

Lappeenrannan teknillinen yliopisto
Teknistaloudellinen tiedekunta
Tuotantotalouden koulutusohjelma

Diplomityö

Olli Makkonen

WWW-SISÄLLÖNHALLINTAJÄRJESTELMÄN KEHITTÄMINEN

Työn tarkastaja: Professori Marko Torkkeli

Työn ohjaaja: Professori Marko Torkkeli

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto

Teknistaloudellinen tiedekunta

Tuotantotalouden koulutusohjelma

Olli Makkonen

WWW-sisällönhallintajärjestelmän kehittäminen

Diplomityö

2010

45 sivua, 5 kuvaa, 1 esimerkki, 1 taulukko

Työn tarkastajat: Professori Marko Torkkeli

Hakusanat: sisällönhallinta, sisällönhallintajärjestelmät, CMS, verkkosovellukset, web-sisällönhallintajärjestelmät, WCM, WCMS.

Keywords: Content management, Content management systems, CMS, Web applications, Web content management systems, WCM, WCMS.

Tässä työssä keskitytään web-pohjaisiin sisällönhallintajärjestelmiin. Teoriaosassa käydään läpi aiheeseen liittyviä tekniikoita, web-ohjelmistotekniikan erityispiirteitä ja sisällönhallinnan käsitettä ja sen järjestelmiä. Teoriapuolella huomataan, kuinka laajaa osaa liiketoimintaprosesseista sisällönhallinta koskettaa. Siksi sen hahmottaminen on vaikeaa ja sitä voi lähestyä monesta eri suunnasta. Erilaisia järjestelmiä on lukuisia ja ne soveltuvat erilaisiin tilanteisiin. Käytännön työnä toteutetaan Codens Oy:n uuden sisällönhallinnan kehitystyö. Tavoitteena on saada huomattavia parannuksia vanhaan järjestelmään. Erityisesti kiinnitetään huomiota sivupohjasysteemiin ja käyttöliittymään, johon tuodaan mukaan web2.0:n tarjoamia uusia työkaluja. Järjestelmään saatiin useita parannuksia, mutta sen kehittämistä on vielä jatkettava, ennen kuin siitä saadaan kaupallisesti valmis tuote.

ABSTRACT

Lappeenranta University of Technology
Faculty of Technology Management
Department of Industrial Management

Olli Makkonen

Development of Web content management system

Master's Thesis

45 pages, 5 figures, 1 example, 1 table

Examiners: Professor Marko Torkkeli

Keywords: Content management, Content management systems, CMS, Web applications, Web content management systems, WCM, WCMS.

This thesis focuses on Web-based content management systems. The theoretical section includes related technologies, special features of Web engineering and content management concepts, and systems. Content management is integrated into most business processes of organizations. Because of that, it is hard to define and it can take many different approaches. There are many different content management systems available for organizations for various kinds of needs. The practical section examines the development of a new content management system for Codens Ltd. The development goal was to significantly improve the legacy system. Particular attention was given to the template system and the user interface. The creation of the CMS and UI rely heavily upon Web 2.0 tools. Development of both appears successful in this trial phase. Work will continue beyond this project's deadline to ensure the product is commercially successful.

ALKUSANAT

Kiitän Codensia.

SISÄLLYSLUETTELO

1	JOHDANTO.....	4
1.1	TAUSTA	4
1.2	TAVOITTEET JA RAJAUKSET	4
1.3	TYÖN RAKENNE	5
2	TERMIT JA TEKNIIKAT.....	7
2.1	WWW	7
2.2	HTML	7
2.3	SELAIMET	8
2.4	PALVELIN	8
2.5	CSS	9
2.6	PHP	9
2.7	JQUERY, AJAX JA JQUERY UI.....	10
2.8	MYSQL	10
2.9	FLASH	11
3	WEB-OHJELMISTOTEKNIikka.....	12
3.1	WEB-OHJELMISTOTEKNIIKAN EROT PERINTEISEEN OHJELMISTOKEHITYKSEEN	12
3.2	WEBKEHITYKSEN PROSESSI	13
3.3	WWW-KÄYTTÖLIITTYMÄSUUNNITTELU	14
4	SISÄLLÖNHALLINTA JA SEN JÄRJESTELMÄT.....	16
4.1	SISÄLTÖ	16
4.2	SISÄLLÖNHALLINTA.....	16
4.3	WEB-SISÄLLÖNHALLINTA	17
4.4	SISÄLLÖNHALLINTAJÄRJESTELMÄT	18
4.5	WWW-SISÄLLÖNHALLINTAJÄRJESTELMÄT	19
4.6	WWW-SISÄLLÖNHALLINTAJÄRJESTELMIEN OMINAISUUDET	19
4.6.1	<i>Tiedonkeruu</i>	<i>20</i>
4.6.2	<i>Työn kulku.....</i>	<i>20</i>
4.6.3	<i>Sisällön hallinnointi.....</i>	<i>21</i>
4.6.4	<i>Sisällön julkaisu.....</i>	<i>21</i>

4.6.5	<i>Muut ominaisuudet</i>	22
4.7	WWW-SISÄLLÖNHALLINTAJÄRJESTELMIEN VERTAILU	22
4.7.1	<i>Joomla</i>	23
4.7.2	<i>Drupal</i>	23
4.7.3	<i>Wordpress</i>	24
4.7.4	<i>Alfresco</i>	24
4.7.5	<i>Microsoft SharePoint</i>	25
4.7.6	<i>Yhteenveto</i>	25
4.8	JÄRJESTELMÄN VALINTA	26
5	WEB-ARKKITEHTI SISÄLLÖNHALLINTAJÄRJESTELMÄ.....	28
5.1	JÄRJESTELMÄN KUVAUS	28
5.2	KEHITYSPROSESSI.....	28
5.3	SISÄLLÖNHALLINTAJÄRJESTELMÄN OMINAISUUDET	29
5.3.1	<i>Yleistä</i>	29
5.3.2	<i>Käyttäjän todentaminen</i>	29
5.3.3	<i>Tietoturva</i>	30
5.3.4	<i>Käyttöliittymä</i>	30
5.3.5	<i>Sivujen editointi</i>	31
5.3.6	<i>Sivupohjan muokkaus</i>	31
5.3.7	<i>Sivurakenteen muokkaus</i>	33
5.3.8	<i>Käyttäjien hallinta</i>	33
5.3.9	<i>Tiedostojen hallinta ja editointi</i>	33
6	TULOKSET, POHDINTA JA TULEVAISUUS	35
7	YHTEENVETO.....	38
	LÄHTEET.....	39

SYMBOLI- JA LYHENNELUETTELO

AJAX	Asynchronous JavaScript And XML
CSS	Cascading Style Sheet
CMS	Content Management System
ECM	Enterprise Content Management
GPL	GNU General Public License
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
RSS	Really Simple Syndication
SQA	Software Quality Assurance
SQL	Structured Query Language
SSL	Secure Sockets Layer
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WCMS	Web Content Management System
WWW	World Wide Web
XML	Extensible Markup Language

1 JOHDANTO

1.1 Tausta

Tämän diplomityön tekijä työskentelee Codens Oy:ssä ja on yksi yrityksen perustajajäsenistä. Codens Oy on vuoden lopussa 2004 perustettu informaatioteknologian alalla toimiva yritys. Yritys tekee erilaisia verkkosovelluksia aina verkkosivuista vaativampiin tietokantasovelluksiin.

Yrityksessä huomattiin, että on iso potentiaalinen asiakaskunta joilla ei välttämättä ole varaa panostaa tuhansia euroja verkkosivujen saamiseksi. Näillä yrityksillä on kuitenkin tarve saada jostain siistit kotisivut, joita on helppo päivittää. Näitä usein ovat esimerkiksi aloittavat yritykset. Vuonna 2007 lanseerattiin verkkopalvelutuote Web-Arkkitehti, jonka tarjoaa tälle asiakaskunnalle laadukkaan kokonaisvaltaisen paketin edulliseen hintaan. Tämä sisältää verkkosivut, sähköpostit, webhotellin jne. Ideana on, ettei asiakkaan tarvitse tietää paljoa tietotekniikasta, vaan kaikki hoidetaan asiakkaalle valmiiksi *avaimet käteen*-periaatteella.

Vuonna 2010 yrityksessä on herätty tarpeeseen uudistaa Web-Arkkitehdin sisällönhallintajärjestelmä ja siihen liittyvä sivupohjasysteemi. Käyttäjäkokemusten perusteella on huomattu puutteita ja kehitetty useita parannusmahdollisuuksia. Toinen syy uuden version tekoon on se, että www-kehitykseen liittyvät työkalut ja kirjastot ovat ottaneet viime vuosina aimo harppauksen eteenpäin. Nyt käytössä on työkaluja, joita ensimmäistä versiota suunniteltaessa ei vielä ollut olemassakaan, tai ne olivat vielä liian alkeellisia. Näillä työkaluilla käyttöliittymään saadaan huomattavia parannuksia. Tarvittavat muutokset ovat niin perustavaa laatua olevia, että päädyttiin tekemään kokonaan uusi versio eikä parantelemaan vanhaa.

1.2 Tavoitteet ja rajaukset

Tämän diplomityön tavoite on saada yrityksen käyttöön uusittu www-sisällönhallintajärjestelmä, joka on valmis myyntiin. Tavoitteena on saada

helppokäyttöisempi ja ominaisuuksiltaan monipuolisempi järjestelmä vanhaan verrattuna. Sisällönhallintajärjestelmää tullaan kehittämään myös tämän diplomityön valmistumisen jälkeen, eikä ole tarkoituskaan, että kaikki suunnitteilla olevat osat olisivat valmiita. Uudessa järjestelmässä hyödynnetään Web 2.0:n mukaan tuomia uusia verkkokehityksen työkaluja kuten: jQuery, jQueryUI ja Ajax. Niitä hyödyntämällä saadaan tehtyä parempi käyttöliittymä ja vältetään turhia sivulatauksia.

Tärkein yksittäinen kehityskohde on sivupohjasysteemi. Sivupohjasysteemi on se osa järjestelmää, jonka avulla asiakkaat voivat tehdä räätälöidyn ulkoasun verkkosivuilleen. Graafikot ovat kokeneet vanhasta systeemistä löytyvät mahdollisuudet muokata erinäköisiä sivustoja rajallisiksi, ja tähän tarvitaan parannusta. Toinen tärkeä parannus liittyy sivujen editointiin. Nykyisessä järjestelmässä täytyy pitää kahta selainta rinnakkain auki, jos haluaa heti tallennuksen jälkeen nähdä miltä sivu näyttää itse verkkosivuilla. Käyttäjäkokenemusten perusteella myös kuvien hallinta kaipaa parannusta. Kuvaa pitää pystyä editoimaan vielä sen jälkeen, kun se on ladattu palvelimelle. Editointitilassa pitää olla työkalut, joilla kuvaa pystyy pienentämään ja rajaamaan. Avoimeen lähdekoodiin perustuvia ohjelmia käytetään ympäri järjestelmää aina kun se on mahdollista ja perusteltua.

Teoriapuolella keskitytään niihin osa-alueisiin, jotka ovat kehitettävän sisällönhallintajärjestelmän kannalta olennaisia. Ensin käydään läpi tekniikat, joita on tarvittu järjestelmän kehityksessä. Sisällönhallintaa käydään läpi yleisellä tasolla ja keskitytään juuri www-aspektin tuomaan näkökulmaan aiheeseen. Ohjelmistotekniikan puolella puhutaan lähinnä web-ohjelmistotuotannon erityispiirteistä perinteiseen ohjelmistotekniikkaan verrattuna.

1.3 Työn rakenne

Työ on jaettu teoriaosaan ja käytäntöpuoleen. Luvut kaksi, kolme ja neljä kuuluvat teoriaosaan ja luku viisi käsittelee käytännön työnä tehtyä sisällönhallintajärjestelmää. Luvussa kaksi käsitellään tekniikoita, joita järjestelmän taustalla tarvitaan ja hyödynnetään. Luku kolme sisältää www-ohjelmistotekniikan erityispiirteitä ja teoriaa. Luvussa neljä

pureudutaan sisällönhallinnan käsitteeseen, sisällönhallintajärjestelmiin ja siihen mitä tulee ottaa huomioon järjestelmää valittaessa.

Viidennessä luvussa käydään läpi uuden järjestelmän ominaisuuksia. Ominaisuuksissa kerrotaan, miten vanhassa järjestelmässä kyseinen kohta oli toteutettu ja miten uudessa se on parannettu. Johtopäätöksissä pohditaan sitä, kuinka hyvin uudella järjestelmällä pystyttiin vastaamaan niihin haasteisiin, joita järjestelmältä odotettiin. Samoin käydään läpi niitä tarpeita, joita järjestelmän jatkokehitys asettaa. Myös teoriapuolen huomioid vedetään yhteen.

2 TERMIT JA TEKNIIKAT

Tässä kappaleessa käydään läpi niitä tekniikoita, joita tarvitaan web-sovellusten kehittämiseen. Tekniikat on valittu sen perusteella, mitä käytännön työnä tehdyn sisällönhallintajärjestelmän tekemiseen on käytetty.

2.1 WWW

WWW (World Wide Web) eli Internet on globaali tietoverkko, joka koostuu miljoonista sivuista, jotka on linkitetty toisiinsa. Sivut sijaitsevat palvelimilla eli verkkoon liitetyillä tietokoneilla. Verkkoon syntyy päivittäin miljoonia sivuja ja samalla vanhoja poistuu. (Keränen, Lamberg & Penttinen 2006, 23).

Internet tarjoaa erilaisia palveluita, joista käytetyimmät ovat sähköposti, WWW ja FTP (File Transfer Protocol) eli tiedostojen siirtoon tarkoitettu protokolla. Verkossa olevat sivut löydetään URL (Uniform Resource Locator) osoitteen perusteella. Osoitteen perusteella löydetään oikea palvelin, joka palauttaa pyytäjälle sivun tiedot. (Keränen ym. 2006, 23-25).

2.2 HTML

HTML (Hypertext Markup Language) on verkkosivujen ulkoasun kuvaamiseen tarkoitettu standardisoitu julkaisukieli. Se koostuu rakennetta kuvaavista elementeistä ja niiden attribuuteista. (Keränen ym. 2006, 30). Kielessä käytetään < (pienempi kuin) ja > (suurempi kuin) merkkejä määrittämään se osa tekstistä mitä ei tulosteta ruudulle. Merkkien sisällä oleva teksti kertoo selaimelle, miten elementti määrittelee sen, kuinka teksti tulostaa ruudulle. (Musciano & Kennedy 1998, 18)

Esimerkki 1: Yksinkertainen HTML-dokumentti

```
<html>
<body>
<h1>Ensimmäinen otsikko</h1>
<p>Ensimmäinen kappale.</p>
</body>
</html>
```

HTML:n kuuluu olennaisena osana myös hyperlinkit, joiden avulla verkossa olevat dokumentit linkitetään toisiinsa. Hyperlinkki voi viitata millaiseen dokumenttiin tai tiedostoon tahansa, joka voi sijaita omalla tai jonkun toisen palvelimella. (Musciano & Kennedy 1998, 7-8)

HTML kielen kehitystä ohjaa W3C(World Wide Web Consortium). Organisaation sivuilta löytyvät suositukset myös muille internetissä käytettäville julkaisukielille. Organisaatio on levittäytynyt maailmanlaajuiseksi ja sen puheenjohtajana toimii WWW:n kehittäjä Tim Berners-Lee. (Keränen ym. 2006, 30)

2.3 Selaimet

Selain on käyttäjän omalla koneella ajettava ohjelma, jonka avulla voi katsella ja ladata Internetissä olevia dokumentteja ja tiedostoja. Selaimissa on yleensä graafinen käyttöliittymä, joka helpottaa verkossa liikkumista. (Musciano & Kennedy 1998, 6) Selainohjelmistoja on useita, joista käytetyimpiä ovat Mozilla Firefox ja Internet Explorer, joiden käyttäjät muodostavat lähes 80 prosenttia kokonaisuudesta. Muita käytössä olevia selaimia ovat Chrome, Opera ja Safari. (W3Schools 2010) Selainohjelmistoilla voi olla pieniä eroja siinä, miten ne tulkitsevat HTML-kieltä, joten sama dokumentti voi näyttää eri selaimella hieman erilaiselta. (Keränen ym. 2006, 31)

2.4 Palvelin

Palvelin on verkossa oleva tietokone, josta löytyy palvelinohjelmisto. Se vastaa selaimen

lähettämiin pyyntöihin, ja lähettää takaisin dataa, kuten esimerkiksi kuvia ja sivuja. Palvelimen voi periaatteessa pystyttää kuka tahansa, mutta yleensä niitä hallinnoivat oppilaitokset, yritykset ja Internet-operaattorit. (Keränen ym. 2006, 26)

Apache on yleisimmin käytössä oleva palvelinohjelmisto. Sen voi asentaa yleisimpiin käyttöjärjestelmiin. Se perustuu vapaaseen lähdekoodiin, eli sen asentaminen on maksutonta. Apacheen on mahdollista asentaa erilaisia lisämoduuleja, joilla palvelimen ominaisuuksia saa laajennettua tarpeensa mukaan. (Keränen ym. 2006, 26)

2.5 CSS

CSS (Cascading Style Sheets) on merkintäkieli, joka kertoo selaimelle dokumenttien ulkoasuun liittyviä tietoja. Se on tyyliohje, josta löytyy tietoja, kuten esimerkiksi minkä kokoista kirjasinlajia dokumentissa tulee käyttää. CSS on kaskadinen, joka tarkoittaa sitä, että useita CSS-dokumentteja voidaan yhdistää yhdeksi ohjeistoksi. (Korpela 2008, 3).

Yleensä CSS on erillinen dokumentti, ja näin samaa tyyliohjetta voidaan hyödyntää usean eri sivun ulkoasua muotoiltaessa. Käyttämällä CSS tyyliä voidaan dokumentin tieto ja sen esilletuonti pitää erillä toisistaan. Näin tekemällä voidaan samaa tyyliä soveltaa helposti koko sivustoilla ja tyylin muutos kaikkiin sivuihin onnistuu yhdestä paikasta. (Keränen ym. 2006, 33) Jos tyyli tiedostosta ei löydy jotain tyylimäärettä, käyttää selain oletusarvoa. (Korpela 2008, 3)

2.6 PHP

PHP (PHP: Hypertext Preprocessor) on avoimeen lähdekoodiin perustuva palvelimella suoritettava ohjelmointikieli. Sen syntaksi on C-kielen kaltainen, ja se sisältää kaikki ohjelmointikielille tyypilliset rakenteet. PHP-kieltä voidaan upottaa HTML dokumenttiin käyttäen `<?php` ja `?>` merkintää. PHP voidaan asentaa yleisimpiin palvelinohjelmistoihin. (Keränen ym. 2006, 34)

PHP mahdollistaa dynaamisten ja interaktiivisten sivujen luomisen. PHP:llä voidaan kirjoittaa HTML-koodia, kuvia tai muita tiedostoja ja sen avulla on myös helppoa

työskennellä tietokantojen kanssa. Se mahdollistaa vaativampien sovellusten, kuten verkkokauppojen tekemisen. (Sæther Bakken & Schmid 2003, 5)

2.7 jQuery, Ajax ja jQuery UI

jQuery on javascript kirjasto, johon on koottu ohjelmointia helpottavia skriptejä. (Bibeaut & Yehuda 2008, 2) Javascript on oliopohjainen ohjelmointikieli, jonka avulla voidaan tehdä dynaamista toiminnallisuutta verkkosivuille. Toisin kuin PHP, javascript suoritetaan vasta selaimessa eikä palvelimella. Sen syntaksi on C-kielen kaltainen. (Musciano & Kennedy 1998, 446)

jQuery on avoimeen lähdekoodiin perustuva javascript-kirjasto. jQuery helpottaa HTML-dokumenttien muokkausta lennossa, mahdollistaa animaatioita, efektejä ja tapahtumia. Sen avulla voidaan tehdä myös Ajax:ia (Asynchronous JavaScript And XML) hyödyntäviä käyttöliittymiä. Ajax on kokoelma eri tekniikoita, jonka suurin hyöty on siinä, että sivuille voidaan ladata palvelimelta sisältöä ilman, että koko sivua tarvitse päivittää uudelleen. Tämä mahdollistaa staattisen HTML-sivun muuntamisen monipuoliseksi käyttöliittymäksi. (Chaffer & Swedberg 2007, 6)

jQuery UI (User Interface) on jQuery kirjaston päälle kasattu laajennus, joka sisältää interaktiivisia työkaluja käyttöliittymien rakentamiseen. UI, kuten koko jQuery on selainriippumaton, joten sitä voi hyödyntää joutumatta miettimään ohjelman toimivuutta eri selaimilla. (Wellman 2009, 9) jQuery UI:stä löytyy erilaisia liitännäisiä mm. valikkoja ja muita hyödyllisiä vimpaimia. Koska se perustuu jQuery:n, on kaikki sen metodit automaattisesti ohjelmoijan käytössä. Sekä jQuery, että jQuery UI on lisensoitu GPL:llä (GNU Lesser General Public License), joten sitä voi vapaasti hyödyntää niin kaupallisissa, kun muissa ohjelmistoprojekteissa. (Wellman 2009, 21)

2.8 MySQL

MySQL (SQL = Structured Query Language) on vapaaseen lähdekoodiin perustuva tietokantaohjelmisto. Tietokanta on tietovarasto, jonne voidaan tallentaa suuriakin määriä

dataa. Tietokannasta on helppo tehdä hakuja ja samaa sisältöä voidaan käyttää halutessa monessa eri päätelaitteessa. MySQL voi käsitellä suorituskyvyn kärsimättä erittäin suuria määriä tietoa. (Keränen ym. 2006, 37)

2.9 Flash

Flash on internetsivuilla käytettävä tekniikka, joka on kehitetty animaatioita ja muuta multimediaa varten. Flash sisältää ActionScript ohjelmointikielen, jonka avulla on mahdollista tehdä pelejä ja kokonaisia verkkosivuja. Kaikki grafiikka siinä on vektorimuotoista, joten Flash-ohjelmaa voidaan skaalata grafiikan laadun kärsimättä. Selaimet eivät tue Flashiä automaattisesti, vaan sitä sisältävien elementtien näyttämiseen tarvitaan ilmaiseksi ladattava lisäosa. (Keränen ym. 2006, 170)

3 WEB-OHJELMISTOTEKNIikka

Web-ohjelmistotekniikka on verraten nuori ohjelmistotekniikan tieteenhaara, joka on erikoistunut tutkimaan web-kehityksessä tarvittavia tekniikoita. Se tutkii työkaluja web-sovellusten kehittämiseksi ja menetelmiä, jotka tukevat niiden suunnittelua, evoluutiota ja arviointia. (Woojong 2005, 9)

3.1 Web-ohjelmistotekniikan erot perinteiseen ohjelmistokehitykseen

Roger Pressmanin (2001, 771) mukaan webkehitykselle ominaisia piirteitä ovat:

1. Verkon hyödyntäminen. Websovellukset sijaitsevat verkossa ja niiden pitää pystyä palvelemaan laajaa joukkoa erilaisia asiakasohjelmia. Ne voivat sijaita Internetissä, intranetissä tai extranetissä.
2. Sisältö on pääosassa. Usein websovelluksen tehtävä on näyttää loppukäyttäjälle informaatiota.
3. Jatkuva kehitys. Perinteisillä ohjelmistoilla päivitykset on suunniteltu ennalta ja niitä julkaistaan aikataulun mukaan. Webohjelman kehitys ja päivitys on jatkuvaa. Varsinkin sisältöä voidaan joutua päivittämään useita kertoja päivässä.

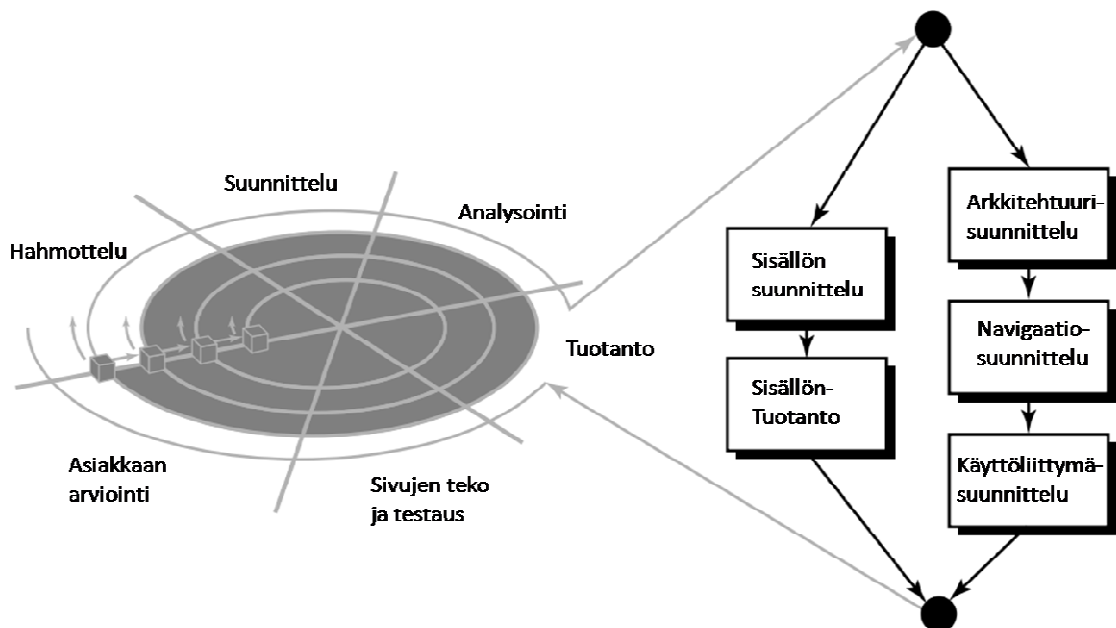
Mendes ja Mosley (2005, 4) taas painottavat sitä, kuinka paljon nopeammassa tahdissa verkkosovelluksiin tuodaan ja hyödynnetään uutta tekniikkaa perinteiseen sovelluskehitykseen verrattuna. Woojong (2005, 9) vertaa web-pohjaista ohjelmistoa puutarhaan. Se kehittyy ja kasvaa, eikä ikinä pysähdy paikalleen. Ohjelman rakenne pitää olla suunniteltu tämä ominaisuus silmälläpitäen. Siinä täytyy olla rakenne, joka tukee kehitystä, mutta antaa myös tilaa ja joustavuutta suuremmillekin muutoksille.

Sovellusten käyttäjäkunta on myös usein vaikeammin määriteltävissä. Sovellukset hyödyntävät hyperlinkitystä ja voivat viitata muihin sivuihin ja tiedostoihin. Tämä asettaa navigoinnin ja käyttöliittymän suunnittelulle erityisvaatimuksia. Webkehittäjien joukko on myös paljon kirjavampi, sillä niiden tekeminen voi onnistua myös amatööreiltä siinä missä myös alan ammattilaisilta. (Mendes & Mosley 2005, 4)

Kuitenkin web-ohjelmien kehittämisessä tulee ottaa huomioon kaikki samat asiat kuin perinteisessä ohjelmistokehityksessä. Web-ohjelmiston kehitys alkaa ongelman määrittämisestä ja siitä kuinka ohjelma tulee tämän ongelman ratkaisemaan. Suunnitteluun tulee myös web-ohjelmoinnissa varata riittävästi aikaa. Suunnittelua seuraa toteutus, testaus ja julkaisu. Prosessin onnistuminen voidaan varmistaa käyttämällä laatutekniikkaa SQA (Software Quality Assurance), jota käytetään myös perinteisten ohjelmistoprojektien laadun valvonnassa. (Pressman 2001, 769-770)

3.2 Webkehityksen prosessi

Pressman (2001, 774-776) esittää web-kehitykseen seuraavaa mallia (ks. kuva1 alla). Se perustuu Boehmin (1986) esittämään spiraalimalliin, jossa samanaikaisesti kehitetään uutta prototyyppiä kun sitä suunnitellaan. Malli on luonteeltaan iteroiva, eli sovellus täsmentyy jokaisella kierroksella. Tämän vuoksi se soveltuu erityisen hyvin web-kehitykseen, sillä web-sovelluksen tuskin ikinä voi sanoa olevan lopullisesti valmis. Boehmin malliin on tuotannon kohdalle lisätty web-kehitykseen vaadittavia työvaiheita.



Kuva 1. Webkehityksen prosessi (Pressman 2001, 775)

Prosessi alkaa hahmottelusta, jossa luodaan tavoitteet webohjelmistolle. Suunnittelussa arvioidaan projektin kesto, kustannukset ja arvioidaan projektiin liittyviä riskejä. Analyysissä asetetaan järjestelmän tekniset vaatimukset, vaatimukset graafiselta käyttöliittymältä, sekä suunnitellaan mitä sisältöä joudutaan tuottamaan. (Pressman 2001, 776)

Tuotanto on jaettu kahteen rinnakkaiseen toimintaan, jossa sisältöä tehdään samalla kun järjestelmän teknistä runkoa kasataan. Sivujen teko ja testausvaiheessa tuotettu sisältö yhdistetään teknisessä kehityksessä syntyneeseen ohjelmaan ja samalla testataan navigoinnin ja muun toimintaa. Ohjelman toimintaa testataan myös erilaisissa ympäristöissä eli mm. eri selaimilla. (Pressman 2001, 776)

3.3 WWW-käyttöliittymäsuunnittelu

WWW-käyttöliittymäsuunnittelu noudattaa kaikkia tavallisen käyttöliittymäsuunnittelun periaatteita. (Pressman 2001, 785) Hyvän käyttöliittymäsuunnittelun kahdeksan sääntöä Shneidermanin ja Plaisantin (2009) mukaan:

1. Pidä komennot johdonmukaisina. Samoista paikoista pitää tapahtua samoja toimintoja ja samoja värikoodeja käytetään samojen toimintojen kuvaamiseen.
2. Tarjoa vakituisille käyttäjille oikoteitä. Edistyneille käyttäjille voi tarjota esim. pikanäppäimiä.
3. Tarjoa informatiivista palautetta.
4. Suunnittele dialogit niin, että niillä on järkevä lopetus.
5. Tarjoa yksinkertainen keino käsitellä virhetilanteet.
6. Anna mahdollisuus perua toiminto helposti.
7. Anna käyttäjälle tunne, että hän on kontrollissa.
8. Älä anna käyttäjän eteen liikaa informaatiota kerralla. Käyttäjän väliaikaiseen muistiin ei mahdu kuin tietty määrä asioita kerralla.

Web-käyttöliittymien käytössä olevat työkalut ovat kehittyneet vasta viime vuosina samalle tasolle kun perinteisten ohjelmistojen. Enää ei jokainen käyttäjän klikkaus aiheuta

uutta sivulatausta, vaan tarvittava elementti päivitetään Ajax:lla. Näin käyttäjä ei koe viivettä, vaan voi toimia vapaasti samalla kun ohjelma taustalla hakee tarvittavia tietoja palvelimelta. (Garnett, 2005)

Pressman (2001, 785) painottaa käyttöliittymän ulkoasun tärkeyttä. Jos ulkoasu ei miellytä käyttäjää heti ensi silmäyksellä, on mahdollista että käyttäjä poistuu. Hän mainitsee myös seuraavia erityispiirteitä verkkosovelluksissa: Käyttäjät eivät halua käyttää vierityspalkkeja ja siksi informaation tulisi näkyä yhdellä normaalin resoluution omaavalla näyttöruudulla. Käyttöliittymän tulisi tarjota navigoinnin kaikkialle, niin eteen kuin taakse. Ei siis riitä, että selaimesta löytyy *palaa takaisin* nappula. Navigointimahdollisuuksien tulisi olla ilmiselviä myös harvoin ohjelmaa käyttäville. Käyttäjän ei pitäisi joutua etsimään ruudulta sitä, mistä pääsee haluamaansa toimintoon.

4 SISÄLLÖNHALLINTA JA SEN JÄRJESTELMÄT

4.1 Sisältö

Sisällöksi voidaan luokitella kaikki yrityksen tuottama tieto. Se voi olla dokumentti, kirjanpitoon liittyvä tapahtuma, äänite, kuva, video, myyntimateriaali, ohjeistus, muistio tai mikä tahansa muu vastaava. (Hasanali & Leavitt 2003, 2) Tieto on ryhmiteltävä tietoyksiköiksi, jotta sitä on mahdollista hallita. Tietoyksiköllä tarkoitetaan tallennettua kokonaisuutta, joka usein sisältää metatietoa. Metatieto on sisältöä kuvaavaa tietoa, joka helpottaa sisällön löytymistä. Metatieto yleensä sisältää ainakin sisällön tuottajan nimen, päivämäärän ja sisältöä kuvaavia avainsanoja. (Salminen 2007)

4.2 Sisällönhallinta

Sisällönhallinta on prosessi, jossa kerätään, hallinnoidaan ja julkaistaan informaatiota mihin tahansa julkaisukanavaan. Yleisimmin sisällönhallinnasta puhutaan liittyen suuriin verkkosivustoihin. Kun verkkosivut kehittyivät pienistä informatiivisista sivuista suuriin alati muuttuviin laajoihin sivustoihin, tuli sisällönhallinnasta tärkeämpää yritysten viestinnässä. Verkkajulkaisu on kuitenkin vain yksi monista informaatiokanavista, joita yrityksen tarvitsee organisoida. (Boiko 2002)

Tiedonkeruu on vaihe jossa organisaatio joko tuottaa sisältöä itse, tai kerää sitä jostain lähteestä. Tietoa editoidaan, pilkotaan ja siihen lisätään metatietoa. Tämän jälkeen tieto syötetään hallinnointikeskukseen, joka yleensä on sähköinen tietokanta. Julkaisuvaiheessa tieto haetaan hallinnointikeskuksesta ja tulostetaan eri kanaviin kuten verkkosivuille, esitteisiin, sähköposteihin ja uutiskirjeisiin. (Boiko 2001)

Sisällönhallinnassa tärkeää on myös työnkuva(eng. workflow), joka tarkoittaa yrityksen sisällään sopimaa kuvausta siiten miten tietty prosessi suoritetaan. Tämä on tärkeää yrityksille, joissa samat työvaiheet toistuvat usein ja aikaa näiden suorittamiseen on rajatusti. Kiinnittämällä prosessiin huomiota yritys voi optimoida työnkulun automatisoimalla rutiineja ja minimoida ajan mikä kuluu prosessin eri vaiheiden

välillä. Työnkulku on tärkeää varsinkin silloin, kun prosessi kulkee organisaation läpi eri osastojen kesken. (Michelinakis 2004, 13-14)

4.3 Web-sisällönhallinta

Web-sisällönhallinta tarkoittaa niitä periaatteita ja käytäntöjä, joita organisaatio harjoittaa verkkosisältönsä keräämisessä, kehityksessä, ylläpidossa ja julkaisussa. Se pohjautuu dokumenttien-, asiakkuuksien ja ohjelmistojen versiohallintaan. (Vidgren ym. 2001, 477) Kuten ohjelmistojen versiohallinnassa, on verkkodokumenteissa tärkeää pitää kirjaa muutoksista, kirjata välitavoitteet ja uusimmat työn alla olevat versiot dokumenteista. (Michelinakis, 2004, 11) Organisaatioissa web-sisällönhallintaa on jouduttu kehittämään sen jälkeen kun on huomattu, että verkkosivujen sisältöä joudutaan lisäämään, editoimaan ja vanhaa tietoa poistamaan. Jos prosessia ei hallita järkevästi, voi organisaatio kohdata seuraavia ongelmia Goodwin & Vidgrenin (2002) mukaan:

- Pullonkaulat: Sisältöä voi tulla monesta eri lähteestä, ja jos se on yhden henkilön harteilla johtaa se helposti viivästyksiin sisällön julkaisussa.
- Tiedon päällekkäisyys: Samaa tietoa voi olla useassa paikassa ja se päivitetään vain yhteen.
- Navigointi: Jos sisältö ja sen rakenne eivät ole kontrollissa, ei sivuston käyttäjä löydä haluamaansa informaatiota.
- Tiedon lukuoikeudet: Verkkosivuille voi joutua keskeneräistä, tai jopa sinne kuulumatonta informaatiota.
- Yhteneväisyys: Jos sisältöä tulee useasta eri lähteestä, hallinnoimattomana sen ulkoasu ei pysy samankaltaisena.
- Sisällön seuranta: Ilman hallintaa ei kukaan tiedä mikä versio dokumentista on tai kuka sen on tehnyt.
- Liiketoimintaprosessit: Sisältö on usein sidoksissa liiketoimintaprosessiin ja ongelma sisällönhallinnassa heijastuu suoraan koko prosessiin.

Web-sisällönhallinta on tärkeää organisaatioille, jotka tuottavat paljon sivuja usean eri sisällöntuottajan avulla. Web-sisällönhallinta onkin hankalaa määrittää, sillä se on

sidoksissa moneen liiketoimintaprosessiin ja rajanpintaan. (Goodwin & Vidgren, 2002) Tärkeitä asioita siinä on verkkodokumenttien elinkaaren hallinta, dokumenttien varastointi, tiedon sekä metatiedon muokkaus ja mahdollisuus muokata tiedon esilletuontia helposti. Tässä sivupohjat ja niiden muokattavuus ovat avainasemassa. (Vidgren ym. 472-477)

4.4 Sisällönhallintajärjestelmät

Sisällönhallintajärjestelmä (CMS, Content Management System) on järjestelmä, joka on vastuussa edellisessä luvussa kuvatusta sisällönhallinnan prosessista tai ainakin sen osaluueesta. Yrityksillä on käytössä laaja kirjo erilaisia apuvälineitä tiedon hallintaan. Yksinkertaisimmillaan se voi tarkoittaa paperisia muistilappuja tai toisesta ääripäästä monimutkaisia tietokantasovelluksia. (Boiko 2001) Useimmiten sillä kuitenkin tarkoitetaan verkkojulkaisuun tarkoitettujen dokumenttien hallintajärjestelmää. (Boiko 2002) Michelinakis (2004, 29) jakaa sisällönhallintajärjestelmät luokkiin seuraavasti: Dokumenttienhallintajärjestelmät, tietämyksenhallintajärjestelmät, järjestelmäintegraatio-systemit, sähköisen kaupankäynnin järjestelmät ja web-sisällönhallintajärjestelmät.

Hasanali ja Leavitt (2003, 4) luettelevat hyvän sisällönhallintajärjestelmän ominaisuuksia:

- Järjestelmä helpottaa käyttäjää löytämään tarvittava sisällön helpommin ja tuottavammin.
- Pystyy hallitsemaan useammassa eri formaatissa olevaa tietoa.
- Se sisältää selkeän prosessin tiedon viemiseen järjestelmään.
- Sen avulla tieto ja sen esilletuonti on erotettu, ja samaa sisältöä voidaan käyttää uudelleen.
- Mahdollistaa metatiedon lisäämisen.
- Turvaa sisällön jakamisen ja tuottamisen vain niille henkilöille, joilla on siihen oikeus.
- Mahdollistaa julkaisualustojen hakea automaattisesti merkityksellistä tietoa.
- Seuraa kuinka informaatiota käytetään.

Sisällönhallintajärjestelmän päävastuu on tiedon hallinnassa ja sen säilytyksessä, mutta se

voi osallistua myös informaation keräämiseen ja sen julkaisuun. Sisällönhallintaprosessin eri vaiheet eivät ole järjestelmässä erillä toisistaan vaan useimmiten niiden välit hämärtyvät. Usein tietoa syötetään järjestelmään jo ennen kuin se on täysin valmista, jolloin hallintapuolen voi katsoa auttavan tiedonkeruuvaihetta. Julkaisupuoli voi taas sisältää lomakkeita, joita käytetään sisällön tuottamiseen. Näin julkaisu voi osallistua tiedon keruuseen. (Boiko 2002)

4.5 WWW-sisällönhallintajärjestelmät

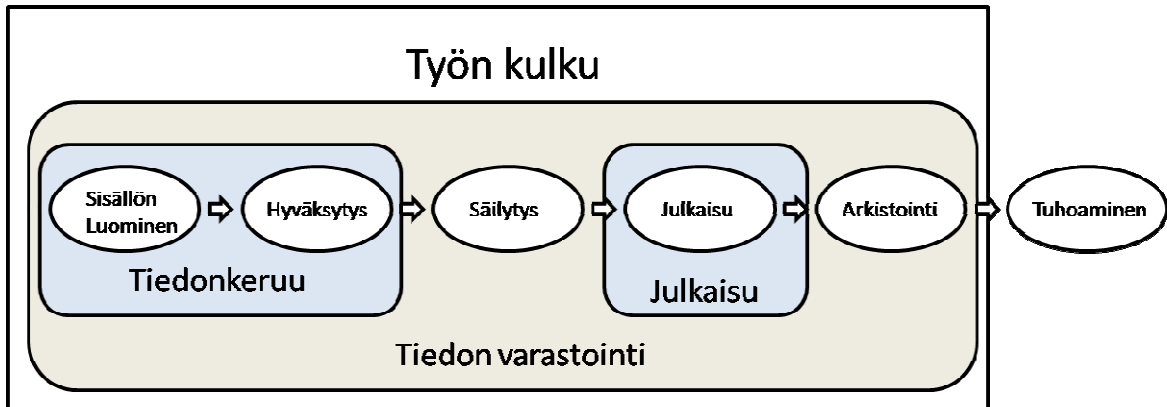
Kun WWW oli kehityksensä alussa, koostui Internet lukuisista tekstieditorilla kirjoitetuista HTML-dokumenteista. Ne jouduttiin siirtämään palvelimelle käyttäen FTP:tä. Sisällöntuottajat joutuivat turvautumaan teknisen osaajan apuun saadakseen tietoa Internetiin. Ongelmaksi muodostuivat rikkonaiset dokumentit ja epäyhtenevä linja sivustoilla, joihin liittyi paljon dokumentteja. Näiden ongelmien, lisääntyvän sisällön ja vierailijoiden määrän aikaansaamana syntyi tarve järjestelmiin, jotka auttavat verkkosivujen julkaisussa. (McKeever, 686)

WCMS (Web Content Management System) eli WWW-sisällönhallintajärjestelmä on ohjelmisto, joka on erikoistunut verkkopalvelujen ylläpitoon. WCMS termiä yleisemmin näitä järjestelmiä kutsutaan vain nimellä CMS tai julkaisujärjestelmä. WCMS auttaa yritystä automatisoimaan sisällön tuotantoa, hallintaa ja sen jakelua. Sisältö näytetään ensisijaisesti verkkosivuilla, mutta useat järjestelmät jakavat sisältöä myös muille päätelaitteille kuten puhelimille. (Rockley, Kostur & Manning 2003, 323)

4.6 WWW-sisällönhallintajärjestelmien ominaisuudet

Browning ja Lowndes (2001) jakavat sisällönhallintajärjestelmän ominaisuudet neljään alaluokkaan: tiedonkeruu, työnkulku, hallinnointi ja julkaisu. Prosessia on havainnollistettu kuvassa 2. He kirjoittavat myös, että ohjelmaa voidaan kutsua sisällönhallintajärjestelmäksi vasta, jos siitä löytyy versiointi, työnkulkua helpottavat toiminnot ja mahdollisuus integrointiin muihin järjestelmiin. Tässä luvussa käydään läpi tärkeimmät ominaisuudet tätä jaottelua hyödyntäen. Luvuissa esiintyvät ominaisuudet esitellään

suurpiirteisessä tärkeysjärjestyksessä, eikä kaikissa järjestelmissä ei välttämättä ole jokaista esitettyä ominaisuutta.



Kuva 2, Web-sisällönhallinnan työn kulku. (Browning & Lowdes 2001 mukaan)

4.6.1 Tiedonkeruu

WCMS sisältää yleensä sisällöntuottamista helpottavan editorin, joka muuntaa tekstin automaattisesti HTML muotoon. Yleensä editoria pystyy käyttämään suoraan selaimesta verkon yli. Yksinkertaisimmissa sovelluksissa editori on vain tavallinen HTML-lomake. Tämä alentaa sisällöntuottajalta vaadittavaa teknistä osaamista. (Rockley, Kostur & Manning 2003, 325) Kuitenkin järjestelmän tulisi tarjota edistyneemmille käyttäjille mahdollisuus muokata suoraan HTML-koodia. (Michelinakis 2004, 36) Sisältöä voidaan tuoda myös tietojenkäsittely ohjelmista kuten Microsoft Wordista tai muista tekstinkäsittelyohjelmista. Järjestelmän pitää pystyä vastaanottamaan tietoa useista eri formaateista. Sisällön ohien pitää pystyä liittämään dokumenttia kuvaavaa metatietoa. Browning ja Lowdes (2001, 8) esittävät, että metatiedon kirjoittaminen tulisi olla yksi uuden dokumentin tallennuksen vaatimuksista.

4.6.2 Työn kulku

Työnkululla tarkoitetaan sitä vaihetta, mikä käydään läpi sisällöntuotannosta julkaisuun ja lopulta sen arkistointiin. Tyypillisesti tämä sisältää hyperlinkityksen tarkistuksen ja sisällön läpikäynnin johtajan tai lakiosaston toimesta. Perinteisessä web-sisällönhallinnassa työn kulun hallinta on ollut erillinen toiminto, joka ei ole edustettuna ohjelmistossa. (Browning & Lowdes 2001, 5) Monissa uudemmissa järjestelmissä tämä on huomioitu ja niistä löytyy ainakin mahdollisuus sisällön hyväksyttämiseen ennen sen julkaisua.

Järjestelmä auttaa prosessia siten, että hyväksyvät tahot saavat automaattisesti tiedon uudesta sisällöstä joka heidän tulee käydä katselmoimassa. Tämä voi tapahtua siten, että sisään kirjautuessa käyttäjä näkee hänelle osoitetut tehtävät. Käyttäjät voivat saada odottavista tehtävistä myös sähköposteja. (Michelinakis 2004, 36) Työnkulkuun kuuluu myös arkistointimahdollisuus, joka voi toimia manuaalisesti tai automaattisesti tietyn ajan kuluttua. (Vidgren ym. 2001, 473) Työnkulun prosessia on pystyttävä muokkaamaan organisaatiomuutosten niin vaatiessa. (Robertson 2001)

4.6.3 Sisällön hallinnointi

Käyttäjien järjestelmään tuoma sisältö tallennetaan tietokantaan tai muuhun vastaavaan tietovarastoon. Sisältöä pystyy muokkaamaan samalla editorilla, jolla sisältöä myös luodaan. Järjestelmä päivittää automaattisesti sivustolla olevan navigoinnin ja sivukartat aina, kun uusi sisältödokumentti lisätään sivurakenteeseen. Editoidusta dokumentista tehdään uusi päivitetty versio ja usein käyttäjillä on mahdollisuus nähdä siihen liittyvä versiohistoria. Historiasta tulisi olla mahdollisuus niin halutessa palata aikaisempaan versioon. Historiasta löytyy yleensä myös dokumenttien kirjoittajien ja eri versioiden muokkaajien tunnukset. Sisältövarasto voidaan integroida organisaation muihin järjestelmiin. (Robertson 2001) Järjestelmästä tulisi löytyä toiminto, joka estää kahden henkilön muokkaamasta samaa dokumenttia samaan aikaan. (Vidgren ym. 2001, 493)

4.6.4 Sisällön julkaisu

Web-sisällönhallintajärjestelmät sisältävät sivupohjasysteemin, jonka avulla tietokantaan tallennettu sisältö tuodaan loppukäyttäjän luettavaksi. Sivupohjan avulla tieto ja sen esilletuonti on erotettu toisistaan. Sisällön esitys pysyy yhtenevänä koko sivustolla. Sisältö voidaan julkaista keskitetysti, eli jokainen verkkoon pääsevä dokumentti katselmoidaan ja hyväksytetään ennen sen julkaisua. (Robertson 2001)

Sisällöstä voidaan tehdä myös RSS-syöte (Really Simple Syndication). Syötteen avulla sisältöä voi julkaista muilla sivuilla ja alustoilla. (Petrovic ym. 2005) Tämä ominaisuus on käytössä erityisesti verkkopäiväkirjojen eli blogien yhteydessä. Jotkut järjestelmät tekevät myös oman version sivuista mobiililaitteille. Järjestelmän tulee myös pystyä julkaisemaan

sisältöä PDF (Portable Document Format)-formaatissa ja mobiililaitteille. (Robertson 2001)

4.6.5 Muut ominaisuudet

Järjestelmän tulisi olla myös rakennettu tavalla, että siihen on helppo kehittää uusia liitännäisiä. Järjestelmään tulisi olla mahdollista liittää uusia ominaisuuksia ilman, että se vaatii suuria ponnisteluja järjestelmän kehittäjiltä. Esimerkkinä voidaan mainita verkkokauppasovelluksen liittäminen yrityksen kotisivuille. (Michelinakis 2004, 32) Järjestelmän käyttöliittymän tulee olla mieluiten selaimessa toimiva. (Browning & Lowndes 2001, 8)

Tietoturvallisuus on myös tärkeä osa järjestelmää. Vaidyanathan (2009) jakaa turvallisuuden seuraavasti:

- Luottamuksellisuus. Tietoa ei saa päästä sellaisen käsiin kenelle se ei ole tarkoitettu.
- Eheys. Järjestelmään ei saa tehdä muutoksia muut kuin siihen luvan saaneet.
- Todennus. Tiedon alkuperä pitää tarkistaa.
- Saatavuus. Tietoturvajärjestelmien on oltava saatavilla niitä tarvittaessa.
- Kiistattomuus. Varmistetaan että tiedon lähettäjä ja vastaanottaja ovat varmasti osallistuneet prosessiin.

Jos järjestelmää ei kehitetä näitä turvallisuusnäkökohtia silmälläpitäen, tulee se lopulta kalliimmaksi jatkuvien tietoturvapäivitysten muodossa aina uuden riskin löytyessä.

4.7 WWW-sisällönhallintajärjestelmien vertailu

Vertailuun on valittu tunnettuja ja yleisiä (BuiltWith.com) sisällönhallintajärjestelmiä. Mukaan on otettu hieman eri tarkoituksiin tehtyjä ja eri kokoluokan sisällönhallintajärjestelmiä. Koska kaikista valituista järjestelmistä löytyy sisällönhallintajärjestelmien normaalit tekniset ominaisuudet (visuaalinen editori, tietokanta, jne.), päätettiin tarkastelunäkökulmaksi valita Browning ja Lowdesin (2001) painottamat versiohallinta-, työnkulku- ja integraatio-ominaisuudet. Nämä ominaisuudet

ovat kuitenkin niitä, joita suuret organisaatiot järjestelmiltä vaativat.

4.7.1 Joomla

Joomla on yksi suosituimmista julkaisujärjestelmistä. Joomla on tehty käyttäen PHP:tä ja MySQL:ää. Joomlaa kehittää iso joukko ihmisiä, ja siihen on saatavilla lukematon määrä lisäosia, joilla se voi räätälöidä omaan tarkoitukseen sopivaksi. Sen käyttöliittymä on käyttäjäystävällinen, mutta vaatii silti paneutumista eikä välttämättä luonnistu hetkessä peruskäyttäjältä. Joomla on suosittu graafisemman lähestymisen omaavien verkkokehittäjien keskuudessa. Joomla'n suurimpina puutteina pidetään sen skaalautuvuutta, siitä ei esim. löydy mahdollisuutta monen sivuston ylläpitoon yhdellä asennuksella. Joomla on vapaata lähdekoodia ja julkaistu GPL(GNU General Public License) –lisenssillä. (joomla.org)

Versiohallinta on yksi niistä ominaisuuksista, joka on Joomla'ssa huonosti edustettuna. Joomla'n 1.0 versiosta löytyy yksinkertainen versioiden hallinta, mutta yleisistä versiokontrollityökaluista versioiden yhdistäminen ja vertailu puuttuvat kokonaan. Joomlaan on kuitenkin saatavilla lisäosia, joilla voi paikata sen puutteita. Hyviä lisäosia ovat ainakin Eexact Revision Manager (ERM) ja Version Control System (VCS). Työnkuvaa auttavat toiminnot ovat samalla tavalla myös lisäosien varassa. Workflow Extension for Joomla!-lisäosa on maksullinen, mutta tarjoaa monipuoliset ominaisuudet. Integraatiot muihin järjestelmiin onnistuvat Joomla'sta löytyvien rajapintojen ansiosta. (joomla.org)

4.7.2 Drupal

Drupal on Joomla'n ohella suosituin vapaan lähdekoodin julkaisujärjestelmä. Sitä pidetään hyvin skaalautuvana, ja se soveltuukin paremmin isojen järjestelmien julkaisupohjaksi. Sen käyttöliittymää pidetään hankalampana kuin Joomla'n, joten se ei aivan aloittelijoille sovi. Drupal on suosittu teknisesti taitavien webkehittäjien keskuudessa. (drupal.org)

Samaan tapaan kuin Joomla'ssa, Drupaliin on saatavilla lisäosia joilla sen työnkuvan hallintaa saadaan parannettua. Byron ym. (2008, 211-233) näyttää kirjassaan, kuinka Drupalin työnkulkua saa muokattua toimimaan isomman organisaation tarpeisiin. Se vaatii

useamman moduulin yhteispeliä, eikä sen pystyttäminen käy käden käänteessä.

4.7.3 Wordpress

Wordpress on verkkopäiväkirjojen eli blogien tekemiseen tarkoitettu vapaan lähdekoodiin perustuva järjestelmä. Se on tehty PHP:llä ja se käyttää MySQL tietokantaa. Sen käyttöliittymä on helppo oppia, ja sopii erittäin hyvin myös aloittelijoille. Vaikka se on tarkoitettu blogien pitämiseen, sillä on mahdollista tehdä tavallisia verkkosivuja. Wordpress sopii hyvin pienempien verkkosivujen alustaksi, mutta monimutkaisten verkkopalvelujen kohdalla se ei ole parhaimmillaan. (wordpress.org)

Wordpressistä löytyy käyttäjille erilaisia rooleja ja työnkulkuun on saatavilla yksinkertaisia toimintoja. Käyttäjälle on mahdollista antaa rooli jossa kaikki käyttäjän tekemä sisältö käy läpi hyväksytysprosessin. Lisää ominaisuuksia on mahdollista saada lisäosien kautta. (Brazel 2010, 343-352) Wordpressissä on yksinkertainen versiohallinta, jossa vanhaan versioon voi halutessaan palata (wordpress.org). Wordpress on mahdollista integroida yrityksen muihin järjestelmiin, mutta ilman työtä se ei sovellu suoraan suuren organisaation tarpeisiin. (Brazel 2010, 379)

4.7.4 Alfresco

Alfresco on javalla kirjoitettu suurille yrityksille tarkoitettu sisällönhallintajärjestelmä. Sen lähestymistapa sisällönhallintaan on isoille organisaatioille tyypillisesti dokumenttienhallinnassa. Alfrescosta on kaksi eri versiota, vapaaseen lähdekoodin perustuva lisenssi ja kaupallinen lisenssi. Alfresco yrittääkin yhdistää vapaan lähdekoodin tuoman suuren määrän kehittäjiä ja lisäosia isojen yritysten vaatimaan tekniseen tukeen ja koulutuspalveluihin. (alfresco.com)

Työnkulun ominaisuudet, versionhallinta ja integraatio muihin järjestelmiin ovat Alfescossa huomattavasti edellä esitettyjä järjestelmiä parempaa. Varsinkin tiedonkeruun puolella se on valmiimpi kokonaisuus. Siitä löytyy mm. toiminto jossa järjestelmä automaattisesti kerää avainsanoja dokumentin metatietoihin, kun se tuodaan järjestelmään. Järjestelmän tietovarastoon voi suoraan tuoda dokumentteja monista eri tekstinkäsittelyohjelmista. (alfresco.com)

4.7.5 Microsoft SharePoint

Microsoft SharePoint on kaupallinen järjestelmä, joka edellä esitetyistä kilpailee lähimpänä Alfrescon kanssa samoista asiakkaista. SharePoint on verkkopohjainen yhteistyöprojekteihin tarkoitettu sisällönhallintajärjestelmä. Se perustuu arkkitehtuuriin, jossa kaiken keskellä on dokumenttienhallintapalvelin. Myös tästä järjestelmästä löytyy rajapinta sille, että muutkin kuin Microsoft voivat rakentaa uusia liitännäisiä.

SharePointista löytyy erittäin paljon ominaisuuksia, ja se soveltuukin monelle erityyppiselle tarpeelle ja organisaatiolle. (sharepoint.microsoft.com)

4.7.6 Yhteenveto

Tähän vertailuun valitut vapaan lähdekoodin järjestelmät lähestyvät sisällönhallintaa enemmän verkkojulkaisun puolelta, kun taas tähän valitut kaupallisemmat sovellukset tulevat dokumenttien hallinnasta. Kaupallisissa järjestelmissä on paremmat valmiudet käsitellä suuren joukon tuottamaa informaatiota ja yhteistyöprojekteja. Avoimen lähdekoodin järjestelmät kuitenkin kehittyvät koko ajan ja lähenevät kaupallisia järjestelmiä myös näissä ominaisuuksissa.

Taulukko 1. Sisällönhallintajärjestelmien vertailu

Järjestelmä	Lisenssi	Hyvä	Huonoa	Parhaimmillaan
Joomla	GPL	+Näyttää hyvältä +Helppo käyttää +Paljon laajennuksia	-Työnkuva -Skaalautuvuus	PK-yrityksen verkkosivujen tai verkkokaupan sisällönhallintajärjestelmänä
Drupal	GPL	+Modulaarinen +Skaalautuvuus	-Työnkuva -Vaikea käyttöliittymä -Vaatii räätälöintiä	Suosittun verkkosivuston räätälöitynä hallintajärjestelmänä
Wordpress	GPL	+Yksinkertainen +Helppo oppia +Paljon laajennuksia	-Työnkuva -Käyttäjäroolit -Integrointi	Blogien ja piensivujen alustana
Alfresco	Kaupallinen ja GPL	+Dokumenttien hallinta +Yhteistyö +Integraatio	-Ilmaiversio on betaversio	Suuren yrityksen tiedonhallintajärjestelmänä
SharePoint	Kaupallinen	+Dokumenttien hallinta +Monipuolinen +Yhteistyö +Integraatiot muihin microsoft-tuotteisiin	-Kallis -Lukittuminen yhteen toimittajaan.	Suuren yrityksen yhteisprojektien alustana ja julkaisuportaalina.

4.8 Järjestelmän valinta

Tässä luvussa käydään läpi sitä, mitä pitää ottaa huomioon valittaessa sisällönhallintajärjestelmää. Valitaanko avoimeen lähdekoodiin perustuva vai kaupallinen järjestelmä? Kaupallisten järjestelmien lisenssien ja implementoinnin hinta voi olla moninkertainen avoimeen lähdekoodiin verrattuna. Se ei silti tarkoita sitä, että ilmaisen avoimen lähdekoodiin perustuvan ohjelman käyttöönotto olisi ilmainen. (Voltolina 2009)

Suuremmat yritykset päätyvät usein kaupalliseen järjestelmään. Kaupalliset ohjelmistot on kehitetty ratkaisemaan näiden ongelmia, siinä missä vapaan lähdekoodin kehitys ei ole niin keskitetysti johdettua. Kaupalliset järjestelmät ovat usein aikaisemmin olleet dokumentinhallintaan ja yhteistyössä kehitettävän (eng. collaboration) sisällön hallintaan tarkoitettuja ohjelmia ja ovat kehittyneet sitä kautta verkkojulkaisun puolelle. Vapaan lähdekoodin järjestelmissä ei ole 24/7 tukea ongelmien sattuessa, vaan on turvauduttava foorumeihin tai vastaavaan. Myös koulutuksen ja konsultoinnin saanti on puutteellista verrattuna kaupallisiin ohjelmiin. Vapaan lähdekoodin ohjelmistoissa on myös se vaara, että niiden kehitys yhtäkkiä lopetetaan. (Voltolina 2009 ja Gottlieb 2005)

Yhtä yleispätevää listaa ominaisuuksista, joita järjestelmän tulisi sisältää, ei voida määritellä. Jokaisella organisaatiolla on omat tarpeensa. Jos tarkoitus on hankkia ison organisaation sisällönhallintajärjestelmä, on tärkeää että kaikki osastot liittyvät hankintaprosessiin ja tekevät omat vaatimusmäärittäyksensä. Tärkeintä on tunnistaa, miksi järjestelmää tarvitaan ja mitkä ovat yrityksen sille asettamat liiketoiminnalliset tavoitteet. Isoa järjestelmää valittaessa on sen myös integroiduttava yrityksen nykyisiin järjestelmiin. (Robertson 2001)

Avoimeen lähdekoodiin perustuvia järjestelmiä on käytössä laajalti pienissä ja keskisuurissa organisaatioissa, joiden tarpeet ovat samankaltaiset. Avoimen lähdekoodin ohjelmia on mahdollista räätälöidä myös oman väen piirissä. Usein isot organisaation sisällä räätälöidyt järjestelmät perustetaan jonkun vapaan lähdekoodin järjestelmän päälle. Kaikkia, myös kaupallisia järjestelmiä on räätälöitävä ainakin hieman. Vähintään tarvitaan ulkoasultaan muokattu sivupohja. (Gottlieb 2005) Yksi tärkeimmistä kriteereistä avoimen lähdekoodin järjestelmään päädyttäessä on se, että sen kehitys tulee olla aktiivista. Jos

järjestelmän kehitys loppuu, pakottaa se käyttävän organisaation joko siirtymään toiseen järjestelmään, tai alkamaan kehittää sitä organisaation sisällä. (Michelinakis 2004, 32)

Avoimeen lähdekoodiin perustuvat järjestelmät rakennetaan useimmiten noudattamaan kansainvälisiä standardeja ja niitä voi siksi helpommin integroida keskenään. Kaupalliset sovellukset ovat usein taas tahallaan sopimattomia muiden järjestelmien kesken pitääkseen asiakkaan kourussa valittuun tekniikkaan ja toimittajaan. (Michelinakis 2004, 30)

5 WEB-ARKKITEHTI SISÄLLÖNHALLINTAJÄRJESTELMÄ

5.1 Järjestelmän kuvaus

Web-Arkkitehti sisällönhallintajärjestelmä on tarkoitettu yrityksille ja yhteisöille, jotka tarvitsevat itselleen yksinkertaiset, mutta näyttävät verkkosivut. Järjestelmän etuja on se, ettei uuden sivuston luomiseen tarvitse kirjoittaa koodia, vaan kaiken pystyy luomaan täysin järjestelmän tarjoamilla työkaluilla. Tämä alentaa sivujen tekijältä vaadittavaa teknisen osaamisen tasoa. Asiakkaan tarvitsee täyttää ainoastaan tilauslomake ja toimittaa sisältömateriaali, jonka jälkeen hän saa valmiin sivuston käyttöönsä viimeistään kahden viikon kuluttua. Asiakas voi myös valita vaihtoehdon, että tekee sivut itse järjestelmällä. Tällöin asiakkaalta laskutetaan ainoastaan kuukausimaksut.

Web-Arkkitehti on luonteeltaan hieman erilainen, kun teoriapuolella esitetyt järjestelmät. Siinä missä siellä esitetyiden järjestelmien käyttöönottoon tarvitaan aina asiantuntijoita, tähtää tämä järjestelmä siihen, että sillä voi pystyttää itselleen sivut ilman teknistä tietämystä. Ero johtuu siitä, että useimmat sisällönhallintajärjestelmät pyrkivät palvelemaan suurta joukkoa erilaisia käyttäjäryhmiä. Me taas tietoisesti suuntaamme tuotteemme pienemmille organisaatioille. Tästä syystä myös teoriapuolella painotettu työnkulku on järjestelmässä pienemmässä roolissa.

Tähän asti järjestelmää on käytetty vain omissa projekteissamme, mutta pyrimme uudella järjestelmällä pääsemään tilanteeseen, jossa palveluun liittyy uusia asiakkaita ilman meidän omaa työpanosta. Web-Arkkitehdin kilpailijoita ovat esim. Kotisivukone ja Elisa Markkina-avain. Pyrkimys järjestelmästä on tehdä sellainen, että sivujen teko onnistuisi sillä peruskäyttäjältä, mutta että se vastaisi myös vaativamman käyttäjän tarpeisiin.

5.2 Kehitysprosessi

Järjestelmän kehitystavaksi valittiin tyyli, jossa ensimmäisellä kierroksella tehdään järjestelmän ydinkomponentit valmiiksi. Ydinkomponenteiksi luetaan sivupohjasysteemi,

sivujen editointi, käyttäjien hallinta ja tiedostojen hallinta. Tämän lisäksi järjestelmään rakennetaan rajapinta, jolla sivumoduulit tullaan liittämään sivuille. Ensimmäisellä kierroksella käyttöliittymä rakennetaan suoraan käyttäen jQuery UI:n tarjoamia työkaluja.

Toisella kierroksella mukaan otetaan graafikko ja käyttöliittymäsuunnittelija. Eri ydinkomponenttien käyttöliittymät suunnitellaan ja muokataan lopulliseen ulkoasuunsa. Jokaisessa näkymässä havaittuja puutteita korjataan. Samalla tehdään myös järjestelmän tärkeimmät maksulliset sivumoduulit. Näitä ovat kielten hallinta, uutiset ja tilastot. Tässä vaiheessa järjestelmään lisätään myös näkymä, jossa asiakkaat voivat tilata sivuilleen moduuleita. Toisen vaiheen viimeisenä tehdään liittymä, jonka avulla voi maksaa tilaamansa moduulit.

Uskalsimme valita tämän tyyllisen lähestymistavan järjestelmän rakentamiseen, sillä kyseessä on Web-Arkkitehdin toinen versio. Ensimmäisen version puutteet ovat hyvin tiedossa ja käyttäjiltä saaman palautteen ansiosta suurimmat pullonkaulat on tiedostettu. Riskinä tässä mallissa oli kuitenkin se, että käyttöliittymään tulee niin suuria muutoksia toisella kierroksella, että ensimmäisen kierroksen työtä on mennyt hukkaan. Tämän tyylin etuna on kuitenkin se, että kun järjestelmää pääsee jo testaamaan, on kaikkien helpompi hahmottaa miten käyttöliittymän tulisi toimia.

5.3 Sisällönhallintajärjestelmän ominaisuudet

5.3.1 Yleistä

Järjestelmän koodi on pääasiassa PHP:ta ja Javascriptiä. Palvelinalusta on Apache ja tietokanta MySQL. Olemme pyrkineet löytämään järjestelmään mahdollisimman hyviä vapaaseen lähdekoodiin perustuvia palasia aina kun sellaisen käyttö on mahdollista. Näitä ovat mm. sivueditori, kuvien editointi, sivurakenteen *raahaa ja pudota* -muokkaus. Järjestelmää testataan pääasiassa Firefox ja Internet Explorer selaimissa. Muista selaimista Chrome toiminee, mutta toimintaa muilla selaimilla ei varmisteta.

5.3.2 Käyttäjän todentaminen

Käyttäjä voi kirjautua oman sivunsa ylläpitopuolelle Web-Arkkitehdin kotisivujen

yläkulmasta. Käyttäjä kirjoittaa käyttäjätunnuksen ja salasanan ja lähettää tiedot palvelimelle. Käyttäjän antamia tunnuksia verrataan tietokannasta löytyviin. Jos vastaavat tunnukset löydetään, kirjoittaa järjestelmä käyttäjän koneelle cookien eli keksin. Keksi on selaimen käyttämä tiedosto, johon tallennetaan istunnon tiedot. Myös tietokantaan kirjoitetaan istunnon tiedot, jota verrataan keksistä löytyvään. Jokaisen tarkistuksen yhteydessä keksissä oleva aikaleima päivittyy. Jos leimaa ei päivitetä tietyn aikarajan sisään, vanhenee istunto automaattisesti. Tämän jälkeen käyttäjän on kirjaututtava järjestelmään uudestaan sisään.

5.3.3 Tietoturva

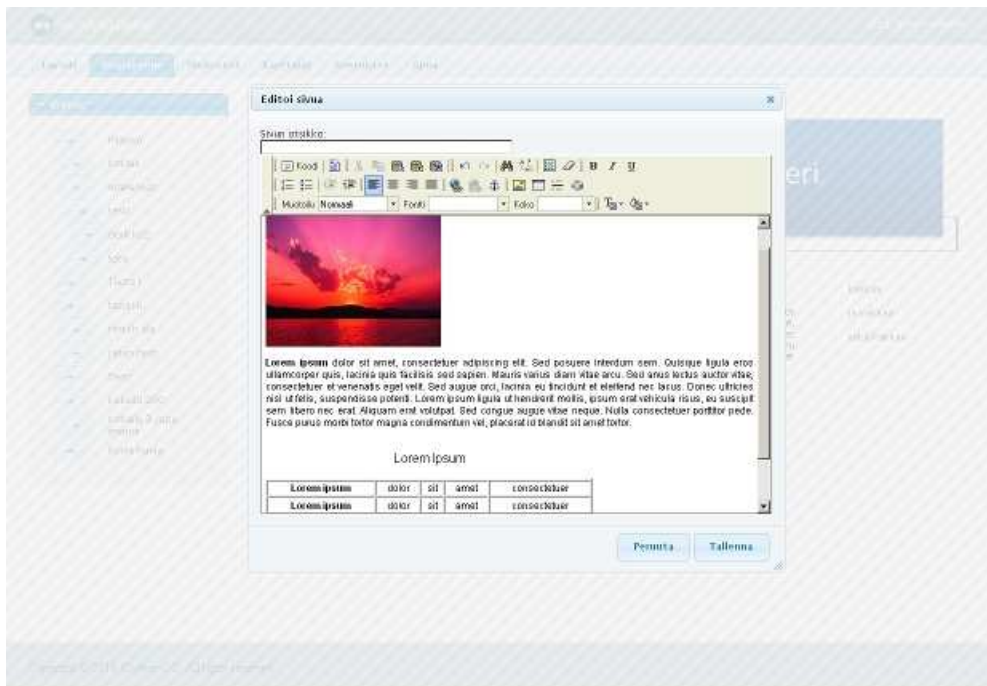
Kaikki sivuston ja ylläpitopuolen tietokantakyselyihin osallistuvat muuttujat ajetaan varmistusfunktion läpi, joka poistaa SQL-injektion mahdollisuuden. SQL-injektio on turva-aukko, jossa hakkeri laittaa tietokantakyselyyn osallistuvaan muuttujaan tahallaan SQL-lauseen. Salasanat on tietokannassa kryptattu, eikä niitä saa avattua selkokieliseksi, vaikka jotenkin onnistuisi saamaan tietokannasta listauksen käsiinsä. Tiedostojen latauksessa käytössä on lista sallituista tiedostopäätteistä, ettei käyttäjän ole mahdollista ladata palvelimelle mitään mahdollisesti vahingollista tiedostoa. Tällä hetkellä sivuston ylläpitopuolen liikenne ei ole salattua, mutta se saatetaan ottaa käyttöön vielä ennen järjestelmän julkaisua.

5.3.4 Käyttöliittymä

Järjestelmän käyttöliittymä on rakennettu hyödyntäen jQuery UI:n toimintoja. Varsinkin kirjaston tarjoama dialogi-toiminto on käytössä ahkerasti. Koko järjestelmä ladataan kerralla käyttäjälle. Tämän jälkeen aina kun käyttäjä tekee jotain, päivitetään ainoastaan tarvittava osa AJAX-komennolla. Tämän avulla vältämme turhia sivulatauksia ja käyttäjä ei koe käyttöä hidastavia viiveitä. Käyttäjälle tarjotaan mahdollisuus lukea pikaohjeita useimmista eri toiminnoista. Pikaohje näytetään automaattisesti puhekuplaa muistuttavassa laatikossa, kun käyttäjä vie hiiren kysymysmerkillä varustetun kuvakkeen päälle. Tarkoitus on myöhemmin tehdä kattavampi ohjeisto, josta löytyy myös opastusvideoita. Järjestelmän peruskäyttö (sivujen päivitys) tulee olla niin yksinkertainen, että sen käyttö onnistuu tavalliset ATK-taidot omaavalta henkilöltä.

5.3.5 Sivujen editointi

Sivun editointitilassa sivurakenne on näkyvillä vasemmalla ja sivuston sivupohja oikealla. Sivua voi vaihtaa oikealta olevasta menusta. Vasemmalta sivupohjasta menurakenteen toiminta on estetty. Viemällä hiiren elementin päälle tulee näkyviin toimintoja, joista tärkein on sivun muokkaus. Sivua voidaan muokata editorissa, joka avataan dialogiin. Kuvassa 3 on sivu avattu editointitilaan. Tallennuksen jälkeen dialogi sulkeutuu ja käyttäjä näkee välittömästi miltä sisältö näyttää itse sivuilla. Editoriohjelmana on käytössä vapaaseen lähdekoodiin perustuva FCKeditor. Siitä löytyy yleisimmät tekstin editointiin tarvittavat työkalut. Käyttäjä voi lisätä myös kuvia ja muita tiedostoja sivulle editorin kautta. Kuvien ja muiden tiedostojen selaukseen on ostettu FCKeditoriin sopiva lisäosa CKfinder.

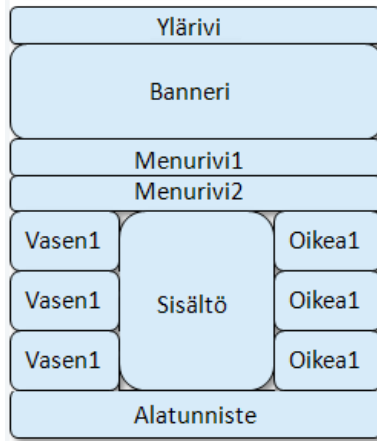


Kuva 3. Sivun editointi

5.3.6 Sivupohjan muokkaus

Sivupohjan muokkaus on järjestelmän isoin uudistus. Vanhassa järjestelmässä oli valittavissa erilaisia sivupohjia, joita pystyi muokkaamaan. Muokkaamalla sai kyllä tehtyä pohjista erinäköisiä, mutta niissä oli silti liikaa rajoitteita. Sivujen ulkoasu rakentaneet

graafikot kaipasivat uusia mahdollisuuksia ja sivujen omistajat halusivat sivuille poikkeuksia, joita ei pystytty toteuttamaan.



Kuva 4. Sivupohjan elementit

Uudessa järjestelmässä ei ole useita sivupohjia, vaan ainoastaan yksi pohja jota pystyy muokkaamaan monipuolisesti. Yhdellä sivupohjalla pystyy tekemään nyt kaikki mitä vanhoilla yhteensä ja paljon enemmän. Sivupohja koostuu ennalta määräytyistä elementeistä, joita käyttäjä voi valita tai jättää valitsematta omalle sivulleen. Kuvassa 4 näkyy sivupohjasta löytyvät elementit. Jokaisen elementin sisältö on täysin käyttäjän päätettävissä. Elementtiin voi laittaa kuvan, tekstiä, sitoa sivuun tai sen voi kytkeä kiinni johonkin sivumoduuliin kuten menuun tai uutisiin. Mitään elementtiä ei tarvitse käyttää siihen mihin se on tarkoitettu, vaan jätämme käyttäjille vapaat kädet ulkoasumuokkauksissa. Sivustolle voi tehdä myös sivukohtaisia poikkeuksia, joka mahdollistaa sen, että esim. etusivu voi näyttää täysin erilaiselta kun muut sivut. Tämä on toteutettu niin, että ensin ladataan tietokannasta yleinen sivupohja, jonka jälkeen ladataan sivukohtaiset asetukset.

Jokaisen elementin CSS-arvoja on mahdollista muokata järjestelmään tehdyllä tyylimuokkaimella. Jokaisen elementin leveyttä ja korkeutta voi muokata, rajoja voi säätää, voi laittaa taustakuvan ja niin edelleen. Myös ulkoisen CSS-dokumentin lataaminen palvelimelle on mahdollista, jos tyyli työkalulla ei pysty johonkin mitä käyttäjä haluaa. Pyrkimys on kuitenkin tehdä tyyli työkalusta niin kattava, ettei tähän olisi tarvetta. Käyttäjän muokkaama sivu on nähtävillä valikon vieressä ja jokainen muutos näkyy ilman viivettä.

5.3.7 Sivurakenteen muokkaus

Vanhassa järjestelmässä menurakenteen järjestystä vaihdettiin pienistä nuolista, jotka sijaitsevat menuotsikoiden vieressä. Osa käyttäjistä ei löytänyt nuolia ollenkaan. Osa niistä jotka löysi, eivät taas ymmärtäneet niiden toimintaa oikein. Uudessa järjestelmässä sivujärjestystä voi muuttaa avaamalla dialogin, jossa sivujen paikkaa voi vaihtaa *raahaa ja pudota*-mallin mukaan. Käyttäjä voi asetella sivut haluamaansa järjestykseen. Uusi järjestys otetaan käyttöön vain, jos käyttäjä tallentaa muutokset.

5.3.8 Käyttäjien hallinta

Käyttäjien hallinta on säilytetty yksinkertaisena. Uusilta käyttäjiltä kysytään nyt myös sähköposti siltä varalta, että käyttäjä unohtaa salasansa. Uutena ominaisuutena tuotiin käyttäjätasot. Käyttäjän rooli voi olla Ylläpitäjä, editoija tai sivuston käyttäjä. Ylläpitäjä pystyy tekemään sivustolla kaikkia muutoksia. Editoija pystyy ainoastaan lisäämään uusia sivuja ja muokkaamaan vanhoja. Sillä ei pysty muokkaamaan sivupohjan ulkoasua tai lisäämään uusia käyttäjätunnuksia. Sivuston käyttäjällä ei pääse ylläpitopuolelle ollenkaan. Nämä käyttäjätunnukset on varattu siihen, että sivuille tehdään salasanalla suojattuja osioita. Harkinnassa on lisätä vielä rajoitettu editoija, joka pääsisi tekemään muutoksia vain ylläpitotunnuksilla tälle erikseen oikeudet annettuihin sivuihin.

5.3.9 Tiedostojen hallinta ja editointi

Järjestelmä sisältää yksinkertaisen työkalun tiedostojen listausta varten (kuva 5). Tiedostoja voi tuhota ja ladata palvelimelle sekä eri kansioden välillä pystyy siirtymään helposti. Tiedostojen lataukseen on myös lisätty palkki, joka kertoo kuinka paljon kuvasta on siirtynyt palvelimelle. Palkki on toteutettu Flash-tekniikalla. Vanhassa järjestelmässä käyttäjä joutui kuvaa ladatessaan katsomaan jähmettynyttä ruutua pitkäänkin, jos ladattava tiedosto sattui olemaan iso ja tähän saatiin nyt parannus.



Copyright © 2010 Codens OY. All rights reserved.

Kuva 5. Tiedostojen selaustyökalu.

Kuvatiedostojen kohdalla on erikseen mahdollisuus editointiin. Vanhassa järjestelmässä ei ollut mahdollista tehdä kuvalle mitään, ja se oli monen käyttäjän mielestä hankalaa. Kuva tulee usein ladattua palvelimelle liian suurena, ja käyttäjät eivät jaksaa pienentää niitä ensin omalla koneellaan kuvankäsittelyohjelmalla. Uudessa järjestelmässä kuvia pystyy pienentämään, rajaamaan, kääntämään ja niiden väritystä voi hieman säätää.

6 TULOKSET, POHDINTA JA TULEVAISUUS

Teoriapuolella huomattiin, kuinka monitahoinen kokonaisuus sisällönhallinta on. Se liittyy lähes kaikkiin yrityksen prosesseihin. Pienemmillä organisaatioilla voi web-sisällönhallinta olla vielä erillinen toiminto, jos sisältöä tuotetaan vaan muutaman henkilön voimin. Isommissa organisaatioissa yrityksen tietoa tuottaa ja käsittelee niin laaja joukko, että ilman järkevää dokumenttien-, versioiden- ja työnkulunhallintaa kokonaisuutta on mahdoton hallita. Isommille yrityksille web-sisältö on vain osa isompaa kokonaisuutta ja yksi lukuisista kanavista johon informaatiota tuotetaan. Tämän takia myös web-sisällönhallintaa voi lähestyä monesta eri suunnasta ja siihen on olemassa paljon erilaisia eri tilanteissa hyvin toimivia ratkaisuja. Web-ohjelmistotekniikka noudattaa suurelta osin samoja periaatteita kun perinteinen sovelluskehitys. Varsinkin viime vuosina web-kehityksen työkalut ovat kehittyneet tasolle, jossa voidaan tehdä samantasoisia käyttöliittymiä kun perinteisissä ohjelmistoissa.

On olemassa lukematon määrä erilaisia sisällönhallintajärjestelmiä, niin avoimeen lähdekoodiin perustuvia kun kaupallisia. Avoimen lähdekoodin ohjelmien ominaisuudet saavuttavat hiljalleen kaupallisia sovelluksia myös työnkulun puolella. Avoimen lähdekoodin ohjelmien kehitys on nopeaa ja niissä voi olla satoja samanaikaisia kehittäjiä. Syy mikseivät kaikki valitse ilmaisia ohjelmia on se, että lisenssikustannus voi muodostaa vain pienen osan projektin kokonaiskustannuksista. Löytyy myös mielenkiintoisia ohjelmia, jotka pyrkivät yhdistämään näiden molempien maailmojen parhaat puolet. Näissä ohjelmisto on saatavilla avoimeen lähdekoodin perustuen, sekä vaihtoehtoisesti lisenssillä. Tarkoituksena on se, että ne yritykset jotka kaipaavat teknisen tuen ja koulutusta voivat halutessaan saada sen. Samalla avoimuus varmistaa mielenkiintoisten lisäosien tulemisen markkinoille ja tämä lisää sovellusten monipuolisuutta ja houkuttelevuutta. Kaupalliset ohjelmat taas lähestyvät standardeja ja myös niistä löytyy rajapintoja lisäosien kehittäjille. Ilmeisesti ostajat vaativat näitä ominaisuuksia ja kehittäjien on pakko muuttaa toimintatapojaan avoimempaan suuntaan kilpailun kiristyessä.

Käytännön puolella kehitystyö oli pääpiirteissään onnistunutta. Järjestelmä oli tämän työn

puitteissa tarkoitus saada siihen kuntoon, että se olisi valmis tuotantoon. Tähän pisteeseen saakka ei kuitenkaan ehditty. Osittain se johtui siitä, että tuli valittua hieman liian kunnianhimoisen aikataulu työn valmistumiselle, osittain muiden työkiireiden takia ja viimeisenä sen takia, että työn valmistumisella oli tiukka aikaraja. Ehdimme ensimmäisen kehityskierroksen loppuun. Kehitystyö sujui kuitenkin ilman suurempia ongelmia. Etukäteen pelättiin, että eri javascript-kirjastojen liittäminen voi tuoda mukanaan yhteensopivuusongelmia. Toinen huolta aiheuttanut seikka oli se, käytettävät tekniikat ovat vielä kehittyviä. Joitain lastentauteja onkin huomattu. Esimerkiksi FCKeditorin uudempaa versiota ei saatu toimimaan jQueryUI:n dialogissa.

Seuraavaksi järjestelmän kehityksessä siirrytään toiseen vaiheeseen. Siinä käymme koko yrityksen voimin läpi eri toimintoja ja mietimme miten niitä tulisi parantaa. Tämän jälkeen graafikko ja käyttöliittymäsuunnittelija tekevät suunnitelmat jokaisen näkymän toiminnoista. Näitä suunnitelmia sitten aletaan toteuttamaan järjestelmään. Ensimmäinen katselmus on jo pidetty sivupohjasysteemin toiminnasta ja sen pohjalta käyttöliittymään tulee muutos elementtien valintaan. Nyt elementtejä visualisoivasta menusta (kuva 2.) voi valita mitä niistä käyttäjä haluaa editoida. Se muutetaan niin, että siitä voi valita mitä elementtejä sivuille otetaan mukaan ja mitä jätetään pois. Itse editoitava elementti valitaan viemällä hiiri oikealla olevan oikean sivun eri elementtien päälle, jolloin elementin päälle ilmestyy nappi, jota painamalla kyseinen kohta aktivoituu. Toiminta on samantapainen kun se, miten sivun editointi nyt toimii.

Kolmas kierros alkaa siitä, kun järjestelmä on tuotantokelpoinen. Kolmannella kierroksella järjestelmää koitetaan yrityksen omissa oikeissa projekteissa. Järjestelmästä mahdollisesti löytyviä bugeja korjataan. Samalla järjestelmään tuodaan uusia sivumoduuleja. Nopeasti tuotavia sivumoduuleja ovat ainakin tapahtumakalenteri, kuvagalleria ja pikakyselyt. Kun on nähty, että järjestelmä toimii, julkaistaan se kaikille käyttäjille. Tämän jälkeen järjestelmä siirtyy jatkuvan kehityksen aikaan. Se tarkoittaa, että lisäämme järjestelmään uusia moduuleita ja parantelemme sitä pikkuhiljaa. Ensimmäisenä kehityslistalla on verkkokauppa. Teoriaosaa kirjoitettaessa heräsi myös ajatus tuoda mukaan versiohallinta ja työnkulkua helpottavia toimintoja. Yksinkertainen hyväksyttämisen prosessi uudelle sisällölle, versiohistoria ja arkisto tuhotuille sivuille ovat suunnitteilla olevia

ominaisuuksia. Peruskäyttäjiä varten löytyy sivupohjan muokkaukseen oikopolkuja, joista saa tuotua sivupohjaan haluamansa teeman yhdellä klikkauksella. Teema voidaan valita galleriasta, johon on tehty valmiiksi iso määrä erilaisia sivupohjia.

Tulevaisuuden visiossa järjestelmästä löytyy kymmenittäin moduuleja, joita käyttäjä voi testata kahden viikon ajan. Jos käyttäjä pitää moduulista, voi hän ostaa moduulin pieneen kuukausihintaan. Käyttäjäkunta koostuu tavallisista pienistä ja keskisuurista yrityksistä sekä yhdistyksistä. Järjestelmää hyödyntää myös kotisivuja omille asiakkailleen tekevät graafikot, jotka eivät kuitenkaan halua koodata itse sivuja. Näitä varten luomme kumppanuusohjelman. Järjestelmä muistuttaa asiakkaille automaattisesti siitä, kun vuosimaksu on erääntymässä sähköpostilla. Sen ylläpidolliset toiminnot automatisoidaan niin, ettei manuaalisia työvaiheita ole kun pakollisissa tilanteissa.

7 YHTEENVETO

Tässä diplomityössä käsiteltiin web-pohjaisia sisällönhallintajärjestelmiä. Teoriapuolella huomattiin, kuinka sisällönhallinta on monitahoinen yrityksen kaikkiin prosesseihin liittyvä käsite, jota voi lähestyä monesta suunnasta. Web-ohjelmien kehitys noudattaa pitkälti samaa kaavaa kuin perinteinen sovelluskehitys. Erilaisia sisällönhallintajärjestelmiä on olemassa paljon erilaisia. Yhtä yleispätevää listaa järjestelmältä vaadittavista ominaisuuksista on mahdotonta määrittää. Se riippuu pitkälti organisaation tarpeista.

Isot yrityksen suosivat kaupallisia järjestelmiä, sillä niillä lisenssikustannus voi olla vain osa projektin kokonaiskustannusta. Ne tarvitsevat myös käyttötukea, koulutusta ja muita palveluita. Avoimen lähdekoodin järjestelmät ovat laajemmin käytössä pienemmillä organisaatioilla. Nämä järjestelmät ovat myös kehittyneet palvelemaan niitä käyttäjäryhmiä, ja siksi niissä työnkulun, versioinnin ja yhteistyön ominaisuudet eivät ole yhtä kehittyneitä kun kaupallisissa järjestelmissä. On myös olemassa järjestelmiä, jotka yrittävät yhdistää molemmat puolet.

Työn käytännön osuudessa kehitettiin Codens Oy:n sisällönhallintajärjestelmää. Järjestelmän kehitys sujui ilman suurempia ongelmia, mutta siinä ei ehditty niin pitkälle kun alun perin oli tarkoitus. Järjestelmä vaatii vielä lisäkehitystä ennen kuin sen on valmis markkinoille. Järjestelmän käyttöliittymä saatiin paljon paremmaksi, kun kehityksessä hyödynnettiin uusia jQuery:n pohjautuvia työkaluja. Järjestelmän suurimmat parannukset tulivat sen sivupohjasysteemiin. Uudessa järjestelmässä ei ole useita pohjia joista valita, vaan yhdestä pohjasta pystyy muokkaamaan lukemattoman määrän erilaisia suoraan järjestelmän tarjoamilla työkaluilla. Tulevaisuuden visiossa järjestelmästä kehitetään automaattinen verkkojulkaisualusta, johon lisätään monipuolisia ominaisuuksia. Pääpaino tulee edelleen olemaan siinä, että järjestelmän peruskäyttö olisi niin helppoa, että sivujen päivittäminen onnistuu keneltä tahansa.

LÄHTEET

alfresco.com, online: (<http://www.alfresco.com/>) [viitattu 15.5.2010]

Bibeaut B, Yehuda K, jQuery in Action, Manning Publications Co. 2008.

Boehm B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes, 1986

Boiko B, Understanding Content Management Bulletin of the American Society for Information Sciences 28, no. 1 2001.

Boiko B, Why Content Management? A CM Domain White Paper. Metatorial Services Inc. & HungryMinds Inc. 2002.

Brazel A, Wordpress Bible, Wiley publishing, 2010

Browning P, Lowndes M, JISC TechWatch Report: Content Management Systems Joint information systems committee 2001.

Builtwith.com, Web and Internet Technology Usage Statistics, online (<http://trends.builtwith.com/>) [viitattu 20.5.2010]

Byron A, Berry A, Haug N, Eaton J, Walker J, Robbins J, Using Drupal, O'reilly 2008

Chaffer J, Swedberg K, Learning jQuery: Better Interaction Design and Web Development with Simple JavaScript Techniques, Packt 2007.

drupal.org, online: (<http://drupal.org>) [viitattu 15.5.2010]

Goodwin, S. Vidgen, R. Content, content, everywhere...time to stop and think? The process of Web content management, Computing & Control Engineering Journal Apr 2002, Volume: 13 Issue: 2.

Gottlieb S, Content Management Problems and Open Source Solutions, Optaros, Inc. 2005.

Hasanali F, Leavitt P, Content Management: A Guide for Your Journey to Knowledge Management Best Practices, American Productivity & Quality Center 2003

joomla.org, online: (<http://www.joomla.org/>) [viitattu 15.5.2010]

Korpela J, CSS verkkosivujen muotoilussa. Docendo 2008.

Keränen V, Lamberg N, Penttinen J: Web-julkaiseminen & multimedia, Docendo, 2006.

McKeever S, Web content management systems: evolution, lifecycle and market, Industrial Management & Data Systems 2003, Volume: 103, Issue: 9.

Michelinakis, D, Open Source Content Management Systems: An Argumentative Approach, The University of Warwick, Warwick Manufacturing Group, 2004

Mendes E, Mosley N, Web Engineering: Theory and Practice of Metrics and Measurement for Web Development, Springer 2005.

Musciano C, Kennedy B, HTML The Definitive Guide, Third edition O'reilly & Associates 1998.

Petrovic M, Liu H, Jacobsen H, CMS-ToPSS: Efficient Dissemination of RSS Documents, Proceedings of the 31st VLDB Conference, 2005

Pressman R, Software engineering: a practitioner's approach, McGraw-Hill 2001

Robertson J, How to evaluate a content management system, Step Two Designs Pty Ltd 2005 online: (http://www.steptwo.com.au/files/kmc_evaluatecms.pdf) [Viitattu 12.5.2010.]

Rockley A, Kostur P, Manning S, Managing enterprise content: a unified content strategy, Pearson Education, Inc., 2003

Salminen Airi, Metatiedot organisaatioiden sisällönhallinnassa, RASKE2-projektin II väliraportti (s. 4-13). Eduskunnan kanslian julkaisu, 2005

Sharepoint.microsoft.com, online: (<http://sharepoint.microsoft.com/en-us/Pages/default.aspx>) [viitattu 15.5.2010]

Vaidyanathan G, Mautone S, Security in dynamic Web Content Management Systems Applications, Communications of the ACM 2009, Volume: 52 Issue: 12.

Veen J, Why Content Management Fails, Adaptive Path 2004 Online (<http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/ADTVPATH/A040401V.pdf>) [Viitattu 12.5.2010].

Vidgen R, Goodwin S, Barnes S, Web Content Management, 14th Bled Electronic Commerce Conference 2001.

Voltolina V, CMS Primer: Open Source vs. Commercial, Online 2009
(<http://www.foliomag.com/2009/cms-primer-open-source-vs-commercial>) [viitattu 15.5.2010]

W3Schools, browser statistics 2010. online:
(http://www.w3schools.com/browsers/browsers_stats.asp) [viitattu 15.5.2010]

Woojong S, Web engineering: principles and techniques, Idea Group publishing 2005.
wordpress.org, online: (<http://wordpress.org/>) [viitattu 15.5.2010]

Sæther Bakken S, Schmid E, PHP Manual, PHP Documentation Group 2003

Wellman D, jQuery UI 1.6 The User Interface Library for jQuery, Packt 2009