti käyttämisestä tulee hankalaa ja ongelmallista eikä kyseistä järjestelmää haluta käyttää. Tämän vuoksi on kehitetty ISO-standardi käytettävyydelle, joka määrittää järjestelmän käytettävyyttä ja tyytyväisyyttä siihen.

ISO 9241-11 -standardi määrittää käytettävyyden seuraavasti: "Se vaikuttavuus, tehokkuus ja tyytyväisyys, jolla tietyt määritellyt käyttäjät saavuttavat määritellyt tavoitteet tietyssä ympäristössä."

- Vaikuttavuudella tarkoitetaan sitä, miten tarkoin ja täydellisesti käyttäjä saavuttaa tavoitteensa.
- Tehokkuus tarkoittaa tavoitteiden saavuttamista suhteutettuna käytettyihin resursseihin.
- Tyytyväisyydellä tarkoitetaan käyttäjän tyytyväisyyttä laitteen tai järjestelmän käyttöön, vuorovaikutuksen sujuvuuteen ja sen tulokseen.

Käytettävyys liittyy todella vahvasti tuotteen toiminnallisuuteen ja käyttäjän käyttömukavuuteen. Tämä ei aina ole pelkästään vain helppouteen liittyvä tekijä vaan monesti myös psykologinen tekijä, joka vaikuttaa käyttäjän tapaan oppia käyttämään asiaa tai tapaa tehdä jotain. Käytettävyyttä on nykyään alettu nostamaan päätöksenteon strategiselle tasolle. Tässä tapauksessa puhutaan käytettävyysstrategiasta, joka yleensä tarkoittaa sitä, että organisaation on määritettävä tavoite käytettävyystyölle organisaatiossaan ja askeleet, miten tavoitteeseen päästään lyhyellä ja pitkällä aikavälillä. Lisäksi on määritettävä, kuinka saadaan käytettävyyden tavoitteet laajemmin tunnetuksi organisaatiossa. Se, kuinka laajasti käytettävyys otetaan osaksi tuotesuunnittelua, vaihtelee paljon yrityksissä.

Monet yritykset ja organisaatiot nostavat käytettävyyden todella tärkeäksi osaksi tuotekehitystä. Monissa muissa on tunnistettu tarve käytettävyyden kehittämiseksi, mutta mahdollisia rakenteita ei ole vielä olemassa. Tämä on yleensä todella tärkeää monissa suurissa organisaatioissa, joissa tuotteen käytettävyys on yksi tärkeimmistä tuotteen ominaisuuksista. Tällaisia yrityksiä ovat yleensä teknologiatuotteet, esim. Nokian GSM-puhelimet. Käytettävyys on yksi tärkeimmistä asioista, koska tuote ei toimi, jos sitä on vaikeata käyttää.

Näitä periaatteita noudattamalla järjestelmästä uskottiin saatavan hyvä ja helppo. Netissä oli monia suomalaisiakin järjestelmiä, jotka olivat keskitasoisia järjestelmiä, mutta uskoimme saavamme näin kehitettävästä palvelusta paremman. Mielestämme olemassa olevat karttapalveluiden heikkoudet olisi korjattu käytettävyydeltään ja käyttöliittymiltään paljon paremmiksi. Näin ollen tämä parantaisi järjestelmän käyttömukavuutta ja ennen muuta toisi järjestelmälle uusia käyttäjiä sekä saisi näkyvyyttä maailmalla. Käytettävyyteen kannattaa panostaa ja siihen on erityisesti panostettava, kun suunnitellaan julkista tietojärjestelmäpalvelua, kuten työssä kehitettyä karttapalvelua. (Don't make me think - A common sense approaching to web usability 2006.)

4 TUTKIMUKSEN TOTEUTUS

4.1 Työn aloittaminen

Työ aloitettiin toimeksiantajan kanssa alkuvuodesta 2008. Työ aloitettiin suunnittelemalla työn toteutusta projektin Oliomallinnus-tekniikkaa käyttäen, suunnittelemalla projektin aikataulua ja jakamalla projektia pienempiin tehtäviin. Tein työstä tehtiin suunnitelmat, jotka tarkastettiin toimeksiantajan kanssa ja niihin tehtiin parannuksia ehdotusten mukaisesti, kunnes suunnitelma oli tarpeeksi selkeä ja hyvä esitettäväksi projektiin halutulle pilottiasiakkaalle. Työn aikana pidettiin projektipalavereja, jossa päätettiin seuraavat tavoitteet projektille, jotka oli saavutettava, ennen kuin seuraava tavoitepalaveri projektista pidettäisiin. Palavereissa sovittiin uusia tavoitteita ja toiminnollisuuksia järjestelmään ja kerrattiin jakson aikana toteutetut asiat, kuinka hyvin nämä oli toteutettu, jäikö jotain kesken ja onko aikataulussa pysytty. Kun tavoitteet olivat valmiina, pidettiin uusi tavoitepalaveri, jossa käsiteltiin järjestelmän kehittämiseen liittyviä asioita ja kerättiin uusi tehtävälista, joka oli seuraavana tavoitevälietappina tarkoitus toteuttaa. Tällä tavalla järjestelmää vietiin koko prosessin ajan eteenpäin. Koko projektin aikana oli ideana suunnitella järjestelemää kuntia ajatellen, koska kyseisen järjestelmän asiakasryhmänä olivat kunnat ja järjestelmälle oli jo ensimmäinen pilottiasiakas.

4.2 Esitutkimus

Esitutkimusvaiheen toteuttaminen oli tässä kehitysprojektissa helppoa, koska työssä oli jo valmis asiakas projektin alusta alkaen. Tämä helpotti järjestelmän kehitystä jo esitutkimusvaiheesta alkaen, koska tiedettiin, että järjestelmää suunniteltiin Uuraisten kuntaa silmällä pitäen, ja näin ollen oli selvä käsitys, minkälaiselle asiakkaalle kyseinen järjestelmä tulisi käyttöön. Tämän lisäksi oli jo hyvä mielikuva siitä, mitä he haluaisivat järjestelmältä. Näin ollen saatiin hyviä näkökulmia ja ideoita, minkälainen karttajärjestelmä olisi paras kuntaa ajatellen. Myös järjestelmän tulevia asiakasryhmiä ajatellen nämä olivat korvaamattomia tietoja ja kokemuksia, koska tarkoituksena Codewisella oli saada tuotetta muillekin Suomen kunnille.

Järjestelmän suunnittelu- ja iteraatiovaihe aloitettiin tammikuussa 2008. Aluksi järjestelmästä tehtiin kuvallinen demo, jolla voitiin havainnollistaa, miltä lopullinen

järjestelmä mahdollisesti näyttäisi kuvan muodossa. Näin ollen asiakas saisi jonkinlaista ideaa ja ajatusta siitä, miltä kyseinen järjestelmä tulisi näyttämään. Tuottajalla olisi myös jotain esiteltävää järjestelmästä, joka kuvaisi tuotetta paremmin kuin ideologia kyseisen järjestelmän taustalla. Kuvallisen demon aikana tutkittiin myös teknisiä ratkaisuja, joita olisi mahdollista ja järkevää toteuttaa. Lisäksi toivottiin, että asiakaskin osaisi kuvallisen demon jälkeen helpommin ideoida, miltä järjestelmä näyttäisi ja antaa omia mielipiteitä järjestelmän kehittämisestä.

Teknisen mahdollisuuksien yhteydessä tutkittiin myös eri mahsup-rajapintojen välisiä haasteita, mahdollisia eroavaisuuksia ja kaikkien mashup-rajapintojen mahdollisia vahvuuksia valinnan suhteen. Näiden ratkaisujen aikana otettiin huomioon kaksi kattavinta karttamashupin tarjoajaa, sekä näiden vahvuudet ja heikkoudet. Kumpikin mashup-rajapinnan tarjoaja oli iso ja näkyvä IT-yritys, joka voisi tuoda monella tapaa järjestelmälle näkyvyyttä, eroavaisuutta karttajärjestelmien tarjoavien yritysten massasta sekä paljon uusia mahdollisuuksia kehitettävyyden ja laajennettavuuden kannalta.

Näitä asioita tutkittiin yhdessä toimeksiantajan kanssa, ja palaverissa tuotiin esille vahvuudet kaikista mashup-tuottajista sekä mahdollisten mashup-rajapintojen vahvuudet. Näin lopulta päädyttiin kahteen mahdolliseen palveluntarjoajaan. Jatkosuunnitelmasta karsittiin oma karttapalvelin, koska tämä nähtiin liian suurena riskinä ja aivan liian isona haasteena järjestelmän kehityksen kannalta. Suurimpana riskinä ja haasteena tässä olisi myös kyseisen järjestelmän toimivuuden luotettavuus varsinkin, kun olemassa olevilla ilmaisilla kartta-mashupeilla pystyttiin toteuttamaan yhtä hyvin ja luotettavimmin kyseisiä palveluita. Niiden etuna oli, että rajapintojen takana oli paljon enemmän palvelintehoa, jolloin luotettavuus oli paljon parempi kuin keskittymällä omaan karttapalvelimeen. Näin ollen karttapalvelu-järjestelmälle saataisiin ilmaisten palveluntarjoajien kautta paljon suurempi luottamus, joka voitaisiin esittää asiakkaille myyntivaiheessa jolloin heidät olisi helpompi vakuuttaa järjestelmän toimivuudesta ja luotettavuudesta.

Oman karttapalvelimen lisäksi karsimme pois Yahoo:n tarjoaman karttarajapinnan. Yahoo:n kyseinen karttamashup oli monipuolisuudeltaan ja kehitykseltään aivan liian suppea järjestelmän tarpeisiin ja myös järjestelmän kehitettävyyden kannalta. Varsinkin, kun tarkkaili Googlen ja Microsoftin rajapintojen mahdollisuuksia ja

lisäksi resursseja mitä tiedettiin näiden kyseisten tahojen panostavan kartta mashuppien kehitykseen. Lisäksi Yahoo:n Mapsin pois jättämiseen vaikuttivat Googlen ja Microsoftin rajapintojen mahdollisuudet ja niiden kehittämiseen panostettavat resurssit. Haasteellisimpia suunnittelussa ja karttamashup-rajapinnan päättämisessä olivat ilmaisten palveluiden lisenssit, joihin tutustuminen vei tietysti suunnittelulta oman aikansa. Kun näihin oli perehdytty tarpeeksi, tultiin siihen tulokseen, että kyseisestä järjestelmästä ei voitu tehdä suljettua järjestelmää ilman, että siitä olisi joutunut maksamaan. Toisaalta tämä ei ollut ongelma, koska järjestelmän koko idea oli olla suljettu järjestelmä. Tosin kehityksen pohjasta pystyisi helposti toteuttamaan suljetun järjestelmän, jos tähän olisi tarvetta.

4.3 Ratkaisuvaihtoehtojen pohdinta ja perustelut

Karttamashup-rajapinnan valinnassa otettiin erityisesti huomioon mashupin helppous, nopeus ja kustannukset sekä näkyvyys. Olin opintojeni aikana kehittänyt aiemmin opiskelutyönä järjestelmän Google Maps-rajapinnalle, ja puolsin tämän vuoksi Google Maps-rajapinnan käyttämistä. Perimmäinen syy oli se, että kokemukseni tulisi avuksi kehityksen aikana. Olin aikaisemmin opintojeni aikana tehnyt järjestelmän, joka pohjautuu Google Maps -mashup -rajapintaan ja tämän vuoksi osasin sitä jo hieman. Järjestelmän kehityksen aikana pidettiin karttamashup-rajapinnan vaihto avoinna muille mahdollisille karttamashup-rajapinnoille, koska se voisi olla tarpeen tulevaisuudessa. Yksi hyöty suoranaisesti olisi tässä saada lisää eroavaisuutta olemassa oleviin mashup-rajapintoja yhdisteleviin järjestelmiin, koska ne pääosin olivat Google Maps:n päällä ja tämä oli aika loppuun käytetty idea. Koska Microsoft Live Maps oli uusi, tämä voisi olla se eroavaisuus, jota tarvittaisiin näkyvyyden saamiseksi. Lisäksi joku asiakas saattaisi välttämättä haluta käyttää Microsoftin Live Maps rajapintaa tai jotain muuta vastaavaa karttamashup-rajapintaa kuin Google Mapsia.

Karttamashup-rajapintojen selvityksen ja vertailun jälkeen päädyttiin Google Maps-rajapintaan osaksi kokemukseni vuoksi, mutta myös sen vuoksi, että olemassa olevat materiaalit oli helppo muuntaa PHP-kielelle kuin vertailukohteenä olevalla Microsoftin Live Maps-karttamashup-rajapinnalla.

Kohdassa: " Järjestelmän toiminnallisuuden esittely" selvitetään tarkemmin, mistä kaikista osista järjestelmä koostuu. Tässä kohdassa on myös arkkitehtuurikuva järjestelmästä, joka kuvaa tarkemmin, kuinka järjestelmä kommunikoi mashup-rajapinnan kanssa käyttäjälle ja tietokannalle.

4.4 Vaatimusmäärittely

Projektin vaatimusmäärittelyssä oli asiakastapaamisen jälkeen tehtävänä kerätä pakolliset ja valinnaiset vaatimukset toteutettavalle järjestelmälle Uuraisten kunnan näkökulmasta ajatellen. Muutenkin kyseisen järjestelmän vaatimukset ja puitteet rakentuivat pääosin ajatellen kuntia, jotka olivat kyseiselle järjestelmälle pääasiallinen kohderyhmä, mutta järjestelmä jätettiin dynaamiseksi ajatellen muitakin tulevia asiakasryhmiä.

Toiminnalliset vaatimukset

1. Järjestelmässä tulee olla ylläpitäjän tunnistautuminen ja autentikointi (pakollinen).
2. Ylläpitäjän tulee pystyä lisäämään uusia kohteita kartalle (pakollinen).
3. Ylläpitäjän tulee voida hallita kartan kohteita (pakollinen).
4. Kohteiden hallinta täytyy voida muokata kartalta ja listasta (pakollinen).
5. Ylläpitäjän tulee voida piilottaa kohteita (pakollinen).
6. Ylläpitäjän tulee voida luoda järjestelmään uusia käyttäjiä (pakollinen).
7. Ylläpitäjän tulee voida vaihtaa salasanaansa järjestelmään (valinnainen).
8. Järjestelmän tulee lähettää uudelle ylläpitäjälle käyttäjätunnus ja salasana sähköpostilla (valinnainen).
9. Ylläpitäjän tulee voida tilata uusi salasana unohtuneen tilalle sähköpostiinsa (valinnainen).
10. Käyttäjän tulee voida selata kohteita kartalla (pakollinen).
11. Käyttäjän tulee voida valita näytettäviä hakukohteita valmiiden hakuehtojen mukaan (pakollinen).

12. Käyttäjällä tulee olla mahdollisuus ilmoittaa kohteen lisäämisestä kartalle (pakollinen).
13. Käyttäjällä tulee olla mahdollisuus ilmoittaa osoitteen virheellisyydestä (pakollinen).
14. Käyttäjällä tulee olla mahdollisuus ilmoittaa kohteen poistamisesta järjestelmästä. (pakollinen).
15. Ylläpitäjän tulee voida lisätä uusia kategorioita (valinnainen).
16. Ylläpitäjän tulee pystyä hallitsemaan kategorioita (valinnainen).
17. Ylläpitäjän tulee voida lisätä kategorioihin uusia kohteita (valinnainen).
18. Ylläpitäjän tulee voida piilottaa kategorioita (valinnainen).

Ei-toiminnalliset vaatimukset

1. Järjestelmän tulee toimia vähintään Opera 9.0, Mozilla Firefox 2.0, IE 6.0 -selaimilla (pakollinen).

Esitutkimuksessa ja projektisuunnitelmassa on määrittelty ja päätetty, että kun näistä ylläolevista vaatimuksissa pakolliset vaatimukset on täytetty, projekti katsotaan tällöin onnistuneeksi. Näin tämän projektin aikana ei täysin käynyt, mutta toimeksiantajan puolesta tehtiin päätös, että projekti on onnistuneesti hyväksytty. Syynä tälle oli, että projektin puitteissa järjestelmästä saatiin toimiva kokonaisuus ilman palvelun ylläpitopuolta. Ylläpidon osuutta pystyttiin hoitamaan myös käsin ainakin sen aikaa, kunnes järjestelmän ylläpito-osuus valmistuisi. Koska ylläpidon osuus jäi puuttumaan kokonaan, tätä kohtaa en pysty tässä laajemmin kommentoimaan ja kertaamaan.

En ole nähnyt tarpeelliseksi lisätä tähän käyttötapauksia vaatimusmäärittelystä. Näin pyrin pitämään toteutuksen dokumentoinnin mahdollisimman ytimekkäänä, jolloin se pysyy mielekkäämpänä.

4.5 Suunnittelu

Järjestelmän suunnittelu aloitettiin omien päätöksien ja asiakkaan ensitapaamisen jälkeen saatujen tietojen perusteella. Järjestelmästä tehtiin pieni esitelmän tapainen infopaketti, joka lähetettiin asiakkaalle pilottisopimuksen yhteydessä, jonka asiakas hyväksyi. Teknistä dokumentaatio suunnittelun yhteydessä tehtiin toimeksiantajalle

Codewise Oy:lle, ei niinkään loppukäyttäjäasiakkaalle, jonka kanssa pilottisopimus oli solmittu. Syynä siihen oli se, että pilottiasiakas ei tätä vaatinut järjestelmältä ja oppinnäytetyön todellinen asiakas ja toimeksiantaja oli Codewise Oy.

Suunnitelmallisuus oli selvää koko järjestelmän aikana ja järjestelmää aloitettiin kehittää EVO-ohjelmiston mallisesti, koska tämä valittiin esitutkimuksen aikana jo parhaimmaksi kehitysmalliksi tämän tyyppiselle projektille. EVO-malli on yksi ohjelmistotuotannon malli. Erona siinä yleisimpään vesiputousmalliin on se, että siinä mennään prototyyppi-mallille luontaisesti koko putki loppuun saakka eikä palata korjaamaan vaiheita. Erona tässä prototyyppi-malliin on, että EVO-mallilla toteutettaville järjestelmille kerätään N-kappaletta toiminnallisuuksia ja ne määritellään, suunnitellaan, toteutetaan, testataan ja otetaan käyttöön järjestelmään, ja näin ollen tulee uusi versio järjestelmästä, kunnes seuraavassa päivityksessä tehdään samat vaiheet uudestaan. Syy tälle valinnalle oli, että järjestelmä olisi todellakin pilottiprojekti ja näin ollen prototyyppi-kehitysmalli oli todella lähellä EVO-ohjelmistokehitysmallia. Kehittämisen kannalta ei ollut aivan selvä, minkälainen lopullinen järjestelmä tulisi olemaan, minkä vuoksi versioinnin ja riskienhallinnan kannalta tämä kehitysmalli olisi paras. Myös ajankäytön vuoksi nähtiin parhaaksi tehdä käyttäen tätä ohjelmistokehitysmallia, koska näin ollen projektimallin mukaisesti ei tarvitse palata taaksepäin korjaamaan tuotoksia, kuten vesiputousmallille on ominaista.

4.6 Tekniikat ja käytänteet

Esitutkimuksesta asti oli selvää, että järjestelmä tultaisiin toteuttamaan käyttäen PHP-ohjelmointikieltä. Hyödyt ja asiat, jotka saivat tähän päätymään, oli oma kokemukseni kyseisestä kielestä, joka teki kehityksestä helpompaa. Toinen asia oli ohjelmointikielen monipuolisuus ja sen soveltuminen varsinkin www-järjestelmiin. Yksi tärkeä ja keskeinen syy tekniikan käyttämiseen oli sen hinta, eli tekniikka oli ilmainen. Toimeksiantajan puolesta tuli ehdotuksena projektin kehitykseen, että kehityksessä käytettäisiin jonkinlaista kehitysrunkoa, joka nopeuttaisi kyseisen järjestelmän kehitystä ja jonka ansiosta saataisiin toteutettua helpommin järjestelmälle suunniteltu modulaarinen rakenne.

Tämän ansioista koko järjestelmä saataisiin helpommin toteutettua ja myös pysyttäisin järjestelmän kehityksen aikana tavoitteessa luoda mahdollisimman laajentuvainen ja helposti kehitettävä järjestelmä myyntituotteeksi. Yhtenä tehtävänä oli etsiä mahdollisia kehitysrunkoja sen lisäksi, mistä toimeksiantajalla oli jo vankka kokemus monen aikaisemman projektin ajalta. Loppujen lopuksi päädyttiin käyttämään CodeIgniter- nimistä kehitysrunkoa. Tässä on pysytty koko kehityksen ajan ja tämä on huomioitu koko kehityksen ajan.

4.7 Ensimmäisen prototyypin toteutus

Toteutusvaihe jouduttiin aloittamaan hieman suunnittelun vaiheen päälle, koska asiakkaalla, Uuraisten kunnalla, oli halu päästä näyttämään järjestelmää Rakentaminen ja talotekniikka -messuilla Jyväskylän Paviljongissa. Ensimmäiseksi rajapyykiksi asetettiin messut, johon mennessä piti olla järjestelmän ensimmäinen prototyyppi valmis. Vaatimus saatiin täytettyä, vaikkakin tämä prototyyppi järjestelmästä oli täysi raakile nykyisestä valmiista järjestelmästä.

Järjestelmä toteutettiin todella tiukalla aikataululla, mikä karsi myös hiomiselta aika paljon aikaa. Tämän seikan vuoksi järjestelmä oli ainoastaan toimiva demo tulevasta järjestelmästä, jota pystyttiin vain toimintojen perusteella kuvaamaan valmiiksi järjestelmäksi. Järjestelmän kartalla kyllä näkyi prototyyppi demoon asiakkaalta haluamat asiat. Prototyypin kehityksen suurimmaksi haasteeksi oli saada järjestelmästä luvattu demo valmiiksi Rakennus ja talotekniikka -messuille 2008, johon asiakas halusi saada järjestelmän näytteille. Tämän haasteen lisäksi järjestelmä oli täynnä virheitä ja bugeja, mikä johtui vain siitä, että aikaa ei ollut tarpeeksi hioa tätä kyseistä demoa valmiiksi.

Tämän vuoksi messuille toimitettu järjestelmä oli aika lailla itsenäisesti tehty demo järjestelmästä, mikä ei vastannut suunniteltua järjestelmää muuta kuin käyttöliittymätasoltaan. Kooditasolta tarkasteltuna järjestelmä oli hyvin erilainen. Voisi sanoa, että tämän ensimmäisen prototyyppiversion jälkeen järjestelmästä alettiin tehdä kokonaan uusi. Tämän aikana runko, kehitysrungot luotiin uudestaan, koska toimitettu demoversio oli aika lailla huono ja jotain piti tehdä eri lailla. Ensimmäisen prototyypin aikana järjestelmästä saatiin PHP-koodin rungoltaan sellainen, mitä lopullinen järjestelmä edustaa. Koska Google Maps-mashup-rajapinta käyttää

Javascript-ohjelmointikieltä, niin tämä piti saada korjattua niin, että se toimisi mahdollisimman dynaamisesti. Lisäksi Javascript-osuus demojärjestelmästä oli aika staattinen, minkä vuoksi sen kehittäminen tässä muodossaan olisi ollut todella vaikeata.

4.8 Järjestelmän kehittyminen

Ensimmäisen sovitun rajapyykin jälkeen pidettiin palaveri järjestelmän jatko-suunnittelusta, minkä jälkeen järjestelmä muuttui aika paljon tähän asti saadusta prototyypiversiosta. Prototyypiversiossa oli käytetty tekniikoina PHP CodeIgniter -kehitysrunkoa ja melkein staattista Javascript-tiedostoa. Seuraavaan versioon Javascript hoidettiin MVC-arkkitehtuuri mallisesti JavaScript MVC -kehitysrunkoa hyväksi käyttäen. Näin järjestelmästä saatiin paljon dynaamisempi ja helpommin laajentuva tulevaisuuden kehitystä ajatellen. Myös näiden tekniikoiden avulla latausajat pienenivät huomattavasti, jolloin myös käyttö parani käyttäjän kannalta huomattavasti. Nyt saatiin järjestelmää optimoitua paremmin kuin tähän mennessä.

Suurin osa uudelleen kehittämisestä meni tämän uuden JMVC-järjestelmän opettelemiseen, koska järjestelmän kehittäminen oli aloitettu vasta marraskuussa 2007. Järjestelmän uutuuden vuoksi dokumentaatio oli todella vähäistä ja suppeata. Tämä vaikeutti huomattavasti opettelemista, sillä monista asioista ei ollut minkäänlaista dokumentaatiota. Tässä kohdassa pystyin käyttämään omaa ohjelmistokehityskokemustani hyödyksi. Tämä vei huomattavasti suunniteltua enemmän aikaa eikä sitä oltu osattu ennakoida suunnittelussa, koska tekijäryhmällä ei ollut aikaisempaa kokemusta pilottiprojekteista.

4.9 Ylläpito

Julkinen osuus saatiin valmiiksi syyskuuhun 2008 mennessä. Tämä valmistui projektin resurssien puitteissa, vaikka aikataulullisesti oli ollut tarkoitus saada myös järjestelmän ylläpito puolivalmiiksi. Julkisen osuuden valmistuttua katsottiin toimeksiantajan kanssa parhaaksi, että oma osuuteni tutkimustyöstä ja projektin toteutuksesta oli kuitattu valmiiksi. Tähän tulokseen päädyttiin, koska olin käyttänyt järjestelmän suunnitteluun monin kerroin enemmän aikaa, kuin mitä oli suunniteltu. Tämän vuoksi olin ylittänyt tutkimukseen varatun ajan.

Ylläpidollisesti kyseinen sovellus jäi paljolti avoimeksi ja tältä osin kesken. Tämä johtuu myös siitä, että sovelluksen kehittämiseen meni suunniteltua enemmän tunteja. Asiaa myös vaikeutti se, että sovelluksesta oli tarkoitus tehdä mahdollisimman moduläärinen ja dynaaminen, mikä vaikeutti toteutusta ja suunnittelua. Lyhyesti koostettuna vaatimukset täytyi toteuttaa niin, että tuote olisi jatkojalostettavissa nopeasti ja helposti uusien asiakkaiden käyttöön. Tämän vuoksi kaikki koodi piti olla mahdollisimman dynaamista, jolloin käyttöönotto helpottuisi ja olisi mahdollisimman nopeata. Näin ollen ylläpito jäi kokonaan tekemättä, ainakin omalta osaltani. Tätä en kerinnyt tekemään tämän tutkimustyön teknisen osuuden aikana. Ylläpito-osuuden toteuttaminen jäi järjestelmän jatkokehityssuunnitelmaksi ja näin ollen avoimeksi asiaksi.

4.10 Testaus

Testauksesta ei tehty omaa vaihetta tai erillistä testaussuunnitelmaa. Tämä karsittiin pois sen vuoksi, että aika oli rajallinen kehityksen kannalta ja suunniteltu aika oli ylittynyt projektin osalta. Testaus oli lähinnä kehittäjän tekemää ohjelman testausta, jossa tarkoituksena oli katsoa, toimivatko tehty toiminto sekä muut toiminnot. Todellinen testaus oli tarkoitus toteuttaa loppukäyttäjällä pilottiprojektin julkistamisen yhteydessä eli suorittaa ns. ”beta-testaus”, josta saadaan todellisia loppukäyttäjien kokemuksia, kommentteja ja virheitä, jotka korjataan sen jälkeen järjestelmään. Näin ollen saadaan hyviä ja tuoreita ideoita järjestelmän kehitystä ajatellen. Suurena riskinä tässä on myös, kuinka pidetään järjestelmä mahdollisimman vakaana asiakkaalle asennettavassa versiossa. Tämä vaatii paljon suunnittelua ja myös toimintojen testausta ennen asennusta. Tämä saataisiin vähemmällä työllä aikaan, jos olisi tehty erillinen testaussuunnitelma ja mahdolliset systemaattiset testaukset järjestelmälle. Näin löytyisi kasapäin mahdollisia virheitä, nopeusongelmavirheitä ja muita ongelmia.

4.11 Karttasovelluksen haasteet

Suurin haaste karttasovelluksen kehittämisessä oli selvästi uusi käytettävä tekniikka ja teknologia, jonka kanssa toimittaessa tulikin yllättäviä tilanteita, joihin ei suunnittelun aikana oltu osattu tarpeeksi varautua. Haasteen tasoa nosti suunnittelun aikana kehitetty ajatus sovelluksen hyödyntämisestä jatkossa, minkä vuoksi kehittämisessä ei voitu hyödyntää niin suoranaista elinkaariajattelua kuin räätälöidyssä ohjelmistossa

yhdeksi asiakkaalle. Tässä tapauksessa oli ajateltava, miten mahdollisimman helposti järjestelmää saataisiin monistettua uusille asiakkaille. Tässä vaatimuksena oli tietysti järjestelmän dynaamisuus, jotta sen jatkotuottaminen eri asiakkaille olisi mahdollisimman helppoa. Tämän vuoksi pelkästään oikean ohjelmistokehitysalustan löytäminen kesti useamman kuukauden, joka osaltaan hidasti suunnitteluvaiheesta toteutusvaiheeseen siirtymistä.

Myös alkuperäinen kehitystapa huomattiin puutteelliseksi järjestelmän kehitysvaiheessa, minkä vuoksi toteutuksessa tuli paljon uutta opittavaa ja muutosta. Tämän tapaisista projekteista ei ollut ollenkaan aikaisempaa kokemusta ja tämän vuoksi oli tärkeitä pitää tarkata linjat projektinhallinnallisesti koko projektin ajan. Resurssien vuoksi projektinhallinta tämän projektin kohdalla koki ongelmia, koska projektinhallintaa ei oltu tarpeeksi resursoitu. Tämä voi olla yksi osasy, minkä vuoksi projekti viivästyi alkuperäisestä suunnitellusta aikataulusta. Projektin myöhästymiseen voi olla syynä myös se, että koska kokemusta tämän tapaisista järjestelmien kehittämisestä ei ollut ennestään, ei työaikaarviointeja oltu osattu tehdä tarpeeksi tasapainottamaan aikataulupaineita projektin kohdalla. Lisäksi yllättävän suureksi haasteeksi muodostui Javascript-osio järjestelmän kehityksessä, mikä omalta osaltaan yllätti aika paljonkin. Suurimpana syynä oli se, että myös Javascript-osio piti saada mahdollisimman oliopohjaiseksi tai muuten helposti käytettäväksi ja kehitettäväksi. Näin ollen Javascriptin osalta aikaa meni paljon uuden tekniikan opiskelemiseen.

5 TULOKSET JA JOHTOPÄÄTÖKSET

5.1 Järjestelmän toiminnallisuuden esittely

Kehittämiprojektin aikana kehitetty web-järjestelmä on pääasiallisesti suunniteltu Suomen kuntia silmällä pitäen. Palvelun tarkoituksena on luoda oma portaali kuntalaisten käyttöön nykyisille ja tuleville asukkaille, jossa on mahdollista kuvata kunnan alueella olevat palvelut, yritykset ja mahdolliset muut palvelut, kuten esimerkiksi tonttien sijainnit kaikkineen mahdollisuuksineen. Ensimmäisessä vaiheessa oli tarkoitus ainoastaan piirtää tontit koordinaatteineen ja lisätietoineen sivustolle.

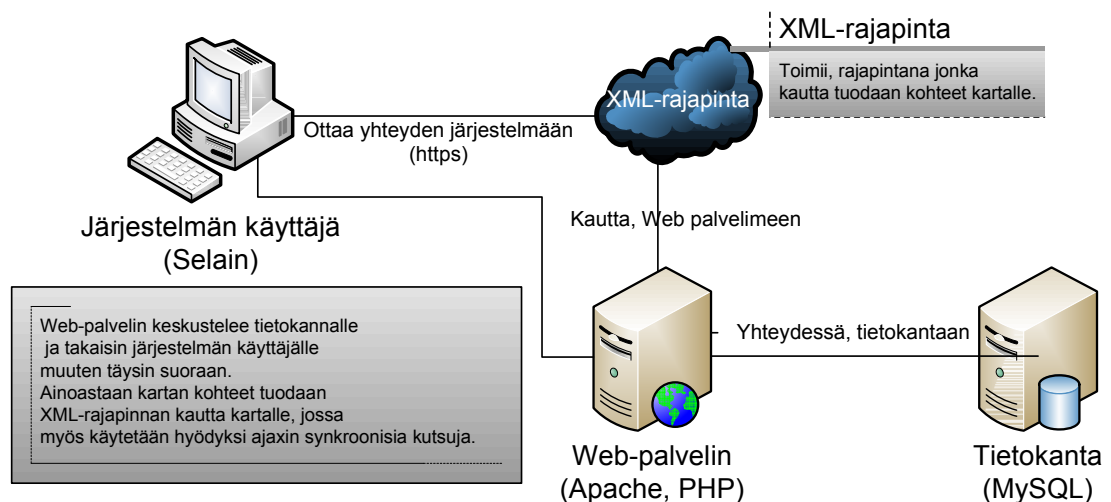
Uusille asukkaille tämä portaali on hyvä tiedonlähde kertomaan, mitä palveluita kunnilla on tarjolla, sekä myös näkemään välimatkat omalta tontilta tai talolta palvelujen luo. Näin ollen perheelliset näkevät, kuinka lähellä on koulu, kauppa, kirjasto tai mikä muu palvelu sattuu kiinnostamaan. Myös itse kunta voi parantaa tällaisen palvelun avulla näkyvyyttään ja imagoaan. Markkinoilla ei ole olemassa tämän tapaista sopivaa palvelua, joka olisi kohtuullisen hintainen kunnille.

Kehitetty karttapalvelu hyödyntää yleisiä parhaaksi nähtyjä Web 2.0 -tekniikkoja, teknologioita ja käytänteitä. Tämän vuoksi teknologian puolesta asiakkaille ei tule tästä minkäänlaisia lisäkustannuksia, vaan tämän palvelun voi integroida asiakkaan nykyiseen sivutilaan. Palvelu on monella tapaa helposti ja nopeasti saatavissa uudelle käyttäjälle suunnittelun ansiosta. Järjestelmä on kehitetty niin, että mahdolliset uudet kehitystyöt järjestelmään ovat helposti tehtävissä, jolloin jokaiselle kunnalle tai yritykselle palvelu on muokattavissa täysin tarpeiden mukaan, mikä tekee järjestelmästä joustavan moneen suuntaan. Dynaamisen ja modulaarisuuden takia myös kyseiset mainostyöt on mahdollista hoitaa todella helposti ja nopeasti. Järjestelmässä käytetty karttamashup-rajapinta on mahdollisimman helppo vaihtaa toiseen, jos tälle on erityinen tarve. Näin ollen tätä järjestelmää ei ole millään tapaa sidottu tiettyyn karttamashup-rajapintaan, vaan mahdolliset muutokset ja päivitykset tältä osin on otettu jo alusta asti huomioon. Tätä en todellakaan näkisi huonona asiana. Kyseinen järjestelmä käyttää Google Maps -karttaa, jonka päälle tuodaan eri

elementtejä, mikä tekee tästä paljon interaktiivisemman palvelun, jolla käyttäjä saa palvelun käytöstä miellyttävän kokemuksen.

Miksi tällainen toteutettiin? Huomattiin, että internet on muuttava alusta ja tämän tapaiselle web-järjestelmälle oli käyttöä varsinkin kunnissa yms. yhteisöissä, ja se hyödyntäisi itse kuntaa ja myös sen asukkaita. Tästä hyvinä esimerkkeinä ovat useat järjestelmät, jotka ovat soveltaneet Web 2.0:n mukanaan tuomaa sosiaalisen median käytäntöä, joka toimii myös tässä järjestelmässä jollain tapaa. Tämän perusteella on hyvä saada kunnalle web-järjestelmä, joka toimisi hyvänä tietolähteenä palveluista, nähtävyyksistä, tonteista yms. asioista kunnan asukkaille ja myös turisteille. Järjestelmän kehityksen kannalta tämä mahdollinen järjestelmä pystyisi laajenemaan vaikka kuinka laajaksi portaalijärjestelmäksi, johon olisi mahdollista integroida kunnan uutisia, tulevia tapahtumia, tiedotuksia ja jne.

Alapuolella on arkkitehtuurinen kuvio, jossa on kuvattu, kuinka kommunikaatio toimii käyttäjältä web-palvelimelle ja sieltä takaisin. Kuviossa näkyvät myös järjestelmään liittyvät osat ja komponentit, jotta se hieman selventäisi, miten tämä järjestelmä on rakennettu.



KUVIO 1. Järjestelmän arkkitehtuuri

5.2 Saavutukset

Tutkimus- ja kehitystyöprojekti, jossa tutkittiin Web 2.0-tekniikoita, rakentui karttamashup-rajapintojen ympärille. Pähkinäkuoressa projektissa oppimani asiat ovat:

Lähtiessäsi tutkimaan, kuinka toteuttaa massiivinen portaalijärjestelmä, varaa tutkimuksen esitutkintaan paljon aikaa varsinkin silloin, jos tulet toimimaan tuntemattoman asian, kuten mashup-rajapintojen yhteydessä. Yhtenä suosituksena tämän tapaisissa tutkimusprojekteissa olisi käyttää ohjelmistonsuunnitteluun mallina 'Nopeita metodeja'. Näiden menetelmien ansiosta tutkimus- ja kehitystyössä ei tulisi niin paljon pullonkauloja ja mahdollisesti projekti pysyisi paremmin aikataulussa. Vaikka tässä on omat riskinsä, tämä kuitenkin nopeuttaisi asiana etenemistä niillä osaluilla, joilla yleisesti tiedetään tulevan pullonkauloja ajankäytön kanssa.

Tutki laajemmin alan kirjallisuutta aiheesta, koska kirjoissa on yleisesti todella paljon vinkkejä kyseisestä tekniikasta, teknologiasta tai tässä tapauksessa Web 2.0 -ideologiasta ja ilmiöistä. Kirjat auttavat etenemään asiassa paljon paremmin kuin internet-lähteet, koska näissä tieto on kirjoitettu yleensä yksiselitteisesti, ja tieto on kirjoissa myös paljon turvallisempaa kuin internetissä. Internetissä on myös yleisesti todella paljon asiaa, mutta tämä tieto ja materiaali on yleensä ripoteltu ympäriinsä, ja tämän vuoksi kirjasta saa parhaiten irti sen tiedon, jota tarvitsee projektin läpi viemiseksi.

Päätä ajoissa kehitysrunko toteutettavalle ohjelmistokehitykselle. Tämä on vaikeampaa kuin uskoisi. Jos projekti on ainutkertainen, eli ei ole tarkoitus monistaa tuotetta eteenpäin tai jos tarkoitus on monistaa tuotetta, muistakaa kehityksen aikana myös panostaa projektin hallinnointiin, koska tämä on tunnettu pullonkaula tämän tapaisessa ohjelmistokehitystyössä.

Uskon, että näillä muutamalla vihjeellä projektin läpivienti helpottuu paljon. Itse opin ennen kaikkea, että projektissa suunnitelmat ja määritykset ovat tärkeimpiä. Tätä yleensä vähätellään, kuten tämänkin kehityksen aikana, mutta monikin toteutettava asia olisi ollut helpompaa, jos se olisi suunniteltu perin pohjin. Projektin luonteesta ja aikataulusta johtuen suunnitelmia ei tehty tarpeeksi huolellisesti.

Suunnitelmat ja määritykset lykkääntyivät, joten projektia piti toteuttaa alkuperäisten suunnitelmien ja palavereissa asetettujen tavoitteiden mukaisesti.

5.3 Vastaukset tutkimuskysymyksiin

Tässä luvussa tarkennetaan vastauksia tutkimuskysymyksiin.

1. Mihin asioihin suunnittelussa pitää kiinnittää huomioita toteutettaessa mashup-tekniikkaan perustuva järjestelmä?

Moniin asioihin täytyy panostaa, kun aloitetaan määritellä mashup-tekniikkaan perustuvaa järjestelmää. Ensimmäinen asia on ehdottomasti se, minkälaiseen mashup-tekniikkaan perustuvaa järjestelmää ollaan suunnittelemassa. Suunnittelussa tärkeintä on määrittää järjestelmän vaatimukset ja lyödä ne lukkoon heti. On selvittävä, onko tarvetta tehdä oliopohjainen järjestelmästä vai ei. Oma kehitystyötäni olisi nopeuttanut huomattavasti, jos järjestelmästä ei olisi tarvinnut tehdä dynaamista ja oliopohjaista javascript-osuuden puolelta, ja järjestelmä olisi saatu alustavassa aikataulussa valmiiksi. Laajennettavuuden kannalta tämä oli yksi pakollisista vaatimuksista, koska muuten tätä ei olisi saanut tehtyä.

Seuraavana suunnittelussa on ehdottomasti käytettävän mashup-rajapinnan valitseminen. Jos intoa on opiskella eikä pelkää haasteita, ei kannata ehdottomasti valita Google Mapsia, jos valinnan kohteena on kartta mashup-karttarajapinnat. Microsoftin Live Maps ei nimittäin ominaisuuksissa ja kehittyvyydessä jäänyt yhtään jälkeen Googlen pidemmästä kehitysiästä. Ainoana todella pienenä ongelmana ja haasteena kehitysprojektille olisi osoittautunut ainoastaan Microsoftin omien ohjelmistotekniikoiden käyttäminen ja tähän liittyvä opetteleminen, joka sekään ei ole välttämätöntä. Kyseisen mashup-rajapinnan saa toimimaan aivan yhtä hyvin PHP-ohjelmointikielellä, vaikka tähän ei suoranaisia ohjeita ainakaan internetistä löytynyt. Nykyään aiheesta löytyy enemmän kirjallisuutta kuinka toteuttaa Web järjestelmä käyttäen Microsoftin Live Maps rajapintaa, jos ohjelmointi kielenä on muu kuin .NET tekniikka. Haasteena on opetella itse tätä asiaa, koska valmiita esimerkkejä ei PHP-kielelle ole niin paljon kuin Google Maps:lle.

Kolmantena tärkeänä asiana pidän XML-puskurirajapinnan. Jos aikomuksena ei ole tehdä täydellistä järjestelmää, jossa voidaan tehdä karttamashup-vaihdos helposti, suosittelen, että kyseistä XML-puskurirajapintaa ei tehdä järjestelmään. Tosin XML on kieli, jota kielellä kuin kielellä voidaan lukea ja jäsennellä. Lisäksi on väitetty, että karttamashup-rajapintojen kanssa toimittaessa tiedot kartalle pystyttäisiin lukemaan nopeammin XML-tiedoston kautta. Tietokantajärjestelmät ovat kuitenkin yhtä nopeita, ellei järjestelmässä ole kohteita useita tuhansia ja käyttäjiä satoja. Mielestäni järjestelmä on ilman XML:ää hieman tietoturvalisempi, vaikka kohteissa ei yleensä ole mitään tärkeää ja arkaluontoista tietoa.

Näitä pitäisin tärkeimpinä kohteina, joihin kannattaa paneutua ennen suunnittelun eteenpäin viemistä. Omassa kehitysprojektissa opin näiltä osin paljon, koska kehitysrunko vaihtui projektin aikana useamman kerran, jolloin järjestelmästä tavallaan tuli useampi versio, ennen kuin se muotoutui omaan viimeiseen malliinsa. Tämä olisi voitu välttää hieman paremmalla suunnittelulla, johon olisi tarvinnut teknisen arkkitehdin eikä pelkän ohjelmistosuunnittelijan näkemystä järjestelmän kehittämisen kannalta. Näin myös kehitykseen suunniteltu aikataulu olisi pitänyt.

2. Miten mashup-tekniikkaan perustuvan palvelun käyttöliittymästä saadaan käytettävyydeltään mahdollisimman selkeä?

Suunniteltaessa mashup-tekniikkaan perustuvaa palvelua korostuu suunnittelun aikana se, että palvelun käytettävyyden ja käyttöliittymän palvelulle pitää olla mahdollisimman yksinkertaisia. Myös nämä mashup-rajapinnat mahdollistavat todella graafisesti monipuolisesti näyttäviä käyttöliittymiä, jotka tekevät käyttöliittymistä todella upean näköisiä. Tämä ei suoranaisesti vaikuta palvelun käytettävyyteen, mutta vaikuttaa käyttäjien käyttömieltyyksiin, jotka ovat täysin psykologisia asioita. Väitän, että käyttäjät mieltyvät käyttämään palvelua, joka on hienon ja ammattimaisen näköinen, kuin käyttöliittymää, joka näyttää lähinnä 90-luvun alussa tehdyiltä internet-sivustolta.

Toinen asia, joka suoranaisesti vaikuttaa käytettävyyteen käyttöliittymän lisäksi, on palvelun nopeus. Tähän asiaan kannattaa panostaa jo suunnitteluvaiheessa, jotta palvelu saataisiin optimoitua mahdollisimman hyvin. On olemassa monia vinkkejä ja tapoja, kuinka tätä nopeuttaa ja näitä löytyy palveluntarjoaja- ja tekniikkapohjaisesti

monia. Ei kuitenkaan kannata panostaa liikaa aikaa vain optimoimiseen, jos kyseessä on nopeudensäätö vain jossain sekunnin sadasosissa.

Kannattaa kiinnittää huomiota siihen, mitä palvelun käynnistettäessä ladataan käyttöliittymään esille. Kannattaa myös miettiä, onko tarvetta ladata kaikki kohteet vai vain osa, jolloin käyttäjät itse pystyisivät lajittelun perusteella tuomaan enemmän asioita esille, jos tahtovat. Käytetään ns. porrastusta, jolloin palvelin ei kuormitu liiaksi ja näin ollen hitaimmillakin yhteyksillä voidaan palvelua käyttää kohtalaisella nopeudella, koska tätä pystytään nykyään tekemään todella helposti AJAX-tekniikan tuomien lisäavustuksien ansiosta. Lisäksi kohteiden graafiset hienoudet on hyvä hienosäätää, jotta esimerkiksi varjostukset kannattaa jättää pois lataamisesta, sillä se nopeuttaa esim. palvelun latautumista. Tämä tietysti vaatii sen, että kyseiseen tekniikkaan on tutustuttava perin pohjin, jotta tämän tapaisia vinkkejä voi hyödyntää optimointivaiheessa.

Google Maps-mashupin kohdalla törmäsin artikkeleihin, joissa väitettiin, että kun kohteet ladataan XML-tiedostosta, kohteet latautuvat nopeammin. Tästä olen eri mieltä ja uskon, että kohteet latautuvat yhtä nopeasti tietokannasta kuin XML-välitiedoston kautta. Tämä kannattaa suunnitella palveluun ainoastaan, jos sille on jotain erityistä käyttöä, esimerkiksi eri järjestelmistä asioiden tuomista omien rajapintojen kautta tai jotain vastaavaa. Muutoin suosittelen tekemään tietojen tuomisen suoraan tietokantahakujen kautta. Näin pystyy latausaikaa nopeuttamaan paljon. Käytettävyyteen kannattaa panostaa, koska monilla pienillä asioilla saa järjestelmän käytettävyyttä parantamaan paljon.

3. Kuinka ilmaiset mashup-rajapinnat eroavat tosistaan?

Ilmaisten mashup-rajapintojen välillä on monia ominaispiirteitä, jotka tekevät niistä erilaisen kuin toisesta. Käsitteenä ilmaiset mashup-rajapinnat on todella laaja ja kattaa kaikki olemassa olevat ilmaiset rajapinnat, joita voi käyttää. Tämän vuoksi tähän vastaukseen rajaan vain ilmaiset karttarajapinnat, jotka ovat lähempänä omaa tutkimustani ja kehitystyötä, jossa olen mashup-rajapintojen kanssa toiminut. Karttarajapintojen lisäksi on olemassa useita muita mahdollisia mashup-rajapintoja, kuten esimerkiksi Facebook-ohjelmistorajapintoja, RSS-syöterajapintoja, CMS-rajapintoja. . RSS-syötteillä tarkoitetaan sivustoja, jotka tarjoavat esim. uutisensa

luettavaksi jonkinlaiseen lukuohjelmaan tai sivustoon. Näistä ehkä suosituin on Googlen Reader-sivusto, johon voi tilata useita uutis- sivustojen uutisia, ja jossa uutiset voi lukea tekstimuodossa. Tämä mahdollistaa myös Blogi-tekstien tilaamisen kyseiseen palveluun. CMS-rajapinnat taas liittyvät sisällön tuottamiseen ja näiden järjestelmien integroimisen muihin järjestelmiin.

Karttarajapinnoista vertailukohteena pidin työssä pidettiin Google Maps-rajapintaa ja Microsoft Live Maps -rajapintaa, joita vertailtiin mahdollisuuksiltaan, laajentuvaisuudeltaan, lisensseiltään ja kehitysalustaltaan. Syynä tälle oli se, että järjestelmä piti saada kehitettyä mahdollisimman halvalla, helposti ja uusimmilla tekniikoilla sekä tavoilla toteuttaa web-palvelua. Tähän sopivimpina mashup-rajapintoina kävivät selvästi suurimmat karttamashup-rajapinnat. Näistä oli olemassa useilta eri tahoilta todella paljon hyvää käytännön kokemusta. Seuraavaksi listaan omien tutkimuksieni pohjalta kerätyt plussat ja miinukset näiden kahden karttamashup-rajapintojen välillä.

Google Maps

- + Helppo käyttöönottaa
- + Jokaisen saatavilla
- + Paljon olemassa olevia esimerkkejä
- + Suuri käyttäjäyhteisö
- + Hyvä ja helppo api dokumentaatio
- + PHP-ohjelmointikielelle olevat esimerkit
- + Todella laaja toimintokirjasto
- Vakiona tylsä ulkoasu
- Liikaa olemassa olevia samantapaisia järjestelmiä
- Javascript- ja Actionscript-ohjelmointikielillä tehtävä looginen liittymä

KUVIO 2. Google Mapsin plussat ja miinukset

Microsoft Live Maps

- + Helpot RSS-mahdollisuudet
- + Hyvä oma apusivusto aloittaville kehittäjille
- + Todella laaja kirjasto olemassaolevilla ominaisuuksilta
- + Luultavasti riittävän hyvä lisenssi kehittäjille, joka on hieman erinlainen kuin Googlella
- Microsoftin tarkoitus pyrkiä saamaan alustaa kehittämään omilla teknologioilla
- Apusivustoilla ainoastaan .NET-esimerkkejä
- Microsoftin tekniikoilla tehtävä kehitys, ajoympristöjen lisenssien hinnoittelu, joka voi noistaa järjestelmän kehityksen hintaa huimasti

KUVIO 3. Microsoft Live Mapsin plussat ja miinukset

5.4 Ohjelmistoprojektin arviointi

Olen todella tyytyväinen projektin lopputulokseen. Kiteytän arviointini kolmeen eri kategoriaan, jotka tässä puran tarkemmin. Ensimmäisenä avaan projektin arvioinnin ohjelmistotuotannon näkökulmasta, toisena puran, kuinka näkisin kunnan näkökulmasta katsottuna projektin parantaneen asiakasryhmälle suunniteltua projektia ja kolmanneksi arvioin sen oman oppimiseni, ammattitaidon parantamisen ja projektin ohjelmistosuunnittelijan näkökulmasta.

Projekti ohjelmistotuotannon näkökulmasta ei saavuttanut sille suunniteltua tulosta tuotostensa kohdalla eikä täysin toimintojenkaan osalta, , mutta toimeksiantajan kanssa sovimme projektin onnistuneeksi. Vaikka tuote ei ollut vielä täydellinen ja kehitettävää jäi ohjelmistotuotteen kohdalla, jotta palvelusta saataisiin täysi ensimmäinen versio. Jäi järjestelmän lisäkehityksen vastuulle, että järjestelmästä saadaan julkaistua virallinen ensimmäinen versio. Puutteen web-järjestelmälle jäi suunnitelmien ja määritysten tuottaminen sekä ylläpito-osuuden toteuttaminen järjestelmään. Ylläpito-osuus jäi toteuttamatta, koska ajallisesti tämä ei ollut enää mahdollista. Se johtui siitä, että aika, joka oli varattu järjestelmän suunnitteluun ja toteuttamiseen, oli ylittynyt, minkä vuoksi ylläpito-osuus päätettiin jättää kokonaan

pois. Tämän vuoksi toimeksiantaja katsoi vastuuni järjestelmän kehityksessä loppuneent. Kaiken kaikkiaan voin silti todeta projektin onnistuneeksi, vaikka osa tuotoksista projektin matkan varrelta jäikin toteuttamatta. Kyseinen järjestelmä kuitenkin saatiin siihen pisteeseen, että se on toimiva, vaikka jotkut asiat jäivät toteuttamatta., Ne eivät kuitenkaan ole niin kriittisiä asioita, että ne vaikuttaisivat onnistumiseen.

Projektin loppuasiakkaan eli kunnan näkökulmasta projektin lopputuotos eli karttajärjestelmä parantaa kunnan näkyvyyttä. Tiedossani ei ole, onko pilottiasiakkaan kanssa otettu kyseinen palvelu käyttöön. Kehityksen lopettamisen aikana sitä ei vielä ehditty ottaa käyttöön ja tietääkseni se ei vielä ole julkisessa käytössä kyseisellä asiakkaalla. Näin ollen ei ole mitään käyttäjäkokemuksia vielä siitä, kuinka hyvin käyttöönotto ja käyttäjien koulutus on onnistunut asiakkaan kohdalla. Itse näkisin asiakkaan näkökulmasta, että tämä olisi uusi ja innovatiivinen tapa edustaa ja markkinoida kuntaa. Uskon, että uusien järjestelmän käyttäjien on helppo omaksua kyseisen järjestelmän käyttäminen sekä hyötyä tästä paljon niin uusina kuin vanhoinakin kunnan asukkaina, jotka käyttävät kyseistä järjestelmää.

Arvioitaessa projektia oman osaamisen, ammattitaidon ja ohjelmistokehittäjän näkökulmasta pidän tätä hyvänä projektina. Suuri kiitos projektista kuuluu Codewise Oy:lle, jonka kautta sain mahdollisuuden päästä toteuttamaan näinkin innovatiivista ja mielenkiintoista aihetta. Tästä on ollut paljon hyötyä oman ammattitaidon kehittämiseksi ja se on auttanut minua uralla eteenpäin. Tärkeää oli päästä oppimiaan ja syventämään ammattiosaamistani Javascript-ohjelmointikielen osalta. Kyseisen kielen kanssa työskentelin koko projektin ajan ja tämä oppimisaika koulutti todella syvästi omaa ammattitaitoani Javascriptin osastolla. Pidän Javascript-ohjelmointikielen oppimista tärkeänä tämän projektin yhteydessä, koska kyseinen kieli on todella yleinen Web-sivustoilla ja monissa Web-järjestelmissä. Ohjelmointikieli mahdollistaa sen, mitä monet palvelinpuolen ohjelmointikielet eivät mahdollista. Ohjelmistokielen ja näiden asioiden oppimisen lisäksi pääsin oppimaan täysin uutta asiaa, jota en opintojen aikana ollut päässyt toteuttamaan, kuten esim. Web 2.0 –tekniikkaan perustuvat rajapinnat ja varsinkin mashup. Oman oppimisen kannalta tärkeänä pidän mahdollisuutta päästä toimimaan osana ohjelmistokehitystiimiä. Varsinkin tässä tiimissä toimiessani pääsin tutkimaan ja toteuttamaan todella mielenkiintoista Web 2.0 –tekniikkaan liittyvää projektia. Näin

opin toimimaan tiimissä ja sain tästä todella hyvää kokemusta, vaikka kehitysprojektin ohjelmointipäävastuu oli minulla tämän ohjelmistoprojektin osalta. Tässä tiimissä työskennellessä sain uusia näkökulmia kehitysprojektille ja uusia vinkkejä ohjelmistokehityksen kannalta. Nämä antoivat uusia ideoita itselleni ja vaikuttivat myös toteutustapoihin.

6 POHDINTA

Tässä pohdintaosuudessa tarkoituksena on pohtia asioita tutkimustyöni ajalta. Tulen ottamaan kantaa siihen, miten onnistuin ratkaisemaan haasteet ja ongelmat tutkimuksen kuluessa, rajoituksiani tutkimusmenetelmien suhteen ja sitä, kuinka näitä olisi hyvä kehittää. Lisäksi pohdin sitä, miten tämä tutkimus lisäsi tietoa tutkimusalueelta eli kuinka paljon tämä laajensi tietämystä Web 2.0:sta ja mashup-rajapinnoista. Kuinka hyvin tutkimuksen tulokset ovat yleistettävissä, ja miten tutkimusentuloksia voidaan hyödyntää teoriassa ja käytännössä? Lisäksi kerron, mitä jatkokehitysajatuksia tutkimus on tuottanut.

Tutkimuksen aikana oli paljon haasteita ja ongelmia, jotka hidastivat eniten ohjelmistoprojektin etenemistä. Suurimpana haasteena tutkimustyöni aikana oli toteutusvaihe eli ohjelmistoprojekti tutkimuksen perustalle. Tämä johtui suurimmaksi osaksi siitä, että kehittämisprojekti oli niin uusi asia ja erilainen teknologia, josta itselläni eikä toimeksiantajan puolesta ollut ennestään kokemusta, jota hyödyntämällä olisi ohjelmistoprojektin aikana tulleet ongelmit voinut minimoida. Tämän vuoksi ohjelmistoprojektin aikana tuli paljon sellaisia asioita vastaan, jotka veivät paljon arvokasta aikaa tämän kehittämisprojektiin aikana. Tämän olisi voinut ehkä jollain tapaa ennakoida ja ottaa huomioon suunnittelun aikana, mutta koska koko projekti oli luonteeltaan niin intensiivinen ja uudenlainen, ei tätä asiaa suunnittelussa osattu huomioda tarpeeksi hyvin. Toimeksiantajalla ei ollut tämän tapaisista työtehtävistä ennestään minkäänlaista kokemusta, minkä vuoksi hekään eivät voineet antaa työarviosuunnitteluun mitään ehdotuksia.

Toiseksi suurimpana vaikeutena oli se, että tein oman tutkimustyöni vähän takaperin, jätöteutin ensimmäiseksi työssäni tulleen järjestelmän, josta saatujen kokemusten ja tutkimusten pohjalta tein opinnäytetyöni. Tämä olisi ollut oikeasti järkevää tehdä niin, että olisin tutkinut teoriaa laajemmin, olisin kirjoittanut ylös hyviä tutkimuksen aikana tulleet johtopäätöksiä, ideoita ja asioita, joita olisi kannattanut miettiä tutkimuksen aikana ja vasta tämän jälkeen olisin toteuttanut itse työn teknisen osuuden. Tämä olisi auttanut tutkimuksen tekemisessä ja ajoituksessa.

Aiheesta on todella niukasti kirjoitettua asiaa olemassa, mutta se vähäkin olisi antaneet paljon paremman lähtökohdan työlle, kuten esim. siitä, miten ohjelmistoprojektissa edetään. Vaikka tein työni niin, että toteutin teknisen työn ensiksi, sain mielestäni tulokset koostettua todella hyvin ja tämä opetti tutkimustyöstä aivan eri lailla, kuin jos olisin tehnyt normaalin tutkimustyön kaavan mukaan. . Mielestäni joissain tutkimus - tapauksissa on hyvä tehdä eri lailla tutkimustyötä, kuin miten normaalien kaavojen mukaan pitäisi tutkimuksen kanssa edetä. Tämä on mielestäni opettavaisempaa ja työstä saa paljon enemmän irti. Varsinkin kun työni oli mielestäni täysin poikkeava tutkimustyö, minkä vuoksi olikin hyväksi poiketa kaavasta sen verran, että työn luonne tulisi paremmin ilmi.

LÄHTEET

Ayers, P., Mathews, C. & Yates, B. 2008. How Wikipedia Works: And How You Can Be a Part of It. San Francisco: No Strach Press.

Hintikka, Kari A. 2007. Web 2.0 - Johdatus internetin uusiin liiketoiminta mahdollisuuksiin. Tiede: Helsinki.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2007. Tutki ja kirjoita. 13 p., osin uud. laitos. Helsinki: Tammi.

Kangas, P., Toivonen, S. & Bäck, A. 2007. Google mainokset ja muita sosiaalisen median liiketoimintamalleja. Espoo: VTT.

Krug, S. 2006. Don't make me think - A common sense approaching to web usability. Berkley CA.

Yee, R. 2008. Pro Web 2.0 Mashups Remixing Data and Web Services. Berkeley, CA : Apress.

A Blogger's Manifesto: Free Speech and Censorship in the Age of the Internet.

Ringmar, Erik. 2007. Viitattu 12.4.2009.

<http://ia360927.us.archive.org/3/items/ABloggersManifestoFreeSpeechAndCensorshipInTheAgeOfTheInternet/ErikRingmarABloggersManifesto.pdf>

Beginning Ajax. 2007. Wrox Article. Viitattu 12.4.2009.

<http://www.wrox.com/WileyCDA/Section/id-303217.html>

Build an ASP.NET 2.0 Virtual Map Custom Control. Peter A. Bromberg. Viitattu

12.4.2009. <http://www.eggheadcafe.com/articles/20060708.asp>

Google Maps API Documentation. 2009. Viitattu 12.4.2009.

<http://code.google.com/intl/fi-FI/apis/maps/documentation/>

Google Mapki. 2007. Viitattu 12.4.2009.

http://mapki.com/index.php?title=Main_Page

Google Maps API Group. 2009. Viitattu 12.4.2009.

<http://groups.google.com/group/Google-Maps-API>

Google Maps API tutorial. 2008. Viitattu 12.4.2009.

<http://econym.googlepages.com/index.htm>

Google Maps announcement on Google Blog. 2005. Viitattu 16.6.2009.

<http://googleblog.blogspot.com/2005/02/mapping-your-way.html>

How cloud computing works. Strickland, J. 2009. Viitattu 12.4.2009.

<http://communication.howstuffworks.com/cloud-computing.htm>

Microsoft Virtual Earth SDKs. 2009. Viitattu 12.4.2009.

<http://msdn.microsoft.com/en-us/library/aa905677.aspx>

Mikä on Web 2.0 ja miksi yritysten kannattaa hyödyntää sitä verkkopalveluiden kehittämisessä. Samcom artikkeli. 2007. Viitattu 12.4.2009.

<http://www.samcom.fi/dms/White-Papers/Samcom-white-paper-Web-2-0/Samcom%20white%20paper%20Web%202.0.pdf>

ProgrammableWeb: How to make your own web mashup. 2009. Viitattu 20.7.2009.

<http://www.programmableweb.com/howto>

RSS tutorial. Denis Sureau. Viitattu 12.4.2009.

<http://www.xul.fr/en-xml-rss.html>

Virtual Earth Interactive SDK. 2008. Viitattu 12.4.2009.

<http://dev.live.com/Virtualearth/sdk/>

Web 2.0 Paul Graham. 2005. Viitattu 12.4.2009.

<http://www.paulgraham.com/web20.html>

Web 2.0, mashups and social networking - what is it all about?. 2007. Viitattu 20.7.2009.

<http://www.tamingthebeast.net/articles6/web2-mashups-social-network.htm>

Web 2.0 – So what is an enterprise mashup, anyway?. 2008. Viitattu 12.4.2009.

<http://www.networkworld.com/news/2008/042408-web-20-so-what-is.html>

Web 2.0 and Cloud Computing. 2008. Viitattu 20.7.2009.

<http://radar.oreilly.com/2008/10/web-20-and-cloud-computing.html>

What is Web 2.0. O'Reilly Network. 2005. Viitattu 12.4.2009.

<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

Youtube APIs and Tools. 2009. Viitattu 20.7.2009.

<http://code.google.com/intl/fi-FI/apis/youtube/overview.html>