



# TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto  
School of Business and Management  
Degree Program in Computer Science

Isto Sipilä

## **Maintenance of Online Services in Small Organizations: Case Lappeenrannan Ruokapiiri**

Diplomityö

2016

53 sivua, 3 kuvaa, 3 taulukkoa

Työn tarkastajat:      Professori                      Kari Smolander  
   Tekniikan tohtori              Erno Vanhala

Hakusanat: pieni organisaatio, verkkopalvelu, verkkokauppa, ylläpito, huolto, sisällönhallintajärjestelmä, tapaustutkimus

Keywords: small organization, online service, e-commerce, upkeep, maintenance, content management system, case study

Työn tavoitteena oli tutkia pienten organisaatioiden ongelmia verkkopalveluiden ylläpidossa ja löytää teknisiä ja muita ominaisuuksia, jotka voisivat helpottaa pieniä organisaatioita vastaamaan tunnistettuihin haasteisiin. Lappeenrannan Ruokapiirin uuden tilausjärjestelmän hankinnasta ja ylläpidosta tehtiin tapaustutkimus. Uusi järjestelmä rakennettiin käyttäen vakiintunutta sisällönhallintajärjestelmää ja ketteriä ohjelmistokehityksen menetelmiä tiiviissä yhteistyössä järjestelmän tilaajan kanssa. Valitulla ratkaisulla onnistuttiin välttämään edellisen järjestelmän, joka oli vaikea päivittää vastaamaan nykyisiä vaatimuksia, ongelmat. Merkittävin kohdattu ongelma projektin aikana oli omien toteutuksien ja alustan välisen yhteensopivuuden ylläpito alustapäivitysten yhteydessä. Tämän ongelman helpottamiseksi suositellaan, aina kun se on mahdollista, yhteisön tukemien laajennuksien käyttöä tai omien laajennuksien julkaisemista kehittäjäyhteisössä.

## **ABSTRACT**

Lappeenranta University of Technology  
School of Business and Management  
Degree Program in Computer Science

Isto Sipilä

### **Maintenance of Online Services in Small Organizations: Case Lappeenrannan Ruokapiiri**

Master's Thesis

2016

53 pages, 3 figures, 3 tables

Examiners: Professor Kari Smolander  
D.Sc. (Tech.) Erno Vanhala

Keywords: small organization, online service, e-commerce, upkeep, maintenance, content management system, case study

The objective of this thesis was to study problems small organizations encounter during maintenance of an online service and to find technical and other features, which may help small organizations to deal with identified difficulties. Case study was conducted on acquisition and maintenance of Lappeenrannan Ruokapiiri's new ordering management system. The new system was built on well-established content management system by using agile software development methods including close collaboration with the product owner. Selected solution proved to be successful in avoiding difficulties encountered during the use of the previous system, which was difficult to upgrade to fulfill current requirements. During the project, the most significant maintenance challenge encountered was to maintain compatibility between custom made implementations and the platform during platform updates. To alleviate this problem, the use of community supported extensions or publishing custom made extensions in the developer community, whenever possible, is suggested.

## **ACKNOWLEDGEMENTS**

This thesis was done for the LUT School of Business and Management at Lappeenranta University of Technology.

I want to thank my supervisors Kari Smolander and Erno Vanhala for patience with my eccentric work schedule. Another thanks to Erno for suggesting me this thesis topic.

I want to thank my friends and family for your support. Special thanks to Sanna for arranging me this opportunity to work with Lappeenrannan Ruokapiiri and to Veli-Ensio for inspiring and helping me during the project.

For the Lappeenrannan Ruokapiiri's board, I enjoyed working with you, thank you very much!

Finally, I want to express my gratitude for LUT for these memorable years.

At Lappeenranta 20.12.2016

Isto Sipilä

# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
1.1	BACKGROUND OF THE STUDY .....	4
1.2	OBJECTIVES AND RESEARCH METHODOLOGY .....	5
1.3	ORGANIZATION OF THE STUDY .....	6
<b>2</b>	<b>E-COMMERCE AND CONTENT MANAGEMENT SYSTEMS .....</b>	<b>7</b>
2.1	E-COMMERCE.....	7
2.2	CONTENT MANAGEMENT SYSTEMS .....	9
<b>3</b>	<b>CASE: LAPPEENRANNAN RUOKAPIIRI .....</b>	<b>13</b>
3.1	LAPPEENRANNAN RUOKAPIIRI'S OPERATING MODEL.....	13
3.2	THE OLD SYSTEM .....	15
3.2.1	<i>Technical Performance</i> .....	16
3.2.2	<i>Usability</i> .....	16
3.2.3	<i>Missing Features</i> .....	16
3.2.4	<i>Upkeep and Maintenance</i> .....	17
3.3	PRELIMINARY WORK .....	18
3.3.1	<i>Preliminary Study</i> .....	18
3.3.2	<i>Initial Requirement Specification</i> .....	19
3.3.3	<i>Resources</i> .....	21
3.4	CONSIDERED IMPLEMENTATION POSSIBILITIES FOR THE NEW SYSTEM .....	22
3.4.1	<i>Building Own System from Scratch</i> .....	22
3.4.2	<i>Building Own System Using a CMS</i> .....	22
3.4.3	<i>Using an E-commerce Website Service</i> .....	24
3.5	SELECTED SOLUTION .....	25
3.5.1	<i>WordPress</i> .....	25
3.5.2	<i>WooCommerce</i> .....	27
3.5.3	<i>Plugins</i> .....	27
3.6	COURSE OF THE PROJECT .....	29
3.6.1	<i>Development Process</i> .....	30
3.6.2	<i>Initial Development</i> .....	31

3.6.3	<i>Continuous Development and Maintenance</i> .....	34
<b>4</b>	<b>DISCUSSION</b> .....	<b>39</b>
4.1	SELECTED SOLUTION .....	39
4.1.1	<i>Compatibility Issues Caused by Updates</i> .....	40
4.1.2	<i>Custom Implementations and Long-term Maintenance</i> .....	41
4.2	DEVELOPMENT PROCESS .....	41
<b>5</b>	<b>CONCLUSIONS</b> .....	<b>43</b>
	<b>REFERENCES</b> .....	<b>45</b>

## **SYMBOLS AND ABBREVIATIONS**

AJAX	Asynchronous JavaScript and XML
CSS	Cascading style sheets, a style sheet language
CMS	Content management system
GUI	Graphical user interface
HTML	Hypertext Markup Language
JS	JavaScript, a programming language
LUT	Lappeenranta University of Technology
OSL	Open Software License
PEAR	PHP extension and application repository
PHP	PHP: Hypertext preprocessor, server-side scripting language
PLT	Page load time
PS	Page size
PSP	Payment service provider
RDBMS	Relational database management system
UI	User interface

# 1 INTRODUCTION

Small organizations are using online services for a wide range of purposes, including e-commerce and marketing. With current development tools, platforms and services available it is not necessary to build an online service from the ground up. By using a content management system (CMS), a small business can build an online service without professional programmers and with a limited budget (Mirdha, et al., 2014). Various companies provide services for building online service for a customer by using one of the available CMSs or by providing their own tools, which makes it possible for a customer to create a web site themselves without any need for programming experience.

In this thesis, the focus is on online services built by using a CMS and problems related to upkeep and maintenance of such services in small organizations. Case study is conducted on one particular case where a small organization acquired a new system for its e-commerce and ordering management purposes. Within the framework of this study, the new system was developed, deployed and maintained. The objective of this study is, on the basis of the case project, to find technical and other features which may help a small organization to deal with possible difficulties that may emerge during a process of acquiring and maintaining an online service.

## 1.1 Background of the Study

During the second half of the year 2013, a need for a new order management system was emerging in Lappeenranta Ruokapiiri, a small association operating in Lappeenranta, Finland. The association's main goal is to help local food producers and interested customers to find each other. They provide a service, which works as a bridge between the local producers and their members. Members of the association can order products directly from the local producers without an intermediary taking its own share of the profit. The system providing this service was reaching its limits because of the growth of the association. Its performance and features were insufficient and caused unnecessary workload to the associations' staff and the user experience of ordering products were found as cumbersome and old fashioned.

The board of Lappeenranta Ruokapiiri started to consider two possibilities: should they hire someone to upgrade the existing system or should they acquire a new one. One of the members of the board knew the university student, the author of this study, who was studying software engineering in Lappeenranta University of Technology (LUT) at the time and contacted him. This initiated the process of investigation of the possibilities and eventually deployment of the solution for the problem. During the process Erno Vanhala, member of the board of Lappeenranta Ruokapiiri and doctoral student in LUT at the time, suggested that the case has interesting possibilities for academic research and that led to the realization of this thesis work.

## **1.2 Objectives and Research Methodology**

In this thesis work, problems of small organizations running an online service are studied. The objective is to identify problems small organizations encounter during maintenance and upkeep of an online service and to find technical and other features, which may help small organizations to deal with those identified difficulties.

The base knowledge of the field is established by literature review which explores the earlier research made about e-commerce and CMSs for running online services. The literature review presents groundwork for the study by explaining the core concepts and setting the premise for the case study.

Answers to the introduced research question are searched by the means of case study. The knowledge collected over the time of the case project, which consists of the observations made by the study's author and the feedback collected from the other project stakeholders, is documented and discussed. For fulfilling the objective, conclusions from the made observations are drawn.

### **1.3 Organization of the Study**

The thesis begins with the literature review in the chapter 2. The bulk of the work, the case project, is presented in the chapter 3 that includes the description of the operating model of Lappeenrannan Ruokapiiri, the old system and its flaws, preliminary work done before the execution of the project, considered and selected solutions and the course of the project. The chapter 4 presents the discussion of the findings made during the project and includes analysis of the selected solutions. In the chapter 5 the conclusions are drawn and it ends the work.

## **2 E-COMMERCE AND CONTENT MANAGEMENT SYSTEMS**

This chapter serves as an introduction to the topic and provides the required background information for the subsequent discussion. The review of the literature is conducted and the results are presented in this chapter including commentary when considered appropriate for the sake of comprehensibility of the issue. The first subchapter describes the concept of e-commerce, presents different options available for small organization to build an e-commerce service and presents experiences of using and implementing e-commerce services found in literature. In the second subchapter, the concept of content management system is explained and the research literature of different available CMSs is presented.

### **2.1 E-commerce**

In the business to consumer context, which is the relevant context for this study, (Grandon & Pearson, 2004) defines e-commerce as:

*“The process of buying and selling products or services using electronic data transmission via the Internet and the www.”*

This process is usually implemented as an online service which provides an e-commerce website. Typical e-commerce site follows so called on-line/transactional web site design strategy described in (Wen, et al., 2001). In the on-line/transactional strategy, an electronic catalog of products is presented and customer can browse the catalog and order products by using a web browser. The whole transaction is done online, including payment.

Purchases from e-commerce websites can be paid via bank transfer or more commonly using an online payment option provided by a payment service provider (PSP) or payment gateway service connected to the e-commerce website. These services provide an option to use various electronic payment methods such as credit card and real-time bank transfer made possible by online banking systems. A merchant has an account in a payment service and as described in (Berger, et al., 2002), when a customer makes a purchase from an online store, a connection is established between the merchant’s system and the third-party payment gateway. The payment information is uploaded and the gateway transmits this information

to a processing authority for authorization of the transaction. This authority can be, for example, a bank which online banking services a customer uses. If the payment is authorized, and not denied, the gateway notifies the merchant in real-time of completion of the sale.

According to (Hong & Zhu, 2006), the adoption of e-commerce requires a substantial degree of technical and organizational competence from a company. It was found that the size of the company was negatively related to e-commerce adoption and e-commerce migration. According to the study, large companies tended to be slow to migrate to the internet platform. Based on this finding it may be suggested that for a small and agile organization the adoption of e-commerce could be more approachable option than for a big organization.

In (Pelkkikangas, 2010), the results of the survey carried on Finnish e-commerce administrators to learn about the state of Finnish e-commerce are presented. The first goal was to find which e-commerce platforms are used and the second goal was to explore what administrators of the e-commerce sites would want from the used platforms in the future. The survey was targeted for two different groups of stakeholders: people managing the content of the web stores and people working on the maintenance of the sites. Regardless of the used platform, the adding of new products was considered too time consuming and it was reported to require too much work. Managing products and discount campaigns were also reported to be difficult. People maintaining e-commerce services reported that the most difficult tasks were appearance modification, installation of platform updates, transferring data from other systems, integration to other systems and installation of extensions. Answers divided evenly on all these tasks. The lack of technical knowledge was reported to be the most important cause for the difficulties. Many maintenance tasks were reported to need programming skills and only a part of the people who participated in the survey reported to possess those skills. When using self-made platforms, the implementation of the upgrades was considered to be expensive and time consuming.

For a small organization, there exist several ways to implement an e-commerce service. One option is to build a service from scratch, which is the most expensive and demanding alternative, because it is time consuming and requires programming skills. Another option is to build on pre-existing CMS, which provides undeniable benefits described in the next

chapter. There exist open source CMSs that have been built for constructing any kind of website, such as WordPress (WordPress, 2015b), Joomla (Open Source Matters, 2015a) and Drupal (Drupal, 2015). In addition, there exists CMSs developed solely on e-commerce purposes, such as Magento (Magento, 2015a) and PrestaShop (PrestaShop, 2015a). Third option is to use services of the companies providing ready-made e-commerce solutions. When using this kind of service, the e-commerce website can be deployed without any programming skills and all management can be done through a web interface. There are several such services in Finland alone, such as MyCashFlow (Pulse247, 2015a), Valmiskauppa (Suomen Valmispalvelut, 2015) and Vilkas (Vilkas Group, 2015).

## **2.2 Content Management Systems**

According to (Boiko, 2002), content is information with a context. In the context of CMSs, the information is surrounded by metadata which is a simplified version of the context and provides meaning of the information. Computers cannot manage information directly and can only handle data, so the problem is simplified in this way and in CMSs information is handled indirectly through metadata. This makes it possible to manage content with computer's data capabilities.

In (Benevolo & Negri, 2007), content management is defined as:

*“system of methods and techniques to automate the processes of content collection, management and publishing using information technologies”*

According to (Benevolo & Negri, 2007), the working of a CMS consists of these three phases, or systems, mentioned above in which the information goes through when a CMS handles it. In the collection system, information and its metadata is received and transformed to content components. In the management system, these content components are stored in a database and the system is responsible for organizing the content. This management system should provide tools for finding the answers to any questions about the content. Third system, the publishing system, extracts content from the database and creates publications of it. Templates define how content needs to be treated and are used for constructing publication pages.

Using a CMS for managing content of an online service provides various benefits. According to (Seadle, 2006):

*“A content management system (CMS) offers a way to manage large amounts of web-based information that escapes the burden of coding all of the information into each page in HTML by hand.”*

This means that when a user wants to add new content to a web site which is not using a CMS, the actual source code of the website has to be edited. For example, if a case is considered where an author wants to publish a news article on a news site not using CMS, the actual source file of the page, which forms the web page, has to be modified. This requires technical knowledge and, depending on the implementation of the website, programming skills. When using a CMS, an author can publish an article on a website by using, for example, a graphical user interface (GUI), therefore not needing any programming skills.

Because websites built using a CMS are template based, according to (Mirdha, et al., 2014), the content is separated from the presentation allowing the quick change of the appearance, layout and structure of the website. Because of this separation, the presentation can be changed without affecting the content. Therefore, in the example-case presented above an author of a news site can work with the content and a person responsible for maintenance and development of the website can work on the presentation without interfering author's work of producing content. This example can be extended to any type of website or online service, including e-commerce websites.

There exist several studies comparing different available CMSs. In study (Wiklund, 2015), the process of implementing e-commerce website with Drupal for a small company is discussed and can be seen as a good point of reference for the case project of this study. At first, in (Wiklund, 2015), CMSs developed for multi-purpose usage was reviewed including WordPress, Joomla and Drupal, and after that, CMSs focused solely on e-commerce purposes, such as Magento and Ubercart. Magento, Drupal Commerce and Ubercart made it to the final ground. Magento was considered extremely strong e-commerce platform, but its features were found to be limited if the website has to offer something else besides a web

shop. There exist two different editions of Magento: commercial Enterprise Edition of Magento and free Community Edition with limited feature set. Because the feature set of the free Community Edition did not compare well to the feature sets of the other free solutions, it was discarded. According to the author, Ubercart provides a solution that will work straight out of the box and Drupal Commerce will require significantly more initial configuration than other options. Because Drupal Commerce integrates more heavily with Drupal and popular Drupal modules, which would make it lighter solution for sites already using these modules, it was selected as the best solution in the case.

In paper (Mirdha, et al., 2014), seven open source CMSs are compared on the dimensions of system requirements, performance, security, built-in features, support and management. The seven CMSs compared, which were Alfresco, Typo3, DotNetNuke, Drupal, Joomla, WordPress, and Plone, were chosen to the subjects based on their e-commerce market share, Google page rank, brand strength and rate of adoption. A medium scale business website was designed using each of the CMSs and were evaluated against set criteria. The conclusions were that Joomla offered most flexible system requirements and Joomla, Drupal and WordPress are on the same line when measured security features. Typo3 was considered best when measuring performance and DotNetNuke offers most built-in features. Joomla is considered to be the best on managing assets and WordPress's feature set was considered to be the worst in this category. It is concluded that all compared CMSs are easy to use.

The comparison of three CMSs: Joomla, Drupal and WordPress, is presented in (Patel, et al., 2011a). CMSs are evaluated based on their page performance. Similar website was built on each of compared CMSs. Websites were hosted on local and live server and, for example, measurements of page load times (PLT), page sizes (PS), number of requests send to server and number of CSS and JS files were taken. When running websites on local server Drupal was considered the best performer when running informative site because of its small PS and short load times. When goal is to build intranet site with multiple objects and fast response times Joomla was consider the best because it was measured to handle load better than Drupal. In the local server test case, WordPress measured the worst of the three in most of measurements. When websites were run on a live server, which is considered the more important test case by the authors, the results changed. In the live server test, WordPress was

measured to be the best performer on the most of the areas and is considered the best option by the authors.

In study (Patel, et al., 2011b) Joomla, Drupal and WordPress are compared and the goal of the study was to find out which one is preferable for which purpose. The CMSs are compared by their features, popularity and average budget used when building websites with each one. The figures are acquired from different surveys. In the conclusion, it is stated that Joomla, Drupal and WordPress are the most popular CMSs. Joomla is recommended for novices and it is considered cheaper than other options. Drupal is said to provide most facilities and flexibility. WordPress is said to have the highest documentation support.

It can be seen that studies presented above are focusing on differences in CMSs related to features affecting possibilities to use them in different use cases and variables such as popularity, cost and technical performance. However, there exist little discussion about maintenance and upkeep features of those CMSs. Within the scope of this study, the research focusing on upkeep and maintenance features of online services build on available CMSs were not found.

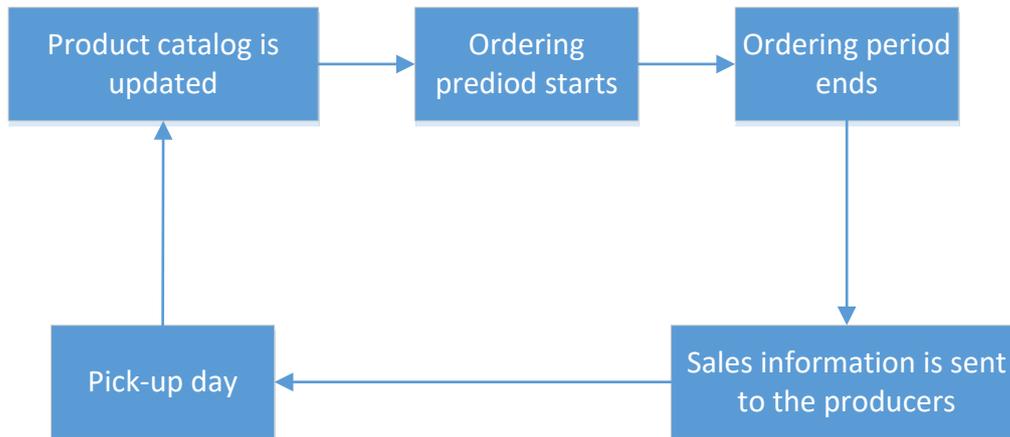
### **3 CASE: LAPPEENRANNAN RUOKAPIIRI**

This chapter describes the case project. First, Lappeenrannan Ruokapiiri's operating model and its dependency on the used online service is explained. That is followed by the description of the old system and its shortcomings, which led to the acquisition of the new system. Third subchapter describes the preliminary work done for exploring the different solutions, preliminary requirements set for the new system, available resources Lappeenrannan Ruokapiiri had and how those influenced the project. After that, different considered solutions are presented and the chosen option is described. In the last subchapter, the course of the project is described in detail.

#### **3.1 Lappeenrannan Ruokapiiri's Operating Model**

Lappeenrannan Ruokapiiri is an association which aim is to help local food producers and interested customers to find each other. The association is a small organization operating in Lappeenranta, Finland. They provide a service, which works as a bridge between local producers and the association's members. In monthly schedule between September and May, the association takes orders from its members and passes them to local producers. On agreed date and place, members of the association pick up their orders. There are currently over 70 customers taking a part. The association favors organic products but the locality of the producer is the main factor. Lappeenrannan Ruokapiiri is non-profitable organization. (Lappeenrannan ruokapiiri ry, 2015)

In order for accomplishing its objective, Lappeenrannan Ruokapiiri needs an online service providing membership register, order management and e-commerce functionalities. The operating model of the association is described in the process diagram below.



**Figure 1.** The operating model process diagram.

**Product catalog is updated:** At the beginning of the monthly cycle, the producers taking part on the association’s activities inform the association about the products they are offering for upcoming ordering period. They are required to tell the pricing information and the quantities of the products if there is going to be a limited batch. The communication between the association and the producers is conducted by the e-mail or by the phone.

The user with stock manager rights to the system updates the product stock after receiving information from the producers by inputting the products, quantities and prices into the system. The opening date of the ordering period is announced to the members of the association in advance and the product stock has to be up to date before that date.

**Ordering period starts:** On agreed date, the possibility of placing an order is opened to the members of the association. Every member has a user account and can login to the system with a username and a password for placing orders. User can see available products without logging in but cannot place orders. Payments are made online with the exception of some products which are paid on the pick-up day. The system is open for new orders for a week.

**Ordering period ends:** When one week has passed the possibility to place new orders ends.

**Sales information is sent to the producers:** After the end of the ordering period, the lists of ordered products and product quantities are sent to the producers. The producers have one week to process the orders and prepare for the delivery.

**Pick-up day:** Products are delivered to the members of the association on the pick-up day, which occurs a week after the ending of an ordering period. A group of volunteers consisted of the members of the Lappeenrannan Ruokapiiri runs the pick-up event. Producers deliver their products to the event. The member coordinating the event has a list for every member, which contains what products they have ordered and in which quantities. When arriving at the event the member gets the list and can collect the ordered items.

### 3.2 The Old System

The old system was introduced in 2011 and made it possible to accept more than 20 participants thus making the growth of the association possible (Lappeenrannan Uutiset, 2011). Before that, all the orders were managed manually. The old system was specifically made for the organizations like Lappeenrannan Ruokapiiri and was originally used by similar association called Jyväskylän opiskelijoiden luomuruokapiiri ry.

The old system, named Ruokapiiri, is an open source GPL licensed application. It is developed for the most part by Asko Soukka by using server-side scripting language PHP and it is required to install PHP Extension and Application Repository (PEAR) packages to run it. The system is freely available online and it is hosted in the Google Code website (Soukka, n.d.).

During the year 2013, the growing number of members and producers lead to issues with the old system. The increased number of members meant increased amount of orders and the increased number of producers meant the increase in number of products that has to be inputted to the system.

### **3.2.1 Technical Performance**

In the old system, placing orders functionality was implemented as a single web page that showed available products and a single form, which users filled and submitted. The whole product catalog was downloaded and shown at once on a single page. According to the members of the board, who were dealing with the system frequently, the increase in the number of products caused prolonged loading times. Earlier when less products were available the loading times were tolerable. In addition, when customers were placing their orders, the saving of the order, the submitting of the form, was reported to take unacceptable long time. Thus, it is reasonably to conclude that the system was not sufficient anymore in the terms of performance and the implementation was outdated.

### **3.2.2 Usability**

The user interface was reported to be cumbersome causing unnecessary workload to the user. On the administration side the managing of different products and producers were reported to be difficult. All products were presented in a single list without any kind of sorting options. There were no means for arranging products in different categories. Every time user added a new product, it was placed on the bottom of the list. Sorting of the list by producers or by any other parameter had to be done manually by dragging the product from the bottom of the list to the desired place on the list.

When making an order, products were presented to the customer as a single list in the similar way as to the administrative user. No sorting or paging functionalities were provided. User made an order by scrolling through a long list on a single page and inputted the number of the products wanted in the box next to every product. This was considered cumbersome and the user experience did not match the experience customers were accustomed to when using similar websites such as e-commerce sites.

### **3.2.3 Missing Features**

In monthly schedule between September and May, the orders are taken and passed to the producers. At the end of every month, the producers deliver their products to the agreed

location where the members come to pick up their orders. To be able to manage the task a person running the pick-up event needs two lists: one containing the orders sorted by the ordered and one containing the orders sorted by the producer. The old system did not have the functionality to produce the summary report of the monthly orders in the format needed in a delivery day. This work, which could have been automatized, was done by hand and caused unnecessary workload.

Every member of the association is required to have an account in the system in order to place orders during ordering periods. The old system's database did not provide all required fields for the required member information. The contact information, for example, could not be stored in the system; therefore, the membership register of the association was maintained manually in the separate system. Maintaining duplicate lists of the members containing overlapping information was considered redundant by the board of the association.

The old system did not provide any payment gateways to the members, therefore all payments were done by bank transactions. Automatic confirmation messages of the succeeded orders and possibility to add product images were missing. In general, it is reasonable to assert that many basic features found in the most popular e-commerce websites were absent.

#### **3.2.4 Upkeep and Maintenance**

One of the members of the board of Lappeenrannan Ruokapiiri had been in charge of upkeep and maintenance of the old system. His task was to communicate with the web hosting service and solve all occurred technical problems with the system. However, it was imminent that he was not able to continue in this role in the future.

The developer of Ruokapiiri application had quit the maintenance and the development of it although it was still in use. The system is free and open source and it is therefore possible for someone else to continue its development; however, it has not been done.

Ruokapiiri application comes with a readme file, which has installation instructions included, and the code is commented in the Finnish language. No other documentation

exists. It requires formidable commitment to get familiar with the program code and to continue the development from that premise. In the readme file the developer of the application comments the state of the documentation in the following way, translated by the author from Finnish (Soukka, n.d.):

*“No new installations of the Ruokapiiri has been done in many years, so even these sparse instructions are probably obsolete.”*

### **3.3 Preliminary Work**

During the second half of the year 2013, it was imminent that something had to be done for the system used to run the services of Lappeenrannan Ruokapiiri. The system could still perform its tasks but the problems were accumulating in the accelerating pace because of the growth of the association. The board predicted that if the growth continued in the next year the work of running the tasks with the old system could become impossible.

#### **3.3.1 Preliminary Study**

The board of Lappeenrannan Ruokapiiri had two options: the old system could be upgraded to match the current requirements or they could acquire a completely new system. Because the member of the board who had maintained the old system was not available for the task in the future, the association started to seek help from outside. One of the members of the board knew the university student who was studying software engineering at LUT and contacted him. The meeting was set between him and the board.

The meeting was held on October 14, 2013. It was noted that Ruokapiiri application was not maintained anymore and it was found to be old fashioned by design. Its documentation was lacking and it seemed unreasonable to use available resources for upgrading it when better options were available. The decision was made to obtain a completely new system and the student was assigned to explore the possibilities for implementation of the new system and report his findings in the next meeting.

The preliminary study exploring different options was produced and presented in the meeting in January, 2014. During the meeting, it was decided that the student who had investigated the possibilities should also implement the new system.

### 3.3.2 Initial Requirement Specification

The preliminary requirements for the new system were collected during the meeting on October 14, 2013. The requirements list was composed by writing down the ideas which surfaced during the conversation. These preliminary requirements formed the base for the investigation for finding possible options for implementing the new system, which are discussed in-depth in the chapter 3.4. The table containing preliminary requirements, translated from Finnish, is presented below. The requirements are presented in the same format as during the board meeting dealing the issue on January 15, 2014. The requirements are not sorted by any priority.

**Table 1** Initial requirement specification.

<b>ID</b>	<b>Requirement</b>
1	E-payments
2	Own account for logging in for every member
3	Summaries after every ordering period, by producers and by orderers
4	Filtering and sorting of the product list by product types, producers etc.
5	Product images, shown with mouse click
6	Sorting of the products in the admin panel
7	Producer register
8	Membership register, possibility to inactivate a member
9	User profiles with different rights
10	Sending orders to the producers automatically after a review
11	Automatic confirmation messages by e-mail of the new orders
12	Possibility to hide products without deleting them
13	Room for growth, 2x or 10x compared to the current system

14	Ordering quotas, product is shown on the list although it is out of stock
15	Importing product information from Excel
16	Statistics of the orders by producers and easy exportability to Excel
17	Undo-button to admin panel
18	Cart shouldn't be emptied after pressing back button
19	Possibility to empty the cart and to go back and add more products anytime
20	Confirmation message which looks like an invoice
21	Possibility to edit confirmation messages
22	Manual for basic user and admin

The main request for the new system was the lack of the intolerable problems of the previous system. The performance issues, related to the growing amount of data the system had to process and show in the browser, were seen as a major problem and shouldn't be present in the new system now or in the foreseeable future. The requirement number 13 in the table 1 describes that new system should perform effortlessly under the load of ten times higher than currently present with the old system.

The usability issues and the lack of features helping in managing the growing number of different products inputted in the system were considered a major issue with the previous system. The requirements describing features helping to reduce the workload of the admin user and increasing efficiency were considered important. For example, the possibility to generate summary reports at the end of the ordering period containing orders sorted by producers and orderers, described in requirement number 3 in the table 1, were seen as essential feature.

As described in the chapter 3.2, the old system's database did not provide all required fields for the required member information; therefore, the membership register of the association was maintained manually in the separate system. These kinds of redundancies were generating unnecessary workload and the requirements for membership register and producer register, requirements number 7 and 8 in the table 1, were considered important.

It was a consensus between the members of the board that the new system should provide user experience and features which were familiar from various e-commerce sites. Features that are typically available in popular e-commerce sites such as e-payments, adding products to the cart and browsing products on the filtered lists were absent in the old system.

During the development process, new requirements appeared and were implemented during the project, which is considered a common phenomenon in the software projects executed in the iterative manner (Haikala & Mikkonen, 2011). The development process and the elaboration of the requirements is discussed in the chapter 3.6. Not all requirements in the preliminary requirement specification were considered important in the later parts of project. Requirements number 5, 10, 15, 17 and 21 in the table 1 were not implemented during the project, for example.

### **3.3.3 Resources**

Lappeenrannan Ruokapiiri is run by volunteers. Only a small subset of the members was taking part on the actual running of the association besides being in the role of the customer. The members running the association have their own fulltime jobs and other daily activities. This makes time a scarce resource. The process of acquiring and building the new system causes extra effort for the members who are already giving their time and effort without payment. The workload caused by the process of developing and maintaining the system after it is deployed was therefore preferred to be low as possible.

Lappeenrannan Ruokapiiri is a non-profit organization. It gets its income for running its activities from its members. New members pay a fee of 30 euros when joining to the association and after that an annual fee of 15 euros. If the association has, for example, 80 members, it totals 1200 euros' annual income plus the joining fees of the new members. It can be therefore said that the size of the budget can cause some limitations if the intention is to outsource the development and maintenance of the system.

It is safe to assume that there is lack of knowledge on running software projects in small organizations which are not operating in the field of information technology. This can cause challenges during a project, as majority of stakeholders are unfamiliar with the process and

must be introduced to it. However, in the case of Lappeenrannan Ruokapiiri, there were exceptionally many IT-professionals on the board thus making challenges caused by above-mentioned reason a minor issue or not an issue at all.

### **3.4 Considered Implementation Possibilities for the New System**

Lappeenrannan Ruokapiiri decided to build a new system. In this chapter, considered possibilities for building the new system are described and arguments, which were present for and against each option, are described. There are three different options: build the new system from scratch, build the new system using a pre-existing CMS or use an e-commerce website service.

#### **3.4.1 Building Own System from Scratch**

The most rudimentary option is to build a new system from scratch. This is the most demanding option, because it requires most programming skills, time and money. In paper (Pelkkikangas, 2010), it is concluded that when using self-made platforms, the implementation of the upgrades was considered to be expensive and time consuming.

Lappeenrannan Ruokapiiri had hired one person for the work. In the case where there is only one person responsible for the development and maintenance of the system, the fate of the previous system, where the developer quits and nobody continues the work, could be a possible outcome. In the option of using a well-established, documented and popular open source platform with millions of users as a base, there is a little or no risk for this happening. This was considered important.

#### **3.4.2 Building Own System Using a CMS**

Building the new system by using a pre-existing CMS as a base was thought to provide significant benefits compared to building the system from ground up. The usage of a well-established, documented and popular CMS was considered to ensure the continuity of the system and to make it maintainable in the future when personnel changes. When implementing the system, there is no need to develop all the features from scratch because a

ready-made CMS itself contains basic features and various different extensions are available for extending the core functionalities.

When using a pre-existing CMS, the system can be built by combining ready-made parts and configuring them. Only those required features which are not available as extensions or are not provided by a CMS itself needs to be implemented by programming. In addition, when using an open source CMS all the ready-made parts are fully customizable by modifying the code.

In the preliminary study, the following CMSs were considered: Drupal, Joomla, Magento, PrestaShop and WordPress. Drupal, Joomla and WordPress are general purpose CMSs and Magento and PrestaShop are specifically designed for e-commerce purposes.

Drupal is the third most popular CMS in the world and it is used by 2.1% of the all the websites. Its CMS market share is 5.0% (W3Techs, 2015). Drupal is flexible and modularity is one of its core principles. The Drupal project is open source software and it is distributed under GNU General Public License. The project was started in the year 2000 by Dries Buytaert and Hans Snijder, both were students of University of Antwerp. The first version was released by Dries Buytaert in January of 2001 (Drupal, 2015).

Joomla is the second most popular CMS in the world and it is used by 2.8% of the all the websites. Its CMS market share is 6.5% (W3Techs, 2015). According to (Wiklund, 2015), Joomla is considered to be a compromise between WordPress and Drupal, because it is more customizable than WordPress but does not require as much technical expertise as Drupal. Joomla is distributed under GNU General Public License and the first version was released on the year 2005 (Open Source Matters, 2015b).

WordPress is the most popular CMS in the world. It is used by 25.3% of all the websites and its CMS market share is 58.7% (W3Techs, 2015). WordPress was originally designed as a blogging platform but has eventually developed to a fully featured CMS. Thousands of plugins available that extend its capabilities makes its functionality nearly limitless (WordPress, 2015b). WordPress is released under GPLv2 license, or later, of the Free

Software Foundation (WordPress, 2015a). The first version was released by Matt Mullenweg, one of the founders of WordPress, on May 27, 2003 (WordPress, 2003).

Magento is the fourth most popular CMS in the world and it is used by 1.2% of the all the websites. Its CMS market share is 2.9% (W3Techs, 2015). Magento is open source CMS focusing solely on e-commerce. Magento is offered as a free Community Edition and as paid versions such as Enterprise Edition, which offers more features than the free one. Magento Community Edition is distributed under the Open Software License (OSL) v. 3.0 (Magento, 2015c). The first version was released on March 31, 2008 (Magento, 2015b).

PrestaShop is used by 0.5% of the all the websites. Its CMS market share is 1.3% (W3Techs, 2015). PrestaShop is a free open source CMS focusing solely on e-commerce purposes. PrestaShop is released under the OSL 3.0 and its modules are released under the Academic Free License 3.0 (PrestaShop, 2015b).

### **3.4.3 Using an E-commerce Website Service**

Third considered option was to use one of the e-commerce website services mentioned in the chapter 2.2. When using an e-commerce website service, the web store can be deployed without any programming skills and all management can be done through a web interface. These ready-made solutions offer an easy way to start an e-commerce website and provide different levels of paid packages with different feature sets. Some services provide free options. For example, MyCashflow provides a free option, but it is extremely limited and allows only 10 products in the store (Pulse247, 2015b).

Limitations in the customization capabilities was considered the most prominent deficiency of the e-commerce website services. In the case of Lappeenrannan Ruokapiiri, multiple specific features were required, as seen in the table 1, and after studying the various e-commerce website services it was concluded that fulfilling those requirements could not be guaranteed if an e-commerce website service would be selected as a method of implementation. Another unwanted property was monthly payments.

### 3.5 Selected Solution

During the preliminary study, it became clear that building the new system by using a pre-existing CMS was the most rational choice. Although not entirely impossible, the option of completely self-made system was quickly abandoned because of the option's substantial weaknesses considering the available resources and future prospects. Usage of the e-commerce website service was considered to provide too limited possibilities.

When comparing different CMS options, Magento and PrestaShop were discarded because there was possibility for feature requests unconventional to the conventional e-commerce website. The forthcoming system was seen as an order management system and a member register besides the web store functionality.

Drupal, Joomla and WordPress were all seen as capable systems for providing features and customizability needed in the project. Because the developer had previous experience on WordPress, the decision was made, in the meeting on January 15, 2014, to build the new system using WordPress. It was decided that e-commerce functionalities would be added to WordPress by using WooCommerce plugin by WooThemes (Automattic, 2015a).

#### 3.5.1 WordPress

WordPress is based on PHP and it uses MySQL relational database management system (RDBMS). To run WordPress PHP version 5.2.4 or newer and MySQL 5.0.2 or newer is required (WordPress, 2015c). It is recommended to use Apache or Nginx servers for running WordPress, but any server that supports PHP and MySQL will work (WordPress, 2015d).

A WordPress site is customized by using themes. A WordPress theme is a collection of files that work together to create the design and functionality of a WordPress site. Theme includes a stylesheet file *style.css*, which contains CSS style information and description of the theme, various template files and can optionally use a functions file, which is named as *functions.php* (WordPress, 2015c).

WordPress uses templates to generate pages dynamically. Templates are PHP source files, the building blocks of a WordPress site, which are used to generate the pages requested by visitors. A simple WordPress web page can be made of three basic blocks, which are a header, the content and a footer. Each of these blocks is generated by a template file. The presentation of the content can be edited by editing templates. Template files are made up of HTML, PHP and WordPress template tags. Template tags are used to display information dynamically. For example, using *get\_header()* template includes the *header.php* template from current theme's directory and using *wp\_login\_form()* template provides a simple login form for use anywhere within WordPress. (WordPress, 2015c)

In WordPress, the content is presented in two forms: Posts and Pages. Posts are more dynamic, time depended and flexible form of content than Pages. WordPress was originally developed as blogging platform and the name "post" comes from that background. Every blog post is handled as a Post. When using WordPress to build an e-commerce website, every product added to the store is handled as a Post by WordPress. In order to show Posts, The Loop, which is PHP code used by WordPress to display posts, is used in the template files. Pages are for content that is less time-dependent than posts. Pages can be organized into pages and subpages. It is possible to build a WordPress site, which consists only of pages. Common pages found on websites, such as "About" and "Contact" pages are handled as a Page in WordPress, for example. (WordPress, 2015c)

A theme can optionally use functions file, which is named *functions.php*, and acts as a plugin. A functions file is loaded automatically during WordPress initialization for both admin pages and external pages. If the function should be available to more than one parent theme, the function should be created in a plugin instead of a functions file. (WordPress, 2015c)

Plugins are the way to extend WordPress. The core of WordPress is designed to be lightweight in order to maximize flexibility and minimize code bloat. Users can use WordPress Plugins to add more functions and features to fulfill their needs. WordPress Plugins are composed of PHP scripts (WordPress, 2015c). Plugins can be acquired from WordPress Plugin Directory, which contains tens of thousands of plugins (WordPress, 2015e).

Information and documentation of WordPress can be found from WordPress Codex, the online manual for WordPress (WordPress, 2015c).

### 3.5.2 WooCommerce

WooCommerce is an open source WordPress Plugin which integrates seamlessly with WordPress and provides all the basic e-commerce functionalities to it (Automattic, 2015a). WooCommerce consists of the core and the extensions. The core part is free, but WooThemes, the developer of the WooCommerce, provides free and paid extensions to it. Third parties are also able to develop and publish WooCommerce extensions. Extensions can be explored and acquired from the WooCommerce Extensions page (Automattic, 2015b).

WooCommerce is the world's second most popular e-commerce solution with its 8% market share of websites using e-commerce technologies (BuiltWith, 2015).

### 3.5.3 Plugins

When building the new system, various free and some paid plugins were installed to obtain required features and functionalities. Some plugins were installed for maintenance and development purposes. All the plugins used and developed for the system are listed below in the Table 2. Descriptions are provided by the authors of the plugins. (Automattic, 2015b) (WordPress, 2015e)

**Table 2** Installed WordPress Plugins

<b>Name</b>	<b>Source</b>	<b>Price</b>	<b>Description</b>
Admin Menu Editor	WP Plugin Directory	Free	Let you directly edit the WordPress admin menu. You can re-order, hide or rename existing menus, add custom menus and more.
BackUpWordPress	WP Plugin Directory	Free	Simple automated backups of your WordPress powered website.

Database Browser	WP Plugin Directory	Free	Easily browse the data in your database, and download in CSV, XML, SQL and JSON format.
Disable Users	WP Plugin Directory	Free	This plugin provides the ability to disable specific user accounts.
Maksukaista Payment Gateway	(Paybyway, 2015)	Free	Maksukaista Payment Gateway Integration for WooCommerce.
Members	WP Plugin Directory	Free	A user and role management plugin that puts you in full control of your site's permissions. This plugin allows you to edit your roles and their capabilities, clone existing roles, assign multiple roles per user, block post content, or even make your site completely private.
Menu Items Visibility Control	WP Plugin Directory	Free	Control the display logic of individual menu items.
SB Welcome Email Editor	WP Plugin Directory	Free	Allows you to change the content, layout and even add an attachment for many of the inbuilt WordPress emails. Simple!
Search By Product tag - for WooCommerce	WP Plugin Directory	Free	The search functionality in WooCommerce doesn't search by product tags by default. This simple plugin adds this functionality to both the admin site and regular search.
Under Construction	WP Plugin Directory	Free	Makes it so your site can only be accessed by users who log in. Useful for developing a site on a live server, without the world being able to see it.
WooCommerce	WP Plugin Directory	Free	WooCommerce is a powerful, extendable eCommerce plugin that helps you sell anything. Beautifully.

WooCommerce Colors	WP Plugin Directory	Free	This plugin adds a new section called WooCommerce on the Customize menu, allowing you to easily change color of buttons and WooCommerce elements.
WooCommerce Product Vendors	(Automattic, 2015b)	\$79	Set up a multi-vendor marketplace that allows vendors to manage their own products and earn commissions, or simply assign commissions from sales.
WooCommerce Products Per Page	WP Plugin Directory	Free	Integrate a 'products per page' dropdown on your WooCommerce website! Set-up in seconds!
WooCommerce Vendor List Widget	Developed for the case	-	Widget that shows list of the product vendors with links to the vendor pages
WooCommerce yhteenvedot	Developed for the case	-	Creates an admin page for generating summaries of the orders by orderers and by producers.
WooThemes Helper	(Automattic, 2015b)	Free	Hi there. I'm here to help you manage licenses for your WooThemes products, as well as help out when you need a guiding hand.

### 3.6 Course of the Project

This chapter describes the course of the project from the January, 2014, when the meeting was held to start the development of the system, to the fall of the 2016. At first, the used development process model is described and the initial development phase of the system is described in detail. All the challenges occurred during the initial development phase are presented. Lastly, the continuous development and maintenance period from fall 2014 to fall 2016 is described including all the occurrences significant to this study.

### 3.6.1 Development Process

The system was developed following agile software development practices although not any specific methodology was followed. In the Manifesto for agile software development, following principles are stated (Highsmith & Cockburn, 2001) :

- individuals and interactions over processes and tools
- working software over comprehensive documentation
- customer collaboration over contract negotiation
- responding to change over following plan

The product owner was deeply involved during the development process and some members of the association's board were integral part of the development team. The system was developed in iterative manner and feedback and suggestions were collected from the Lappeenranta Ruokapiiri in testing sessions held at four to six week intervals during the year of 2014. The system was deployed on the web host's server during the development and was always fully accessible to the product owner. The collected feedback guided the development until the system was considered ready for deployment.

After the deployment, e-mail worked as a feedback channel between the developer and the product owner. Contact person from the product owner's side submitted feedback, including user experiences and suggestions for new features, during the usage of the system to the developer and the feedback was used for building improvements to the system. Therefore, it can be said that in addition to methods of agile software development process the used process was inspired by continuous deployment process and followed those practices excluding the automation of the usage data collection. Continuous deployment is described in (Olsson, et al., 2012) in the following way, in the context of evolution of companies' software development practices:

*“At this stage, software functionality is deployed continuously, or at least more frequently, at customer site. This allows for continuous customer feedback, the ability to learn from customer usage data, i.e. real usage data.”*

### 3.6.2 Initial Development

The initial development was conducted in five iterations. At the end of each iteration testing session was arranged where six to eight people from the association and the developer of the system participated. Meetings were held in relaxed and conversational manner which enabled constructive working environment and open communication between every participant. During these sessions, the deployed features were tested, bugs were reported and feature requests were made. Several requirements not included in the preliminary requirements were added during the iterations. Observations of each testing sessions are summarized and presented in the table 3 below. After the table, each iteration is described and most important observations are presented in more detail.

**Table 3** Testing sessions

<b>Iteration</b>	<b>End date</b>	<b>Observations and suggested improvements</b>
1	April 24th	<ul style="list-style-type: none"><li>- Various UI (user interface) and usability improvements were suggested</li><li>- The summary function requirement was elaborated and clarified</li><li>- New feature requirement: add on/off switch for the shop</li><li>- Password recovery feature reported to not work as intended</li><li>- Bug in user account creation</li></ul>
2	June 2nd	<ul style="list-style-type: none"><li>- Various UI and user experience improvements were suggested</li><li>- Translation improvements</li><li>- Two new functionalities requested</li><li>- System should be in use on August, 2014</li></ul>
3	July 3rd	<ul style="list-style-type: none"><li>- The summary function was delivered and tested</li><li>- Various UI and user experience improvements were suggested</li><li>- Refinements to the features were suggested</li><li>- Requirement for product images was discarded</li></ul>

4	August 10th	<ul style="list-style-type: none"> <li>- On/off switch delivered and tested</li> <li>- Various UI and user experience improvements were suggested</li> <li>- Refinements to the features were suggested</li> <li>- Two new functionalities requested</li> </ul>
5	Aug 28th	<ul style="list-style-type: none"> <li>- Various UI and user experience improvements were suggested</li> <li>- Refinements to the features were suggested</li> <li>- Payment service integration</li> <li>- Content of the required user manual was described</li> </ul>

The first delivery of the new system was tested in the testing session on April 24th, 2014.

Various UI improvements were requested, including following suggestions:

- move “log out” button to better place
- add “continue shopping” button on the checkout page
- no images on product list page, only in detailed product descriptions
- show more information of products in product list page
- show product categories

By default, WordPress and WooCommerce included fields not needed by the product owner. The system asked user to fill fields asking country and company name, for example. Unneeded fields were removed and new requested fields were added, such as field for bank account number and checkbox for signing up for helping on the delivery day. New feature requirement, the on/off switch for the shop, was added and requirement for summary system was elaborated and its importance emphasized. On/off switch feature was requested because the product owner needed the ability to close the e-commerce section of the webpage between ordering periods. Requirements for the summary function were elaborated and the use case for using summaries by producers and orderers during the delivery day was described in detail to the developer.

During the second iteration requested UI and user experience improvements were made and reported bugs were fixed. New features were not yet implemented in this point. During the second testing session on June 2<sup>nd</sup>, 2014, various UI and user experience improvements were suggested again, such as moving or hiding various fields and buttons, hiding product images from the cart, adding some informative texts and doing some translation improvements. Parts of the WooCommerce plugin and other WordPress plugins used in the system were not available in Finnish language, which was the language the product owner requested, therefore the developer of the system had to translate them from English to Finnish. It was requested that search function should find producers and there should be feature for resetting product stocks after ordering periods. Priority for these requirements were considered low and these requirements were not fulfilled during the initial development of the system. During the second testing session, it was agreed that the system should be ready for use during August, 2014.

On the third iteration, the summary function was delivered and suggested UI, user experience and translation improvements were implemented. At the end of the iteration, testing session was arranged on July 3<sup>rd</sup>, 2014. During the session, the summary function was tested and some flaws were found, of which the most severe was that the summary function page was not shown in the admin panel for the user with the shop manager user rights. The system has three different user rights levels, which are: admin, shop manager and customer. With the shop manager account, the user had to be able to run the e-commerce site and do all management tasks required for running the organization.

During the third meeting, the test group came to the conclusion that product images were not needed in the system. This is a representative example of the initial requirement which was discarded during the development process when the actual system was tested and evaluated by the product owner. The UI and general user experience were refined in each iteration in this way.

On the fourth iteration, the on/off switch feature was implemented and the UI and the user experience was further improved based on the feedback. During the fourth testing session, arranged on August 10<sup>th</sup>, 2014, feedback considering features and user experience was again

collected. The new requirement for database backup feature was made. The used web host had its own backups of the servers but the product owner wanted the ability to make their own backups of the database. In addition, the product owner requested Google Analytics integration which would provide tools for collecting and analyzing usage statistics of the online service. All in all, the product owner was satisfied with the system and the course of the project and the original deployment schedule was still in effect.

The last testing session, which ended the fifth iteration, was held on August 28<sup>th</sup>, 2014. Some minor user experience problems were still reported and request was made to fix them before the deployment. In addition, the summary function incorrectly showed cancelled orders in the summaries, a flaw requiring quick fix before the launch. The system would be receiving its first orders on September 9<sup>th</sup>, 2014, and before that the staff would input all user data and products to the system. The developer was given four days to do final improvements, to finish payment service integration and to make the service publicly available under the right domain.

All problems found during the fifth testing session were solved and all the required tasks were completed in time. During the integration of the payment service the bug was found from the program code of the payment service's WordPress gateway plugin. The bug was fixed with the help of the payment service's customer support. The bug fix was included in the next version of the plugin. The system was deployed successfully on schedule, which was proved to be critical because the updates on the web host's server rendered the old unadaptable system unusable during the summer.

### **3.6.3 Continuous Development and Maintenance**

After the deployment of the system the continuous development and maintenance phase began. Improvements to the system were made and new features were added based on the obtained user feedback. Maintaining the system included technical support given to the product owner by e-mail and installing updates to the WordPress installation and to the various installed plugins listed in the chapter 3.5.3. After the WordPress version 3.7 minor and security updates are installed automatically in the background and manual installation is needed only for major feature releases (WordPress, 2016a).

These features and improvements were requested and implemented around the time of the first ordering period on September and October, 2014:

- show producers in the product list page of the web shop
- show producers name next to a product name in the order confirmation e-mail
- show if promised to come help on delivery day in order information
- improvements to the summary function, including
  - o show comments given with orders
  - o show prices on every row
  - o show the total value of the orders in the summary by producers
  - o show producers name next to a product name in the summary by producers
  - o show list of the e-mail addresses of the all orderers

The list above shows a representative example of the type of feature requests received on the early phases of the adoption of the system. After the first six months, the system was matured in that extend that new feature requests became less frequent and the maintenance tasks consisted mostly of installing updates to the WordPress and to the plugins. These updates sometimes contained feature updates which benefitted the product owner but was mainly installed for the bug fixes and security updates.

During the first ordering period on September, 2014, WooThemes introduced the new major version of the WooCommerce plugin with version number 2.2. Upgrade to the new version broke the custom-made summary functionality of the system. When trying to produce summaries, the system did not print any results. In the WooCommerce 2.2 the way the plugin stored order statuses in the database was refactored. Order statuses were now stored in the *post\_status* field in the *wp\_posts* database table. In the old version, order statuses were stored as a post taxonomy in the *taxonomy* field of the *wp\_term\_taxonomy* database table. This is a good example of an update which breaks compatibility between a platform/plugin and a custom-built WordPress theme/plugin. The plugin providing the summary functionality was updated after the first ordering period to be compatible with the WooCommerce 2.2.

Figure 2 below presents a screenshot showing the UI of the summary plugin in the WordPress admin panel. The user inputs begin and end date of the time period and the system prints tables containing the summary information.

The screenshot shows the WordPress admin panel for 'Lappeenrannan ruokapiiri'. The 'Yhteenvedot' (Summary) plugin is active. The interface includes a sidebar with navigation options like 'Ohjausnäkyvä', 'Media', 'Sivut', 'WooCommerce', 'Tuotteet', 'Tuottajat', 'Yhteenvedot', 'Tilauskausi', 'Käyttäjät', 'Ulkoasu', 'Lisäosat', 'Työkalut', 'Asetukset', and 'Piilota valikko'. The main content area is titled 'Yhteenvedot' and contains instructions: 'Syötä aikaväli ja valitse mitä haluat yhteenvetoon. Voit myös syöttää pelkän aloituspäivän. Käytä päiväyksessä esimerkiksi muotoa pp.kk.vvvv.' Below this, there is a date range selector set to '1.9.2015 - 1.10.2015' and a 'Näytä' button. The table below shows the following data:

Tuote	Tuottaja	Määrä (kpl)	Hinta (€)
Hunaja, 0,5 kg	Hannu Kälviäinen	10	60,00
Peruna van Gogh, 2 kg	Heikkisen luomutila	2	6,00
Sipuli, 1 kg, Luomu	Heikkisen luomutila	1	6,00
Palsternakka, 1 kg, Luomu	Heikkisen luomutila	1	5,00
Tyrnimarja 1 kg, pakaste	Jaakko Nikula	1	20,00
Savustettu ruodoton kirjolohifilee, noin 700-900 g	Kotisavu	1	0,00
Kananmunia, 30 kpl	Kuortisen luomukanala	3	25,65
Naudan luullinen keittoliha n. 1 kg, pakaste	Maatila Peltokukka	1	0,00
Broilerikeitto, 1 litra	Raija Gren	1	7,00
Lammasjauheliha, 1 kg, vakuumpakattu pakaste	Seppo Villa	1	13,00
Hermannin leipä (tumma siemenleipä, n. 500 - 600 g)	Seppo Villa	1	4,50
<b>Yhteensä:</b>			<b>147,15 €</b>

**Figure 2.** The summary plugin.

In May, 2015, it was reported that in some cases more products were added to the cart than users actually intended and therefore some users, by accident, ordered more products than they actually wanted. During the investigation, it was found that the problematic behavior was not caused by a bug in the system but was a usability issue related on the HTML form-elements used in the web shop. Users were reloading the page and were accidentally sending forms more times than intended despite web browser's warning messages. The problem was fixed with the new AJAX (Asynchronous JavaScript and XML) implementation of the "add to cart" buttons in the shop page. Products were now added to the cart without reloading the

page. In the figure 3 below is a screenshot of the web shop showing the product list page and “add to cart” buttons implemented with the AJAX techniques.



Etusivu Kauppa Ostoskori Oma tili Kirjautu ulos

**Tuoteryhmät**

- Joulupaketti
- Juustot
- Kalat
- Kasvikset ja marjat
- Kotiruoka
- Langat
- Leivät ja leivonnaiset
- Lihat
- Makeiset
- Marjasäilykkeet
- Mehut
- Munat
- Muut tuotteet
- Viljatuotteet

**Tuottajat**

- Anne Huhtanen
- Heikkisen luomutila
- Jaakko Nikula
- Jaakkolan tila
- Johanna Jauhiainen
- Kotisavu

## Kauppa

Näytetään 1–20 – 262:sta tuloksesta Lajittelu, oletustapa 20 tuotetta per sivu

Tuote	Tuottaja	Hinta		
6 munan kermakakku	Raija Gren	30,00 €	1	Lisää ostoskoriin
Ahvenkukko	Kotisavu	24,00 €	1	Lisää ostoskoriin ✓
Katso ostoskori				
Appelsiini-inkiväärifudge	Suomen Karamellitehdas	3,50 €	1	Lisää ostoskoriin
Basilikainen Juhlakartano, 500 g	Mäkelän tilajuustola	7,50 €	1	Lisää ostoskoriin
Broilerikeitto, 1 litra	Raija Gren	7,00 €	1	Lisää ostoskoriin
Cappuccinofudge	Suomen Karamellitehdas	3,50 €	1	Lisää ostoskoriin
Emmental 3 kk, 220 g	Mäkelän tilajuustola	3,50 €	1	Lisää ostoskoriin
Emmental 3 kk, 500 g	Mäkelän tilajuustola	7,50 €	1	Lisää ostoskoriin

**Figure 3.** The product list page.

A developer can customize WooCommerce by overriding its default template files in developer’s own WordPress theme. If a developer wants to customize, for example, a form which is used when users edit their account information, the *form-edit-account.php* template file can be overridden with an own implementation of it in the theme. However, sometimes WooThemes updates WooCommerce’s functioning and its default template files in version updates, especially in the major releases such as WooCommerce 2.2 and 2.6 and also in minor releases. These updates can break compatibility between WooCommerce and custom themes and requires developers to check the template compatibility regularly and to do needed updates if necessary (Automattic, 2016c).

The before mentioned AJAX implementation of the “add to cart” buttons broke when WooThemes updated the WooCommerce in early 2016. This required the developer to update theme’s custom made template files in order to retain the desired functionality. This was done in February, 2016. This example represents what kind of compatibility issues custom template files can cause. The fixing of the outdated WooCommerce template files in the theme was revealed to be the most time-consuming maintenance task of the Lappeenrannan Ruokapiiri’s new system. Other plugin updates and WordPress’s updates rarely broke compatibility with the custom-made implementations.

## **4 DISCUSSION**

In this chapter, the observations made during the case project are discussed, analyzed and conclusions are drawn based on the observations. At first, the selected solution is evaluated based on how it performed during the project and how it worked during the maintenance and upkeep period. Encountered challenges are discussed and possible solutions are presented. In the second subchapter, the development process is analyzed and its effects on the maintenance of the online service are discussed.

### **4.1 Selected Solution**

Lappeenrannan Ruokapiiri decided to hire a person from outside of the association to develop and maintain their new online service, which would replace their old and outdated system. The old system was lacking in features and its performance was considered unsatisfactory. The new system was built on popular open source CMS WordPress and desired e-commerce functionality was added with popular e-commerce platform WooCommerce. Custom features were implemented through custom made WordPress theme and plugins. At the end, the product owner was satisfied with the solution and the project can be considered successful. The most important requirements were fulfilled and the old system's problems, described in the chapter 3.2, were avoided. The system was deployed in schedule, which proved to be critical because of the failing of the old system.

Other considered option was to build the new system from scratch. Building an own system from scratch would theoretically provide more customization options, but in practice, the greater amount of needed work and resources would lead to fewer implemented features in the same timeframe than when using a well-established CMS as a base. From the viewpoint of maintenance and upkeep, a system built completely from ground up with few resources, like the Lappeenrannan Ruokapiiri's old system was, could also face the fate of the old system, of being difficult or almost impossible to maintain and upgrade because of its obscurity and lack of support by someone familiar with its technical implementation. The lack of support from a greater developer community developing the system apart from the developer, in this case one person, hired by the association, could easily lead to that. The

problem could be reduced by comprehensive documentation which would make future staff transitions easier and someone else could continue the work later in the future. Despite this, it would still contain more risks than a CMS solution, because there are more possibilities for potential failures and it may be harder to find replacement for the current staff, new people who would be willing to work with an obscure system. It can be concluded that building an online service on a pre-existing and well-known CMS is probably the most reasonable solution for a small organization and it helps in preventing or dealing with several possible maintenance and upkeep issues.

#### **4.1.1 Compatibility Issues Caused by Updates**

With a choice to build an online service by using a well-established CMS, implementation of custom features seems to be the biggest risk factor when considering maintenance and upkeep of an online service in the long-term. In the case of Lappeenranta Ruokapiiri, the WordPress and installed third party plugins received regular updates and the admin of the system had to constantly maintain the compatibility between them and the custom-made WordPress theme and plugins. Especially new versions of WooCommerce e-commerce platform caused time consuming maintenance tasks. Examples are found in the chapter 3.6.3. These findings are in line with (Pelkkikangas, 2010), where installing platform updates and extensions were also found to be difficult tasks.

Maintaining compatibility between components in a system like in the case project requires technical knowledge and programming skills. A small organization needs to hire an administrator, with considerable technical knowledge and ability to understand system's inner workings, to maintain it unless they have in-house skills to do that. Another option would be to build a system by using ready-made building blocks, theme and plugins in a case of WordPress, which are maintained by a developer community or by a company making business with them. Thereby long-term maintenance of a system would be secured with a minimal technical skill from an administrator. Possible problems could be fixed by moderately informed person with the help of the online support channels of the chosen CMS. In the case of Lappeenranta Ruokapiiri, the option of building the whole system from ready-made components was not fully possible because the association requested features

which were not always available as a ready-made and well supported WordPress plugins at the time. However, in the cases when such plugins were available they were always used.

It could be technically possible to not install updates to a CMS and its extensions at all after the deployment, but from the standpoint of computer security it is not acceptable alternative. Popular CMSs are popular targets for cyber criminals and it is reasonable to install available security updates. New versions of WordPress often include fixes for security vulnerabilities.

#### **4.1.2 Custom Implementations and Long-term Maintenance**

For securing a long-term maintenance of an online service and preparing for changes in its administration staff, customizations and custom plugins to a CMS should be developed using generally proven software development practices including writing of a clean and well commented program code and by using the practices recommended in the documentation of the chosen CMS. For example, when interacting with a database in WordPress, instead of directly accessing a database it is recommended to use *wpdb* class defined by WordPress (WordPress, 2016b). By using well documented practices a developer can promote the quality and the maintainability of an online service in the future.

If custom made CMS extensions provide functionalities which may be useful outside of the system they were built for, one way for ensuring their continuity and that they are kept up to date is to publish them. For example, if a WordPress plugin is published in the WordPress plugin repository and its source code is made freely available it is possible that a community is formed for updating and maintaining the plugin. This releases resources of a small organization attached to the maintenance of a service and can help to solve the problem of compatibility issues described in the previous chapter.

## **4.2 Development Process**

In the case project the system was developed by using agile software development practices. The development team consisted of the developer and several members of the Lappeenranta University of Applied Sciences who followed the development, tested the system during the

development and provided feedback and suggestions on every iteration. Numerous changes were made based on these observations as described in the chapter 3.6.2. This close collaboration and direct interaction guided the development and made it possible to shape the system to the product owner's needs from the early phases of the project without time consuming missteps.

For example, several UI and usability improvements were made during the initial development period of the case project and the user experience of the online service could be fine-tuned to the product owner's liking early on thanks to the used development method. In (Pelkkikangas, 2010) it is reported that people managing the e-commerce sites have difficulties with managing the content and adding new products. By designing the service closely with the product owner, these kinds of problems can be avoided and the need for providing support for the product owner later on the online service's lifecycle can be reduced.

If a developer comes outside of an organization, close collaboration with a product owner during a development process may prevent maintenance challenges in several ways. A product owner become familiar with a system and its workings before its deployment. Collaboration can reveal use cases and scenarios which may be not seen in advance and can reveal flaws in a system. Early detection of flaws enables the prevention of problems, which may lead to avoidance of difficult and time consuming maintenance tasks when an online service is in actual use. In addition, active testing by end users during a development can reveal requirements for important features initially not noticed when requirement specification was produced thus possibly preventing problems after a deployment of an online service.

## 5 CONCLUSIONS

This thesis was done in Lappeenranta University of Technology and its objective was to study problems small organizations encounter during maintenance of an online service and to find technical and other features, which may help small organizations to deal with identified difficulties. In order to attain this objective, the case study was conducted on acquisition and maintenance of Lappeenrannan Ruokapiiri's new ordering management system. Lappeenrannan Ruokapiiri is a small association operating in Lappeenranta, Finland.

Within the framework of this study, the new system for Lappeenrannan Ruokapiiri was developed, deployed and maintained. Flaws of the previous system were studied and the new system was designed to avoid these problems and to be able to adapt to the present and future requirements. The new system was built on WordPress CMS and e-commerce platform WooCommerce was used to provide e-commerce functionalities. The product owner made various requirements which were provided by custom-made WordPress theme and plugins. The new system was built using agile software development methods in close collaboration with the product owner and was deployed in schedule. The system was developed and improved according to the product owner's feedback during the maintenance phase.

Selected solution proved to be successful in avoiding difficulties encountered during the use of the previous system, which was difficult to upgrade to fulfill current and possible future requirements. The usage of a well-established CMS and community or company supported extensions, whenever possible, proved to be successful solution for alleviating maintenance and upkeep problems of an online service for the viewpoint of a small organization. The used development method made it possible to shape the system towards the desired goal and quickly adapt to changes in requirements during the project which helped to avoid conflicts after the deployment.

During the project, the most significant maintenance challenge encountered was to maintain compatibility between custom-made implementations and the platform during platform updates. It was not always possible to use community or company supported extensions. For

dealing with this problem, publishing custom-made components in the developer community is a possible solution. However, this is not always possible and the problem of compatibility between actively updated components of a CMS based online service could be a topic for further research. Comprehensive answer and developed solution for this problem could alleviate the challenges of maintenance and upkeep of an online service by a small organization.

## REFERENCES

Automatic, 2015a. *WooCommerce Features - WooThemes*. [Online]

Available at: <http://www.woothemes.com/woocommerce/features/>

[Accessed November 2015].

Automatic, 2015b. *WooCommerce Extensions from WooThemes*. [Online]

Available at: <http://www.woothemes.com/product-category/woocommerce-extensions/>

[Accessed November 2015].

Automatic, 2016c. *Fixing Outdated WooCommerce Templates*. [Online]

Available at: <https://docs.woocommerce.com/document/fix-outdated-templates-woocommerce/>

[Accessed 2016].

Benevolo, C. & Negri, S., 2007. Evaluation of Content Management Systems (CMS): a Supply Analysis. *Electronic Journal of Information Systems Evaluation*, 10(1), pp. 9-22.

Berger, C. et al., 2002. *E-commerce payment solution*. USA, Patent No. US20020103752 A1.

Boiko, B., 2002. *Content Management Bible*. New York: Hungry Minds.

BuiltWith, 2015. *Ecommerce technologies Web Usage Statistics*. [Online]

Available at: <http://trends.builtwith.com/shop>

[Accessed November 2015].

Drupal, 2015. *About*. [Online]

Available at: <https://www.drupal.org/about>

[Accessed November 2015].

Grandon, E. E. & Pearson, J. M., 2004. Electronic commerce adoption: an empirical study of small and medium US businesses. *Information & Management*, 42(1), pp. 197-216.

Haikala, I. & Mikkonen, T., 2011. *Ohjelmistotuotannon käytännöt*. s.l.:Talentum.

Highsmith, J. & Cockburn, A., 2001. Agile software development: the business of innovation. *Computer*, 34(9), pp. 120-127.

Hong, W. & Zhu, K., 2006. Migrating to internet-based e-commerce: Factors affecting e-commerce adoption and migration at the firm level. *Information & Management*, 43(2), pp. 204-221.

Lappeenrannan ruokapiiri ry, 2015. *Lappeenrannan ruokapiiri*. [Online]  
Available at: <http://lappeenrannanruokapiiri.fi/>  
[Accessed November 2015].

Lappeenrannan Uutiset, 2011. *Ruokapiirillä on nyt nettitilausjärjestelmä*. [Online]  
Available at: <http://www.lappeenrannanuutiset.fi/artikkeli/82673-ruokapiirilla-on-nyt-nettitilausjarjestelma>  
[Accessed November 2015].

Magento, 2015a. *Omnichannel Commerce Solutions*. [Online]  
Available at: <http://magento.com/products/overview>  
[Accessed November 2015].

Magento, 2015b. *Open Source Ecommerce Software & Solutions*. [Online]  
Available at: <https://www.magentocommerce.com/download>  
[Accessed November 2015].

Magento, 2015c. *Magento License / Trademarks FAQs*. [Online]  
Available at: <http://magento.com/legal/licensing>  
[Accessed November 2015].

Mirdha, A., Jain , A. & Shah, K., 2014. *Comparative Analysis of Open Source Content Management Systems*. s.l., 2014 IEEE International Conference on "Comparative analysis of open source content management systems," in Computational Intelligence and Computing Research (ICCIC), pp. 1-4.

Olsson, H. H., Alahyari, H. & Bosch, J., 2012. Climbing the "Stairway to Heaven" -- A Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. in *Software Engineering and Advanced*

*Applications (SEAA), 2012 38th EUROMICRO Conference on, 5-8 September, pp. 392-399.*

Open Source Matters, 2015a. *About Joomla!*. [Online]

Available at: <https://www.joomla.org/about-joomla>

[Accessed November 2015].

Open Source Matters, 2015b. *About the Joomla Project*. [Online]

Available at: <https://www.joomla.org/about-joomla/the-project.html>

[Accessed November 2015].

Patel, S. K., Rahod, V. R. & Parikh, S., 2011a. *Joomla, Drupal and WordPress - a statistical comparison of open source CMS*. s.l., 2011 3rd International Conference on in Trendz in Information Sciences and Computing (TISC), pp. 182-187.

Patel, S. K., Rathod, V. R. & Prajapati, J. B., 2011b. Performance Analysis of Content Management Systems- Joomla, Drupal and WordPress. *International Journal of Computer Applications (0975 – 8887)*, May.

Paybyway, 2015. *Paybyway - Documentation*. [Online]

Available at: [https://www.paybyway.com/docs/web\\_payments/?page=plugins](https://www.paybyway.com/docs/web_payments/?page=plugins)

[Accessed November 2015].

Pelkkikangas, O., 2010. *Verkkokauppa-alustojen vertailu ja asiakkaan tarpeiden kartoitus*, s.l.: Jyväskylän ammattikorkeakoulu.

PrestaShop, 2015a. *PrestaShop - Free ecommerce software*. [Online]

Available at: <https://www.prestashop.com/>

[Accessed November 2015].

PrestaShop, 2015b. *FAQ - PrestaShop 1.6 - PrestaShop documentation*. [Online]

Available at: <http://doc.prestashop.com/display/PS16/FAQ>

[Accessed November 2015].

Pulse247, 2015a. *MyCashflow – Helppokäyttöinen verkkokauppaohjelmisto*. [Online]  
Available at: <https://www.mycashflow.fi/>  
[Accessed November 2015].

Pulse247, 2015b. *Perusta verkkokauppa*. [Online]  
Available at: <https://www.mycashflow.fi/perusta-verkkokauppa/>  
[Accessed November 2015].

Seadle, M., 2006. Content management systems. *Library Hi Tech*, 24(1), pp. 5-7.

Soukka, A., n.d.. *ruokapiiri - Revision 92*. [Online]  
Available at: <http://ruokapiiri.googlecode.com/svn/trunk/>  
[Accessed November 2015].

Suomen Valmispalvelut, 2015. *Valmiskauppa.fi - verkkokauppa helposti, nopeasti ja edullisesti!*. [Online]  
Available at: <http://valmiskauppa.fi/etusivu/>  
[Accessed November 2015].

W3Techs, 2015. *Usage Statistics and Market Share of Content Management Systems for Websites, November 2015*. [Online]  
Available at: [http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all)  
[Accessed November 2015].

Wen, H. J., Chen, H.-G. & Hwang, H.-G., 2001. E-commerce Web site design: strategies and models. *Information Management & Computer Security*, 9(1), pp. 5-12.

Wiklund, A., 2015. *Building a webshop using Drupal Commerce with Commerce Kickstart as a Template*, s.l.: Arcada University of Applied Sciences.

Vilkas Group, 2015. *Verkkokaupparatkaisut yrityksille | Vilkas Group*. [Online]  
Available at: <http://www.vilkas.fi/>  
[Accessed November 2015].

WordPress, 2003. *WordPress Now Available*. [Online]  
Available at: <https://wordpress.org/news/2003/05/wordpress-now-available/>  
[Accessed November 2015].

WordPress, 2015a. *The WordPress License*. [Online]  
Available at: <https://wordpress.org/about/license/>  
[Accessed November 2015].

WordPress, 2015b. *WordPress*. [Online]  
Available at: <https://wordpress.org/about/features/>  
[Accessed November 2015].

WordPress, 2015c. *WordPress Codex*. [Online]  
Available at: [https://codex.wordpress.org/Main\\_Page](https://codex.wordpress.org/Main_Page)  
[Accessed November 2015].

WordPress, 2015d. *Requirements*. [Online]  
Available at: <https://wordpress.org/about/requirements/>  
[Accessed November 2015].

WordPress, 2015e. *WordPress Plugins*. [Online]  
Available at: <https://wordpress.org/plugins/>  
[Accessed November 2015].

WordPress, 2016a. *Updating WordPress*. [Online]  
Available at: [https://codex.wordpress.org/Updating\\_WordPress](https://codex.wordpress.org/Updating_WordPress)  
[Accessed 2016].

WordPress, 2016b. *Class Reference/wpdb*. [Online]  
Available at: [https://codex.wordpress.org/Class\\_Reference/wpdb](https://codex.wordpress.org/Class_Reference/wpdb)  
[Accessed 2016].