

Opinnäytetyö (AMK)

Tietotekniikka

Peliteknologia

2017

Teemu Aro

# DRUPAL-SIVUSTON ASENNUKSEN AUTOMATISOINTI

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikka | Peliteknologia

Toukokuu 2017 | Sivumäärä 29

Ohjaaja: Yliopettaja, FT Mika Luimula

Teemu Aro

# DRUPAL-SIVUSTON ASENNUKSEN AUTOMATISOINTI

Manuaalisessa Drupal-sivustoasennuksessa on monia askelia, jotka käyttäjän on tehtävä tuottaakseen toimivan sivustorungon. Useimmat näistä askelista tehdään samalla tavalla kaikissa projekteissa, joten prosessin pystyy suurimmaksi osin automatisoimaan. Opinnäytetyössä on tuotettu menetelmä, jonka kanssa sivuston asennuksen pystyy tekemään nopeammin Drupal-asennusprofiilin avulla.

ASIASANAT:

Julkaisualustat, Julkaisujärjestelmät, Internet, Selaimet

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Game Technology

May 2017 | 29 pages

Instructor: Principal Lecturer, Ph. D. Mika Luimula [Click here to enter text.](#)

Teemu Aro

# AUTOMIZATION OF DRUPAL SITE INSTALLATION

[Click here to enter text.](#)

Manual Drupal site installation has multiple steps which user must take to generate a working website base. Most of these steps are the same in every project, so the process can be mostly automated to speed it up. In this thesis a method has been created to automate the installation process through a Drupal installation profile.

KEYWORDS:

Content Management System, Browser, Metadata, Internet

# SISÄLTÖ

<b>KÄYTETYT LYHENTEET JA SANASTO</b>	<b>6</b>
<b>1 JOHDANTO</b>	<b>7</b>
<b>2 JULKAISUJÄRJESTELMÄ, SISÄLLÖNHALLINTAJÄRJESTELMÄ</b>	<b>8</b>
2.1 Sisällönhallintajärjestelmän ominaisuudet	9
2.2 Sisällönhallintajärjestelmien kustomointi	11
<b>3 AVOIN VAI SULJETTU LÄHDEKOODI</b>	<b>13</b>
<b>4 DRUPAL</b>	<b>16</b>
4.1 Drupalin historia	16
4.2 Drupalin ydin	16
4.3 Drupal-moduulit	16
4.4 Drupalin yhteisöllisyys	17
4.5 Drupal tapahtumat	17
<b>5 DRUPAL-ASENNUKSEN AUTOMATISOINTI</b>	<b>19</b>
5.1 Tehtävänasettelu	19
5.2 Haasteet ja riskit	19
5.3 Drontti Oy	20
<b>6 DRUPAL-ASENNUSPROFIILI</b>	<b>21</b>
6.1 Tarvittavien moduulien asennus	22
6.2 Pohjateeman ja oman teeman asennus	22
6.3 Käyttäjäasetukset	23
6.4 Sisällönhallinta-asetukset	24
6.5 Muut asetukset ja monikielisyysasetukset	24
6.6 README.txt ja asennusprofiilin käyttö	25
<b>7 YHTEENVETO</b>	<b>27</b>
<b>LÄHTEET</b>	<b>29</b>

## KUVAT

Kuva 1. Suosituimmat sisällönhallintajärjestelmät heinäkuussa 2016 (IT-parkki 2016).	9
Kuva 2. SASS-koodin (vasemmalla) kääntäminen CSS-koodiksi (oikealla).	11
Kuva 3. Avoimen ja suljetun lähdekoodin vahvoja puolia.	13
Kuva 4. Suljettu vastaan avoin lähdekoodi (Open Source Development (OSS) under Eclipse Public License (EPL)).	14
Kuva 5. Esimerkkikuva seminaarista DrupalCamp Balticsista vuodelta 2016. (Twitter.com. DrupalCampBaltics seminar 2016.)	17
Kuva 6. Esimerkkikuva koodisprintistä DrupalCamp Balticsista vuodelta 2015. (Twitter.com. Lauri Eskola: DrupalCampBaltics 2015 sprint room.)	18
Kuva 7. Asennusprofiilin tiedostot Drupal-asennuksessa.	21
Kuva 8. Webin suosituimmat viitekehukset.	23
Kuva 9. README.txt	25

## TAULUKOT

Taulukko 1. Sisällönhallintajärjestelmien ohjelmointikielet	10
---	----

## KÄYTETYT LYHENTEET JA SANASTO

CMS	Content Management System, sisällönhallintajärjestelmä
Drupal	Drupal sisällönhallintajärjestelmä
Drush	Drupal Shell, Drupalin komentorivityökalu
PHP	PHP: Hypertext Preprocessor, Drupalin käyttämä koodikieli

# 1 JOHDANTO

Drupal-sisällönhallintajärjestelmässä voidaan uusi sivusto luoda joko käyttämällä oletus-asennusprosessia, tai manuaalisesti alusta alkaen. Oletusasennusprosessi tuottaa toimivan sivustorungon nopeasti, mutta se on erittäin pelkistetty, eikä helpolla laajennettavissa.

Manuaalinen sivuston luominen vie aikaa huomattavasti enemmän, mutta sen kanssa pääsee parempiin tuloksiin. Siinä on useita askelia, jotka useimmissa sivustoissa toteutetaan kuitenkin samalla tavalla.

Opinnäytetyön tarkoituksena on tutkia ja luoda Drupal-sivuston luontia automatisoiva ja nopeuttava menetelmä. Työ toteutetaan Drupal-asennusprofiilina, joka antaa työkalut PHP:llä sivuston hallintaan.

## 2 JULKAISUJÄRJESTELMÄ, SISÄLLÖNHALLINTAJÄRJESTELMÄ

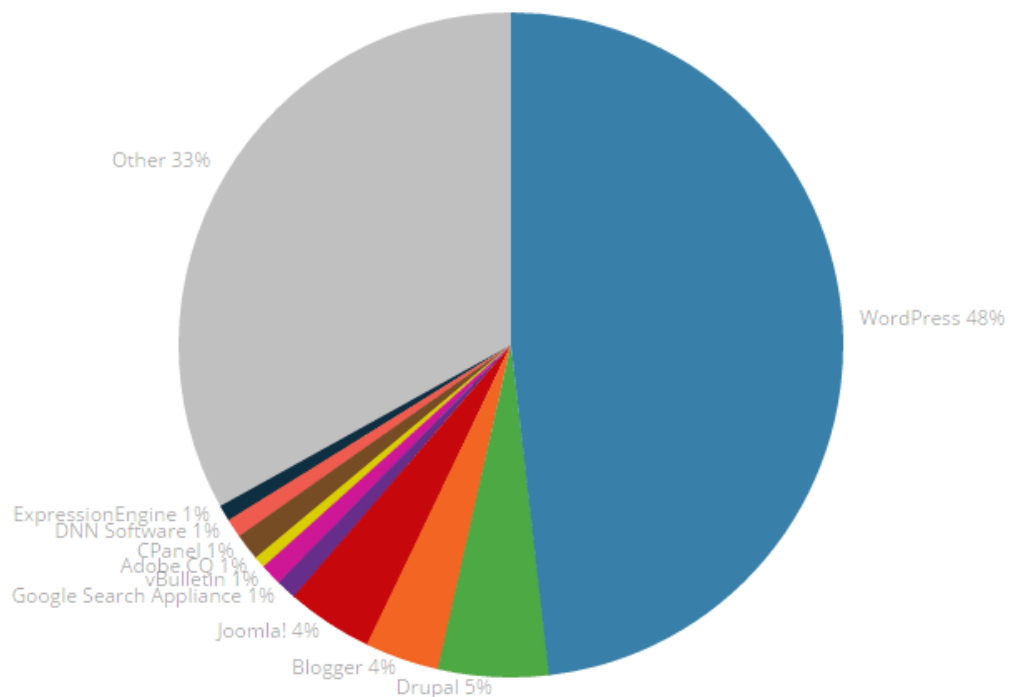
Julkaisujärjestelmä ja sisällönhallintajärjestelmä ovat nimityksiä erilaisille järjestelmille, jolla on tarkoitus julkaista tai ylläpitää sisältöä internetissä. Nimityksiä käytetään useasti monin tavoin ristiin ilman sen suurempaa kategorisointia, mutta karkeasti sanottuna järjestelmät voisi jakaa julkaisu ja sisällönhallintajärjestelmiin niiden syvällisyyden mukaan (Tolvanen 2009). Julkaisujärjestelminä pidetään kevyempiä verkkosivustoja tai -järjestelmiä, esimerkiksi erilaiset blogit tai kuvagalleriat, kun taas sisällönhallintajärjestelmät keskittyvät nimenomaan sisällön hallintaan ja ylläpitoon. Tällaisia järjestelmiä ovat esimerkiksi monet pankkipalvelut, verkkokaupat ja sosiaalisen median palvelut.

Käytännössä siis julkaisu- ja sisällönhallintajärjestelmät ovat pohjimmiltaan varsin samankaltaisia ja erot tulevatkin vasta loppumetrieni erikoistumisesta eri osa-alueisiin. Termejä voidaan käyttää hyvin ristiin, ja tässä opinnäytetyössä käytetään jatkossa lähinnä sisällönhallintajärjestelmä-termiä, koska työn työkaluna oleva Drupal voidaan parhaiten kategorisoida juuri sisällönhallintajärjestelmäksi. Sisällönhallintajärjestelmä-termin voisi ymmärtää kattavan myös muut kuin verkon yli hallittavat järjestelmät (esimerkiksi sulautetun järjestelmän tietokannan hallintajärjestelmä), mutta termi on vakiintunut käytettäväksi nimenomaan verkon yli käytettävistä järjestelmistä. Samaan tapaan myös englanninkielinen termi content management system (lyhennettynä CMS) on vakiintunut tarkoittamaan nimenomaan verkkosivustojen taustajärjestelmiä. Niinpä tässäkin opinnäytetyössä käsitellään nimenomaan verkkosivujen taustajärjestelmiä, kun puhutaan sisällönhallintajärjestelmistä.

Maailma on tällä hetkellä täynnä erilaisia sisällönhallintajärjestelmiä ja niitä kehitellään koko ajan lisää eri tarpeisiin. Maailmalla on käytössä niin useilla sadoilla tuhansilla käytössä olevia suuria järjestelmiä kuin pieniä, esimerkiksi yrityksen itse rakentamia ja ylläpitämiä sisällönhallintajärjestelmiä. Järjestelmien lisenssikulut voivat olla mitä vaan muutamien eurojen ja kymmenien tuhansien välillä, tai järjestelmä voi olla lisensoitu avoimena lähdekoodina, jolloin kenellä tahansa on pääsy järjestelmän lähdekoodiin ja siten itse järjestelmään ilman lisenssimaksuja.



Suosituimmat sisällönhallintajärjestelmät ovat tällä hetkellä avointa lähdekoodia olevat WordPress, Drupal ja Joomla, mutta kuten Kuva 1 esittää, on sisällönhallintajärjestelmien kenttä hyvin pirstaloitunut WordPressin dominoinnin jälkeen (IT-parkki 2016). Avoimen lähdekoodin järjestelmään tuova ilmaisuus on toki luonnollinen myyntivaltti, eikä ilmaisten ohjelmistojen hallintaa markkinoilla voi pitääkään varsinaisena yllätyksenä. Yritysten ja yksityisten kehittäjien on helppo tarttua avoimen lähdekoodin järjestelmään ja testata sopiiko se omiin tarpeisiin.



Kuva 1. Suosituimmat sisällönhallintajärjestelmät heinäkuussa 2016 (IT-parkki 2016).

## 2.1 Sisällönhallintajärjestelmän ominaisuudet

Sisällönhallintajärjestelmän tärkein tehtävä on säilöä tietoa turvallisessa ja ylläpidettävässä muodossa, sekä mahdollistaa sen näyttäminen ja manipulointi internetin yli. Tällöin lähtökohtaisesti tieto tallennetaan sivuston taustalla pyörivään tietokantaan, ja noudetaan sieltä ja näytetään käyttäjälle sisällönhallintajärjestelmän avulla. Osa sisällönhallintajärjestelmien ominaisuuksista on myös...

lintajärjestelmistä pyörii ilman tietokantaa, esimerkiksi kirjoittamalla tieto suoraan teksti-tiedostoihin, mutta vähänkään monimutkaisemmissa sivustoissa tietokanta on tiedon tallentamisen oletus.

Tietokannan dataa manipuloidaan jonkin ohjelmointikielen avulla. Tyypillisesti sisällönhallintajärjestelmien ohjelmointikieli on ollut PHP, jolla muun muassa Drupal, Joomla ja WordPress on kirjoitettu, mutta myös muita käytetään aktiivisesti. Esimerkiksi Python-pohjainen Django-sisällönhallintajärjestelmä ja Java-pohjainen Liferay ovat aktiivisessa käytössä Suomessa ja maailmalla. Taulukko 1:ssä on esitetty yleisimpien sisällönhallintajärjestelmien ohjelmointikielät.

Sisällönhallintajärjestelmä	ohjelmointikieli
Drupal	PHP
Joomla	PHP
WordPress	PHP
Django CMS	Python
Liferay	Java
DNN	ASP.NET

Taulukko 1. Sisällönhallintajärjestelmien ohjelmointikielät

Myös käyttäjänhallinta on erittäin olennainen osa sisällönhallintajärjestelmän toimintaa. Monilla sivustoilla on elintärkeää pystyä luomaan erilaisia käyttäjiä ja käyttäjäryhmiä, joilla on eri tason oikeuksia nähdä, luoda ja muokata sivuston sisältöä ja toiminnallisuuksia. Näin ylläpidon lisäksi voidaan päästää tavalliset käyttäjät järjestelmän dataan kiinni, jolloin verkkopalveluista saadaan monimuotoisia ja interaktiivisia, eikä sivuston sisältö ole pelkästään ylläpitäjien tai kehittäjien tuottamaa.

Loppujen lopuksi kuitenkin sisällönhallintajärjestelmien tärkein ominaisuus on luotettavuus ja tietoturvallisuus. Järjestelmän pitäisi olla jatkuvasti käytettävissä, eikä järjestelmässä saisi olla ammottavia tietoturva-aukkoja. Järjestelmä ei missään tilanteessa saa päästää ketä tahansa kaikkeen tietoon kiinni, saati sitten muuttamaan sivuston toimintaa. Verkkosivuston sisällönhallintajärjestelmän tietoturva onkin nykypäivänä aikamoista tasapainottelua uusien ominaisuuksien ja tietoturva-aukkojen välillä.

## 2.2 Sisällönhallintajärjestelmien kustomointi

Sisällönhallintajärjestelmiä kustomoidaan monin eri tavoin. Likiptäen aina sivustolle luodaan oma teema tai tyyli CSS-tiedostojen (Cascading Style Sheets) avulla, jotka webbrowseri tulkitsee ja näyttää graafisesti. CSS on nykypäivän koodikieliksi varsin vanhentunut syntaksiltaan, ja siksi nykyään onkin tapana kirjoittaa esimerkiksi SASS- tai LESS-koodia, joka käännetään CSS-tiedostoiksi. Kuvassa Kuva 2 on esitetty mitä SASS-koodin kääntäessä CSS-koodiksi tapahtuu. SASS ja LESS tarjoavat monia nykypäivän koodikielelle perinteisiä metodeja, jotka alkeellisesta CSS:stä puuttuvat, kuten muuttujat, silmukat ja koodin sisentämisen.

<pre> nav {   ul {     margin: 0;     padding: 0;     list-style: none;   }    li { display: inline-block; }    a {     display: block;     padding: 6px 12px;     text-decoration: none;   } } </pre>	<pre> nav ul {   margin: 0;   padding: 0;   list-style: none; }  nav li {   display: inline-block; }  nav a {   display: block;   padding: 6px 12px;   text-decoration: none; } </pre>
--	--

Kuva 2. SASS-koodin (vasemmalla) kääntäminen CSS-koodiksi (oikealla).

Sivuston toiminnallisuuden muokkaaminen tehdään monimutkaisissa verkkopalveluissa lähtökohtaisesti pohjajärjestelmään valitulla koodikielellä. Toki järjestelmä pyrkii tarjoamaan mahdollisimman paljon erilaisia asetuksia ja vaihtoehtoja, joilla sivuston perusmuokkausta pystytään tekemään. Siistejä ja toimivia perusverkkosivustoja saakin toki luotua suoraan pohjajärjestelmän tarjoamalla valmiilla työkaluilla, mutta varsinkin isommissa ja yksilöllimmissä verkkopalveluissa on monesti tarve uusille tai erilaisille ominaisuuksille. Näinollen esimerkiksi Drupal- ja WordPress-ratkaisuissa pohjajärjestelmän

kustomointia tehdään lähtökohtaisesti PHP:lla, kun taas Djangossa Python-koodikielellä ja Liferay-järjestelmässä Java-kielellä.

Useimmiten myös sivuston taustalla pyörivää tietokantarakennetta modifioidaan jollain keinoin. Vaikka tietokanta useimmiten on MySQL- tai MariaDB-pohjainen, harvemmin käytännössä tietokantaa muokataan suoraan MySQL-komennoilla. Tyypillisesti sivuston tietokantarakennetta muokataan suoraan sivuston pohjajärjestelmän tarjoaman käyttöliittymän kautta, mutta myös koodillinen muokkaaminen voi tulla kyseeseen tietyissä tapauksissa. Tyypillisiä muokkauksia, joita tehdään sivuston käyttöliittymän kautta, on esimerkiksi uusien sisältötyyppien (uutiset, blogi, perussivu) tai luokittelutermin (tuotteet, tuotekategoriat) luominen. Tällöin sisällönhallintajärjestelmä toki luo uudet tietokantataulut MySQL-komennoilla, mutta nämä on piilotettu varsin syväälle sisällönhallintajärjestelmään. Samanlaisia muutoksia saa automatisoitua myös pohjajärjestelmän koodikielellä, ja tätä käytetäänkin hyväksi opinnäytetyön käytännön osuudessa myöhemmin.

### 3 AVOIN VAI SULJETTU LÄHDEKODI

Avoimen lähdekoodin järjestelmien nousu suljettujen järjestelmien rinnalle on ollut viime vuosina varsin puhuttu aihe ohjelmistokehityksessä, mutta erityisesti internetin sisällönhallintajärjestelmien kehityksessä. Molempien tyyppisissä järjestelmissä on omat hyvät ja huonot puolensa, eikä edut välttämättä ole kovinkaan selkeitä projektin alussa järjestelmää valitessa. Mietinnän arvoisia seikkoja on ainakin lähdekoodin omistajuus, järjestelmän laajennettavuus ja ylläpito, työvoiman saatavuus ja osaavuus, sekä lisenssikustannukset. (Tolvanen. 22.2.2010.)

Avoimen lähdekoodin järjestelmän selkeänä etuna voidaan pitää, että lähdekoodi jää lähtökohtaisesti täysin asiakkaan omistukseen. Tällöin asiakkaalla on täysi hallinta projektiin, ainakin teoriatasolla. Kun asiakas omistaa sivuston lähdekoodin (ja muut sivuston tiedoston, kuten kuvat), on toimittajan tai muun yhteistyökumppanin vaihto tarvittaessa lisenssien puolesta yksinkertaista. Asiakas voi sopimuksen puitteissa kerätä lähdekoodinsa ja muut tiedostonsa ja suunnata seuraavalla tekijälle, jos nykyisen kanssa ei jostain syystä yhteistyö toimi. Käytännössä toki tilanne voi olla eri, varsinkin jos järjestelmä on laaja ja monimutkainen. Tällöin uudella toimittajalla voi olla hyvinkin suuri kynnyks oppia järjestelmän rakenteet ja päästä järjestelmän kehittämiseen sisälle. Käytännössä siis avoimen lähdekoodin järjestelmälläkin voi hyvin päätyä toimittajalukkaan, mutta suljetulla järjestelmällä toimittajalukkoa voi pitää varmana.

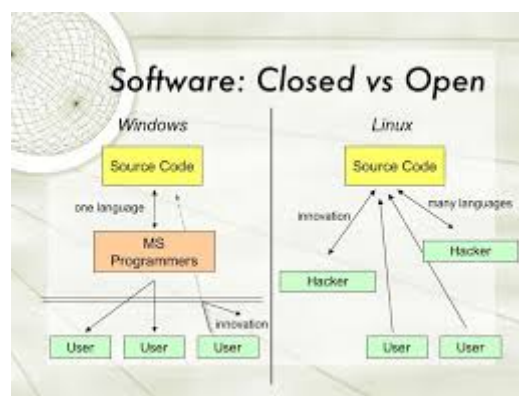
Avoin	Suljettu
<ul style="list-style-type: none"> <li>• Yhteisöllisyys</li> <li>• Koodi asiakkaan omistukseen</li> <li>• Ilmainen</li> <li>• Lisäosien ja laajennuksien saatavuus</li> </ul>	<ul style="list-style-type: none"> <li>• Tekninen tuki</li> <li>• Luotettavuus</li> <li>• Helppokäyttöisyys</li> <li>• Lisäosien ja laajennuksien laatu</li> </ul>

Kuva 3. Avoimen ja suljetun lähdekoodin vahvoja puolia.

Toinen avoimen lähdekoodin selkeä etu on usein lisäosien ja laajennusten helppo saatavuus. Toki hyvin hoidetussa suljetussa järjestelmässä tilanne voi olla täysin sama, mutta varsinkin avoimen lähdekoodin järjestelmiin löytyy tuhansittain erilaisia laajennuksia. Useasti avoimen lähdekoodin järjestelmien ympärille kehittyy yhteisö, joka voi olla hyvinkin aktiivinen tuottamaan ominaisuuksia järjestelmään. Tämä toimii samalla myös huonona puolena avoimen lähdekoodin järjestelmille, sillä hyvien ja huonojen laajennusten erottaminen voi olla hyvinkin työläs prosessi.

Samaan yhteisöllisyyteen liittyy myös työvoiman saatavuus ja osaavuus. Avoimen lähdekoodin järjestelmiä on helppo testailta ja "harrastaa", jolloin niihin on yleensä suljettuja järjestelmiä helpompi rekrytoida uusia perustason osaajia. Jos liki jokainen web-kehittäjä on ainakin muutaman kerran asentanut ja testannut WordPress-sivustoa, on WordPress-kehittäjän löytäminen suhteellisen yksinkertaista. Tämäkin etu tosin voi kääntyä avoimia järjestelmiä vastaan, jos järjestelmien kehittäjien osaaminen jää pintapuoliseksi. Microsoft-sertifioidun SharePoint-kehittäjän löytäminen voi olla hankalampi juttu kuin WordPress-kehittäjän, mutta tämän ammattitaito voikin helposti olla aivan toisella tasolla.

Suljetussa ympäristössä ohjelmistokehittäjät ovat luoneet ympäristön, jossa käyttäjät työskentelevät. Avoimen lähdekoodin ympäristössä käyttäjät työskentelevät suoraan lähdekoodin parissa. Kuva 4 toimii esimerkkinä, miten käyttöjärjestelmä puolella avoin ja suljettu järjestelmä toimivat.



Kuva 4. Suljettu vastaan avoin lähdekoodi (Open Source Development (OSS) under Eclipse Public License (EPL)).

Lisenssikustannukset, tai oikeastaan niiden puuttuminen, on ainakin jollain tasolla positiivinen puoli avoimen lähdekoodin järjestelmissä. Loppupeleissä kuitenkin tämä on monesti varsin pieni asia ja summa, verrattuna koko projektin hintaan. Jos järjestelmän muut ominaisuudet ovat huippuluokkaa, muutaman kymmenen tuhannen lisenssikuluja maksaa mielellään, ja ne säästyykin äkkiä ylläpito- tai kehityskuluissa takaisin. Monessa tilanteessa avoimen lähdekoodin järjestelmässä joutuu nimenomaan ”maksamaan ylimääräistä” muista ominaisuuksista, kun kehitys- tai ylläpitotoimet vievät enemmän aikaa ja sitä kautta rahaa.

Webbisivuston taustajärjestelmän valinta on aina monimutkainen prosessi, ja lähdekoodin avoimuus onkin vain osa isompaa palapeliä. Perttu Tolvanen summaakin artikkelissaan oivasti: ”Voisikin sanoa, että moni hyöty on kaksiteräinen miekka ja riippuu täysin omasta organisaatiosta ja sovelluskohteesta, että onko asia omalta kannalta hyöty vai heikkous.” (Tolvanen. 22.2.2010.)

## 4 DRUPAL

Drupal on yksi ”kolmesta suuresta sisällönhallintajärjestelmästä”, kahden muun ollessa WordPress ja Joomla (IT-parkki 2016). Kun esimerkiksi WordPress on lähtöisin blogialustasta ja sen vuoksi erikoistunut nimenomaan blogityyppisiin sivustoihin, Drupal soveltuu parhaimmin raskaisiin ja monimutkaisiin verkkosivustoihin. Tämä toimii samalla Drupalin vahvuutena, mutta myös heikkoutena. Drupal on kankea käytettävä loppukäyttäjälle ja sen oppiminen vaatii huomattavan määrän työtunteja. Osaavan käyttäjän käsi-ssä Drupal on tehokkaimpia sisällönhallintajärjestelmiä sopivissa sovelluksissa.

### 4.1 Drupalin historia

Drupalin historia alkaa vuodesta 2000, kun hollantilainen tietojenkäsittelytieteen opiskelija Dries Buytaert tarvitsi paikan, jossa keskustella ystäviensä kanssa. Niinpä he vuoden 2000 aikana perustivat uutissivuston keskinäiseen käyttöön. Driesin valmistuttua Antwerpenin yliopistosta kaverukset totesivat tarvitsevansa sivustoa internetin yli, ja tammi-kuussa 2001 Dries päätti julkaista sivuston lähdekoodin, merkiten Drupalin 1.0-version julkaisua. (Drupal.org. Our history.)

### 4.2 Drupalin ydin

Drupalin vakiojulkaisua kutsutaan Drupalin ytimeksi (englanniksi Drupal Core), joka sisältää yleiset ominaisuudet, jota sisällönhallintajärjestelmältä odotetaan. Drupalin ydin yleensä päivittyy muutaman kuukauden välein, sekä suurempia päivityksiä muutamien vuosien välein. Päivityksessä yleensä otetaan käyttäjien tekemiä moduuleja kiinteäksi osaksi Drupalin ydintä, sekä sitä kehitetään eteenpäin Drupal-organisaation toimesta.

### 4.3 Drupal-moduulit

Drupalin ydinon modulaarinen järjestelmä, tarkoittaen että se on suunniteltu toimimaan kolmannen osapuolen moduulien ja teemojen kanssa, jotka voivat laajentaa tai jopa korvata Drupalin oletuskäyttäytymistä muuttamatta Drupalin ytimen koodia. Drupal pitää yti-



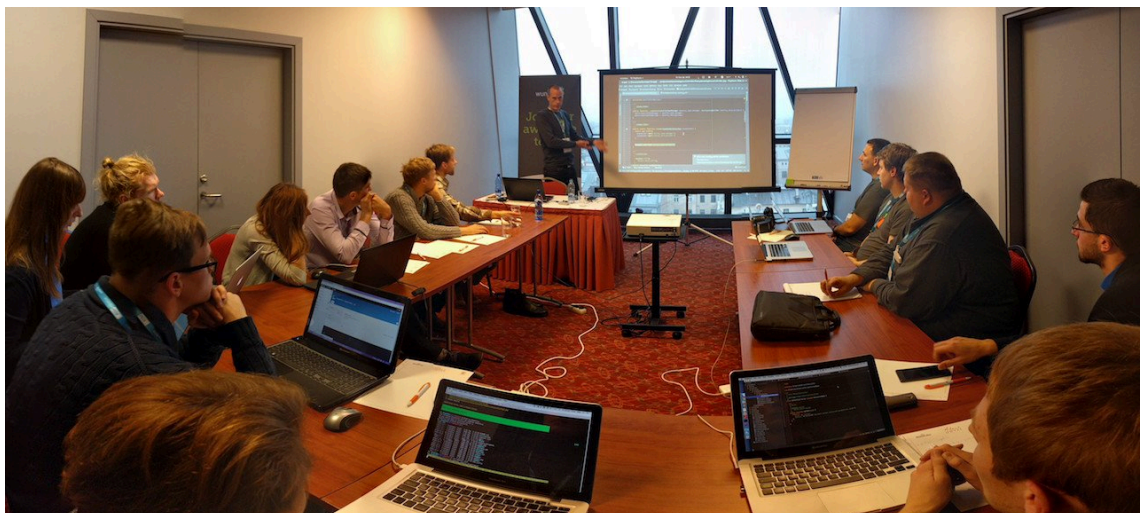
men tiedostot erillään lisätyistä moduuleista ja teemoista, tällä keinolla järjestelmänhallitsija pystyy päivittämään Drupalin ytimen uuteen versioon korvaamatta tehtyjä muutoksia.

#### 4.4 Drupalin yhteisöllisyys

Avoimen lähdekoodin järjestelmänä Drupalin ympärille on muodostunut aktiivinen yhteisö, joka yhdessä kehittää Drupalia. Toukokuussa 2017 Drupal yhteisössä oli 1,3 miljoonaa jäsentä, joista yli 100 000 käyttäjää osallistuu aktiivisesti. Tämä on johtanut yli 37 000 ilmaiseen moduuliin, joilla pystyy laajentamaan ja kustomoimaan Drupalin toiminnallisuutta. (Drupal.org. Homepage)

#### 4.5 Drupal tapahtumat

Avoimen lähdekoodin järjestelmien yhteisöllisyys näkyy parhaiten tapahtumina, joita Drupal-piireissä kutsutaan pienemmiksi DrupalCampeiksi tai isommiksi DrupalConeiksi. Nimestä riippumatta tapahtumien ideana on kerätä Drupal-ammattilaisia ja -harrastelijoita yhteen verkostoitumisen ja tiedonjakamisen merkeissä. Tapahtumat koostuvat esimerkiksi seminaari-tyyppisistä esityksistä, sekä koodisprinteistä. Kuvista Kuva 5 ja Kuva 6 huomaa hyvin seminaarin ja koodisprintin hengen eron.



Kuva 5. Esimerkkikuva seminaarista DrupalCamp Balticsista vuodelta 2016. (Twitter.com. DrupalCampBaltics seminar 2016.)



Kuva 6. Esimerkkikuva koodisprintistä DrupalCamp Balticsista vuodelta 2015. (Twitter.com. Lauri Eskola: DrupalCampBaltics 2015 sprint room.)

Seminaareissa pidetään luentoja erilaisista tavoista käyttää ja kehittää Drupalia. Koodisprinteissä puolestaan keskitytään nimensä mukaisesti koodaamaan. Näissä on yleensä jokin teema, minkä ympärille sprintti kietoutuu, ja osallistujat sitten kehittävät ja korjailevat joko Drupalin ydintä tai muita Drupalin moduuleja. Esimerkkejä koodisprintin teemoista on muun muassa ”suomalaisten maksutapamoduulien kehitys uuteen Drupal-versioon”, ”uuden Drupal-version saattaminen julkaisukuntoon” tai ”Drupalin Facebook-integraation parantaminen”.

## 5 DRUPAL-ASENNUKSEN AUTOMATISOINTI

### 5.1 Tehtävänasettelu

Opinnäytetyön tutkimuskysymykseksi asetettiin ”voidaanko Drupal-asennusta automatisoida ja jos voidaan niin kuinka pitkälle”. Työ on tehty yhteistyössä Drontti Oy:n asiantuntijoiden kanssa ja tulee yrityksen käyttöön onnistuessaan hyvin. Drontti Oy on pieni turkulainen digitoimisto, joka erikoistuu teknisiin Drupal-sivustoihin. Tämä tulee ottaa huomioon työtä tehdessä, eli projektissa voidaan keskittyä keskisuuren Drupal-sivuston asennuksen automatisointiin. Lisäksi projektissa voidaan olettaa asiakkaan olevan lähikohtaisesti suomalainen ja tehdä sen mukaan muita oletuksia, jos ne selkeyttävät automatiikan toimia. Esimerkiksi siis hallintakieleksi voidaan valita oletuksena suomi, ja aikavyöhykkeeksi Europe/Helsinki, mutta automatiikan pitää kuitenkin tukea useampia kieliä ja aikavyöhykkeitä, jos niille myöhemmin tulee käyttöä.

### 5.2 Haasteet ja riskit

Ennen yhteistyön alkamista Drontti Oy:n asiantuntijoiden kanssa tämän opinnäytetyön tekijällä oli hyvin vähäisesti kokemusta sisällönhallintajärjestelmistä. Verkkosivujen toiminnasta tekijällä oli pinnallinen käsitys, eikä Drupalin käyttämiä koodikieliä tullut käytettyä opintojen aika juuri lainkaan. Tämä tulisi varmasti aiheuttamaan haasteita projektin kuluessa.

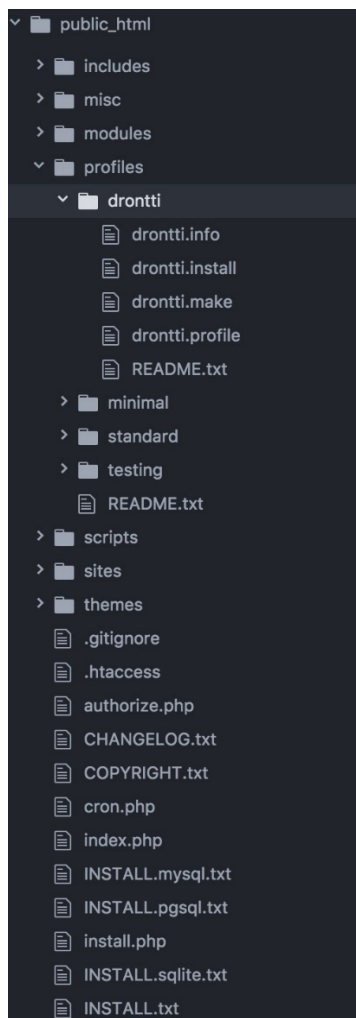
Projektin suurimpana riskinä on luonnollisesti, ettei Drupal-sivustoa saa automatisoitua tai että se ei jostain syystä ole järkevää tai kannattavaa. Drupalin pitäisi kuitenkin tukea asennuksen automatisointia asennusprofiilien muodossa varsin mallikkaasti, joten tämän riskin pitäisi olla varsin pieni ja projektista saada tulokseksi toimiva Drupal-asennusprofiili, jolla Drupalin asennuksen saa automatisoitua ja kustomoitua Drontti Oy:n tarpeisiin.

### 5.3 Drontti Oy

Drontti Oy on nuori turkulainen Drupal-alustalle omistautunut digitoimisto, jolle projekti tehdään. Drontti Oy keskittyy nimenomaan teknisesti haastavien Drupal-sivustojen rakentamiseen ja ylläpitoon. Drontti Oy:n asiakaskuntaan kuuluu erikokoiset yritykset ympäri Suomen. Yrityksen erikoisuutena voi pitää koodikouluja, joita järjestetään erityisesti lapsille ja nuorille suunnattuina. Yrityksen työntekijät ovat myös aktiivisia osallistumaan Drupalin ytimen kehitykseen ja pyrkivät osallistumaan Drupal-tapahtumiin aina kun se on mahdollista.

## 6 DRUPAL-ASENNUSPROFIILI

Drupal-asennusprofiili koostuu vähintään kahdesta tiedostosta: profiilinnimi.info sekä profiilinnimi.profile. Info-tiedosto sisältää perustiedot asennusprofiilista ja profile-tiedostolla hallitaan asennusprofiilin käyttäymistä. Tämän lisäksi sisällytin projektiin .install- ja .make-tiedostot, sekä README.txt:n. Install-tiedoston koodi suoritetaan kerran asennuksen yhteydessä, ja sen vuoksi onkin erinomainen paikka tietyille asennustoiminnoille. Make-tiedosto on Drushin käyttämä aputiedosto, jossa määritellään mitkä moduulit asennusprofiili asentaa, sekä näiden sijainnin tiedostojärjestelmässä. README.txt on yksinkertainen käyttöohje asennusprofiilille. Kuva 7 esittää asennusprofiilin tiedostojen sijainnin tyypillisessä Drupal-asennuksessa.



Kuva 7. Asennusprofiilin tiedostot Drupal-asennuksessa.

## 6.1 Tarvittavien moduulien asennus

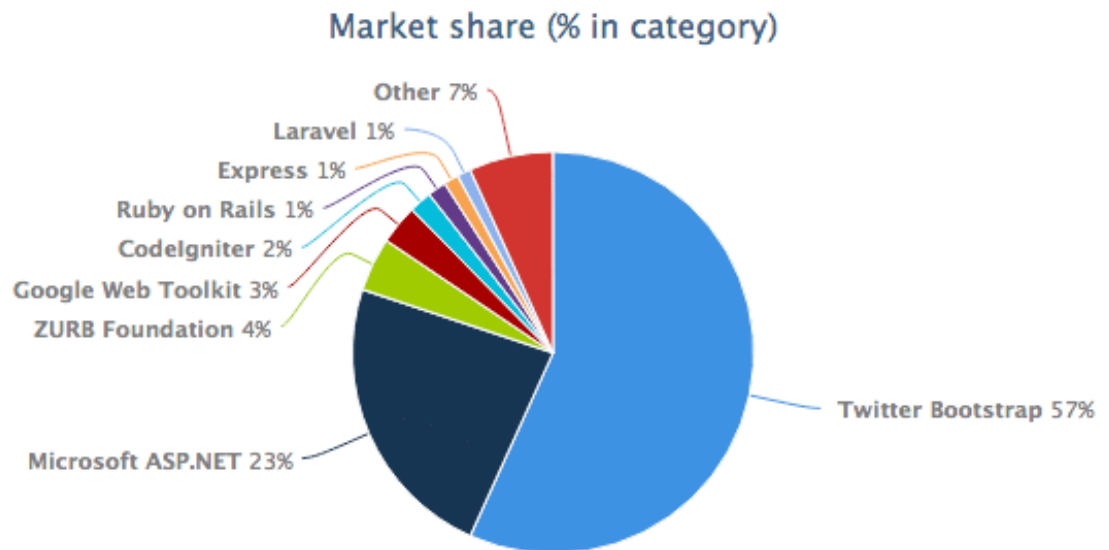
Tarvittavat lisämoduulit ladataan Drushin make-käskyllä, joka kohdistetaan asennusprofiilin make-tiedostoon, esimerkiksi `drush make profiles/drontti/drontti.make -y`. Tällöin Drush make -käsky käy läpi drontti.make-tiedoston, ja lataa moduulit tiedoston osoittamaan paikkaan. Suositeltu asennuspaikka yleisille moduuleille on polku `sites/all/modules/contrib`, jolloin sivustolle uniikit moduulit voidaan kirjoittaa polkuun `sites/all/modules/custom`. Kun moduulit asennetaan `sites/all`-kansioon, niin ne ovat kaikkien sivustojen käytössä, jos käytetään yhtä Drupal-asennusta monen sivuston ylläpitoon.

Drush make -käsky ei vielä asenna moduuleita, vaan pelkästään lataa ne drupal.org-sivustolta. Sama moduulilista on kuitenkin lisätty profiiliin riippuvuuksiin .info-tiedostosta löytyvällä `dependencies[]`-taulukolla, jolloin Drupal-asennus asentaa moduulit automaattisesti ennen muita asennusprofiilin operaatioita.

## 6.2 Pohjateeman ja oman teeman asennus

Asennusprofiili luo automaattisesti oman Bootstrap-viitekehikseen perustuvan alateeman, ja asentaa tarvittavat riippuvuudet. Asennus tapahtuu .install-tiedoston funktiossa `drontti_install_theme()`. Asennusprofiili lataa ensin Bootstrap-viitekehiksen Drushilla, ja sitten kopioi Bootstrap-kansiosta oman teeman aloituspakkauksen omaksi teemakseen.

Bootstrapin valinta oman teeman pohjaksi oli helppo päätös tehdä: 57% markkinaosuudellaan (Burge. 20.5.2016) **Error! Reference source not found.** Bootstrap on ylivoimaisesti suosituin verkkosivuston teemoitusviitekehys (front-end framework). Lisäksi Drupal tukee Bootstrap-viitekehystä mainiosti ja Drupal-Bootstrap-yhteistyö näyttää vain yleistyvän tulevaisuudessa.



Kuva 8. Webin suosituimmat viitekehykset.

Oman teeman luomisen jälkeen asennusprofiili luo teeman alle muutaman jokaisessa projektissa tarvittavan teemoitustiedoston: `front.less`, `footer.less`, `header.less` ja `content.less`, ja lisää näiden tiedostojen käyttöönoton (`import`) päätyylitiedostoon `style.less`. Tämän jälkeen asennusprofiili luo kansion teemassa tarvittaville Javascript-tiedostoille, ja tämän alle `scripts.js` -tiedoston sekalaisille tarvittaville Javascript-skripteille.

Lopuksi asennusprofiili muokkaa sivuston tietokantaa ja ottaa oman teeman käyttöön ja asettaa sen sivuston oletusteemaksi.

### 6.3 Käyttäjäasetukset

Yksi asennusprofiilin päämotivoijista oli sivuston testikäyttäjien standardointi. Kun asennusprofiili luo tarvittavat testikäyttäjät automaattisesti, ei kehittäjien tarvitsisi joka kerta miettiä, mikä testikäyttäjän sähköpostiosoite olikaan tai oliko ylläpitäjä-käyttäjän salasana kehitysvaiheessa `admin` vai `nimda`. Asennusprofiili käy käyttäjäasetukset läpi `.install`-tiedoston funktiossa `drontti_install_users()`.

Aluksi asennusprofiili luo administrator-roolin, jolle annetaan sivuston kaikki oikeudet. Käyttäjäroolit ovat Drupalissa käytössä oleva järjestelmä käyttäjien oikeuksien hallintaan. Moduulit voivat luoda erilaisia käyttöoikeuksia, sekä sääntöjä mitä näillä käyttöoi-

keuksilla saa sivustolla tehdä, esimerkiksi ”saa muokata perussivuja” tai ”saa siirtää tiedostoja sivustolle”. Tämän jälkeen Drupalin hallintapaneelissa voidaan antaa eri käyttäjäröoleille näitä käyttöoikeuksia, ja lopulta yksittäiselle käyttäjälle annetaan käyttäjäröoleja. Systeemi on paikoitellen varsin raskas käyttää, mutta se mahdollistaa hyvinkin tarkan ja uniikin oikeuksien hallinnan Drupal-sivustoilla.

Seuraavaksi asennusprofiili luo muutaman testiylläpitäjä-käyttäjän, ja antaa näille administrator-käyttäjäröolin. Lopuksi funktio luo vielä muutaman tavallisen testikäyttäjän erilaisiin sivustontestausoperaatioihin.

#### 6.4 Sisällönhallinta-asetukset

Asennusprofiili käy läpi sisällönhallinnan perusasetukset `.install-tiedoston drontti_install_content()` -funktiossa. Asennusprofiili luo ja asentaa ”Filteröity HTML”- ja ”Täysi HTML”-tekstimuodot. Eri tiedostomuodoilla voidaan hallita, minkä tyyppistä sisältöä eri käyttäjät voivat sivustolle luoda. ”Filteröity HTML” -tekstimuodon voi antaa kenen vaan sivuston käyttäjän käyttöön, kun taas ”Täysi HTML” -tekstimuoto tulisi antaa vain luotetuille käyttäjille (esimerkiksi ylläpitäjille).

Tämän jälkeen asennusprofiili luo sisältötyypit Perussivulle ja Etusivulle. Sisältötyypit ovat Drupalin peruspalikoita, joihin tallennetaan tietoa, ja joita näytetään joko suoraan sivuina tai sivun osina. Tässä asennuksessa halusimme luoda Perussivun ja Etusivun sisältötyypit erikseen, koska lähtökohtaisesti sivustojen etusivut ovat niin poikkeavia muista sivuista.

#### 6.5 Muut asetukset ja monikielisyysasetukset

Lopuksi asennusprofiili käy vielä läpi monikielisyysasetukset sekä muita sekalaisia asetuksia. `drontti_install_languages()` -funktiossa asennusprofiili asentaa suomen kielen, ja asettaa sen sivuston oletuskieleksi. Asennusprofiilia voisi kehittää eteenpäin tukemaan muitakin kieliä automaattisesti, mutta tässä vaiheessa sivustojen asiakaskunta näytti olevan vahvasti suomalaista, joten päädyimme tämän olevan paras ratkaisu tässä vaiheessa

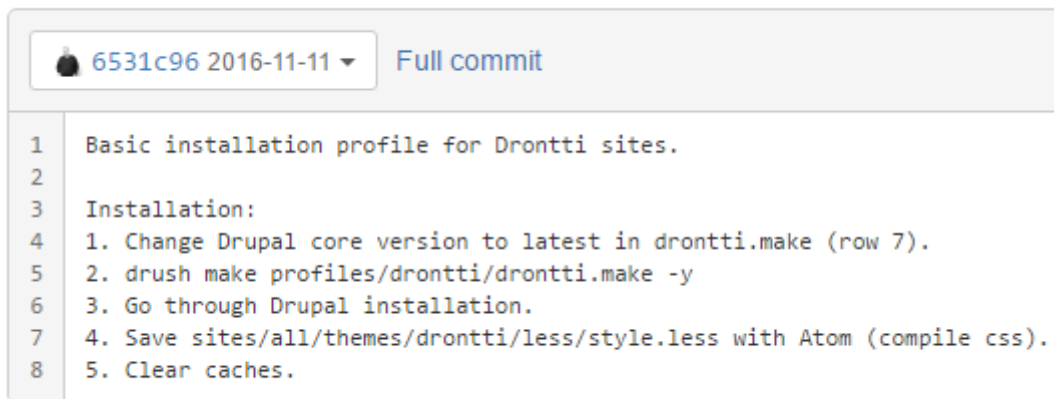
`drontti_install_other()` -funktiossa asennusprofiili käy läpi muut asetukset, jotka eivät ole sopineet muihin kategorioihin. Asennusprofiili poistaa muutaman Drupalin oletuslohkon



käytöstä, koska näille ei yleensä ole tarvetta. Seuraavaksi luodaan etusivu-sivu ja se asetetaan sivuston etusivuksi.

## 6.6 README.txt ja asennusprofiilin käyttö

Asennusprofiilin mukana tulevassa README.txt-tiedostossa on ohjeet asennusprofiilin käyttöön. Asennusprofiilia on hyvin suoraviivaista käyttää, mutta ihan kaikkea ei tämän projektin puitteissa pystynyt tai ollut järkevää automatisoida. Kuten kuvasta Kuva 9 huomataan, on asennusprofiilin käyttäminen hyvin yksinkertaista.

A screenshot of a commit message in a code editor. At the top, there is a commit hash '6531c96' and a date '2016-11-11' with a dropdown arrow, followed by a 'Full commit' button. Below this, the commit message is displayed in a monospaced font with line numbers on the left. The message content is: '1 Basic installation profile for Drontti sites.', '2', '3 Installation:', '4 1. Change Drupal core version to latest in drontti.make (row 7).', '5 2. drush make profiles/drontti/drontti.make -y', '6 3. Go through Drupal installation.', '7 4. Save sites/all/themes/drontti/less/style.less with Atom (compile css).', '8 5. Clear caches.'

Kuva 9. README.txt

Aluksi päivitetään uusin Drupalin ytimen versio drontti.make-tiedoston riville 7. Jos uusimman ytimen versio ei ole vielä tiedossa, löytyy se helpoiten drupal.org -sivustolta (Download & Extend. 28.9.2003.). Tämän jälkeen suoritetaan Drush Make -käsky asennusprofiilin make-tiedostoon. Esimerkiksi sivuston juurikansioista suoritettuna käsky on *drush make profiles/drontti/drontti.make -y*.

Tämän jälkeen käydään läpi Drupal-asennus verkkoselaimella. Asennus koostuu käytännössä muutamasta Seuraava-napin painalluksesta, koska kaikki asetukset tulevat automaattisesti asennusprofiilista.

Seuraavaksi pitää kääntää asennusprofiilin tyylitiedosto .less-tiedostosta verkkoselaimien ymmärtämäksi .css-tiedostoksi. Tämä käy kokemuksemme mukaan helpon asentamalla Atom-tekstieditoriin Less-kääntäjä, jolloin kääntäminen tapahtuu automaattisesti tiedostoa tallentaessa. Lopuksi tulee vielä tyhjentää välimuistit, joko ajamalla komentorivillä *drush cc all*, taikka sitten Drupalin hallintapaneelin kautta (Asetukset => Kehitys => Suorituskyky => Tyhjennä kaikki välimuistit).

## 7 YHTEENVETO

Drupal sisällönhallintajärjestelmässä on kaksi tapaa tuottaa uusi sivusto, manuaalinen ja automaattinen alustus. Molemmista on kuitenkin omat rajoitteensa, eivätkä ne aina toimi tehokkaasti. Sivuston alustamista halutaan nopeuttaa automatisoimalla yksinkertaisia ja projektista toiseen toistuvia työvaiheita.

Opinnäytetyössä on tutkittu eri tapoja nopeuttaa sivuston luontiprosessia ja työn aikana on tuotettu menetelmä, jolla uuden sivuston luomista pystytään nopeuttamaan olennaisesti.

Projektin tavoitteeseen päästiin, sekä tuotettua menetelmää on sen valmistumisen jälkeen toistuvasti käytetty uusien sivujen luontiin. Esimerkiksi [www.ordoaboensis.fi](http://www.ordoaboensis.fi) ja [www.grail.fi](http://www.grail.fi) -sivustot on saatettu alkuun opinnäytetyössä tuotetun Drupal-asennusprofiilin pohjalta. Asennusprofiili toimii mainiosti ja luo yrityksen web-kehittäjille hyvän yhteisen työkalun, jolla luoda nopeasti ja tehokkaasti sivustorunkoja, joissa on samat moduulit asennettuna ja asetukset tallennettuna. Tämä tehostaa paitsi sivuston ensimmäistä pystytyskertaa, myös yrityksen sivustojen ylläpitoa pitkässä juoksussa, kun kehittäjien ei tarvitse miettiä ja ihmetellä, että puuttuuko sivustolta jokin perusmoduuli, tai onko jokin perusasennuksen asetus unohtunut asettaa.

Menetelmää voi vielä jatkokehittää muutamalla alueella, kuten antamalla käyttäjälle muutaman vaihtoehdon tietyissä vaiheissa, sekä nopeuttamalla prosessia edelleen. Asennusprofiili voisi esimerkiksi ladata isomman joukon moduuleita valmiiksi, mutta jättää ottamatta ne käyttöön oletuksena. Näin sivuston laajentaminen olisi entistä nopeampaa, mutta turhat moduulit eivät rasittaisi sivuston vauhtia, jollei niitä erikseen pistä päälle. Asennusprofiiliin haluttavien moduulien valinta on kuitenkin aikamoista viivan piirtämistä veteen, ja tämän työn puitteissa valittu joukko kuvastaa varsin hyvin nykyhetkeä Drupal-moduuleista, jotka eivät ole osa Drupal-ydintä, mutta jotka liki jokaisella sivustolla silti on käytössä.

Yksi suunta mihin asennusprofiilia voisi myös jatkokehittää, on verkkokauppa-moduulien integrointi asennusprofiiliin. Drupal Commerce -verkkokauppa-alusta on hyvin suosittu, ja todella moni verkkokauppa pyörii sen pohjalta, niin kotimaassa kuin ulkomaillakin. Tätä ei varmaankaan kannattaisi lähteä sekoittamaan jokaisen verkkosivuston pohjalla pyöri-

vään asennusprofiiliin, mutta varsin helposti voisi ylläpitää kahtakin asennusprofiilia rinnakkain: toista tavallisille verkkosivustoille ja toista verkkokauppasivustoille. Kuten tavallinen Drupal-asennus, myös Drupal Commerce -asennus on varsin kankea ja moniosainen, ja tätä varmasti voisikin nopeuttaa. Lisäksi tätä asennusprofiilia tehdessä voisi tehdä oletuksen, että verkkokauppaa tehdään suomalaiselle asiakkaalle, ja tällöin voisi esimerkiksi asentaa suomalaisten pankkien maksumoduulit valmiiksi ja asettaa verot ja toimitussysteemit suomalaiseen verkkokauppaan sopivaksi.

## LÄHTEET

IT-parkki. 2016. Suosituimmat julkaisujärjestelmät heinäkuussa 2016. Viitattu 12.11.2016 <http://www.it-parkki.fi/suosituimmat-julkaisujarjestelmat-cms-072016-mika-paras/>

Perttu Tolvanen. 3.11.2009. Käsitesekamelskaa: julkaisujärjestelmä, CMS, portaali, sisällönhallintajärjestelmä. Viitattu 12.11.2016 <http://vierityspalkki.fi/2009/11/03/kasitesekamelskaa-julkaisujarjestelma-cms-portaali-sisallontalintajarjestelma/>

Webhotellivertailu2. Ilmaisen julkaisujärjestelmän valinta (Joomla, Drupal, WordPress, OS-Commerce, Zen Cart). Viitattu 12.11.2016 <http://www.webhotellivertailu2.fi/ilmaisen-julkaisujarjestelman-valinta-joomla-drupal-wordpress-os-commerce-zen-cart/>

Perttu Tolvanen. 25.2.2010. Avoimen lähdekoodin top-10 julkaisujärjestelmät Suomessa 2010. Viitattu 12.11.2016 <http://vierityspalkki.fi/2010/02/25/avoimen-lhdekoodin-top-10-julkaisujarjestelmt-suomessa-2010/>

Perttu Tolvanen. 22.2.2010. Avoimen lähdekoodin julkaisujärjestelmien vahvuudet ja heikkoudet. Viitattu 13.11.2016 <http://vierityspalkki.fi/2010/02/22/avoimen-lahdekoodin-julkaisujarjestelmien-vahvuudet-ja-heikkoudet/>

International Journal of Advanced Research (2015), Volume 3, Issue 12, 677 – 686. Open Source Development (OSS) under Eclipse Public License (EPL)

Drupal.org. About. Viitattu 13.11.2016 <https://www.drupal.org/about>

Drupal.org. Drupal Case Studies. Viitattu 13.11.2016 <https://www.drupal.org/case-studies>

Wikipedia. Drupal History. Viitattu 13.11.2016 <https://en.wikipedia.org/wiki/Drupal#History>

Drupal.org. Our history. Viitattu 13.11.2016 <https://www.drupal.org/about/history>

Drupal.org. Homepage. Viitattu 23.4.2017. <https://www.drupal.org/>

Twitter.com. DrupalCampBaltics seminar 2016. Viitattu 23.4.2017. <https://twitter.com/DCBaltics/status/793780908180537344>

Twitter.com. Lauri Eskola: DrupalCampBaltics 2015 sprint room. Viitattu 23.4.2017. <https://twitter.com/laurii1/status/637556649121722368>