Bachelor Project

Experiences from the Development of a Webshop Using Scrum Methodology

by

Tim Andersson, Gustav Arnesson, Pontus Brengdahl, Anna Ekelund, Claes Kallström, Kalle Olsson, Julia Thudén, Erik Wallvik

LIU-IDA/LITH-EX-G--14/028—SE

2014-05-29
Bachelor Project

Experiences from the Development of a Webshop Using Scrum Methodology

by

Tim Andersson, Gustav Arnesson, Pontus Brengdahl, Anna Ekelund, Claes Kallström, Kalle Olsson, Julia Thudén, Erik Wallvik

LIU-IDA/LITH-EX-G--14/028—SE

2014-05-29

Advisor: Rebecka Geijer Michaeli
Examiner: Aseel Berglund
Abstract

This is a report concerning the software development project Eldflugan, a system developed by eight students at Linköping University. It addresses experience from developing a web-based e-commerce using Scrum, an agile development method, which was used throughout the project. The teams work, how Scrum was applied with both advantages and disadvantages is being lifted. Different development environments have been used due to separate issues and opportunities, which is described. The database management systems, MySQL, and the local version, SQLite, are discussed as well as AJAX and PJAX, tools for giving the users a better and faster experience. To prepare Eldflugan for entering the e-commerce market, a marketing plan including environmental scanning, a SWOT analysis and a STP, has been made. Ethical aspects such as legal, use and handling of personal information and how it is communicated are also being addressed. Finally the report discusses product weaknesses, future opportunities, implementation difficulties and how those could have been prevented. It was found that Scrum is best used when it is utilized fully and a risk analysis can prevent unexpected problems to surface.
# Table of Contents

1 Introduction ..................................................................................................................... 1

2 Method .............................................................................................................................. 2

  2.1 Scrum .......................................................................................................................... 2

  2.2 The Development Process ......................................................................................... 6

  2.3 Development Environment ....................................................................................... 8

3 System Overview ............................................................................................................. 10

  3.1 Product Backlog ......................................................................................................... 10

  3.2 Technical Description ............................................................................................... 11

4 System Specification ....................................................................................................... 20

  4.1 Database ................................................................................................................... 20

  4.2 Graphical User Interface ......................................................................................... 21

  4.3 Logic .......................................................................................................................... 28

  4.4 Refactoring ............................................................................................................... 30

5 Marketing Plan ................................................................................................................ 32

  5.1 Background ............................................................................................................... 32

  5.2 Environmental Scanning ......................................................................................... 32

  5.3 SWOT ......................................................................................................................... 35

  5.4 TOWS ......................................................................................................................... 37

  5.5 STP ............................................................................................................................. 38

  5.6 Marketing Mix ........................................................................................................... 40

6 Ethical Aspects ................................................................................................................ 41

  6.1 Legal Aspects ............................................................................................................ 41

  6.2 Handle User Information ......................................................................................... 41

  6.3 Communicating Ethical Strategy ............................................................................. 42

  6.4 Use of Personal Information for Marketing Purposes ........................................... 42

7 Summary ......................................................................................................................... 43

  7.1 Discussion ................................................................................................................. 43

  7.2 Conclusion ................................................................................................................ 46

References ......................................................................................................................... 48

Appendices ......................................................................................................................... 51
1 Introduction

During the last ten years Swedish e-commerce market has grown by 665 %, proving web-based trading is a very promising market to enter. Eight students at Linköping University developed a web-based e-commerce system for the fictitious company Eldflugan that has two physical stores. This was done during the spring of 2014 using Scrum methodology and advanced python programming. They developed a product with the goal of being user friendly and intuitive. Style and mobile accessibility were also desired features. The ability to complete a purchase quickly, especially if you had used the service before, was an obvious criterion. These product goals formed the basis for the vision developed by the team.

“Eldflugan will make it easier for price sensitive and interior interested individuals to find lamps and accessories in a large product range. They will be able to locate desired products through great search and filtering possibilities. Eldflugan offers cheap and stylish lamps through a user friendly website which is designed to get customers to complete a purchase both swiftly and easily.”

- Eldflugan’s Vision

This report is a description of the team’s development process, technical descriptions and motivations behind the teams marketing decisions. The development environment and ethical aspects are also being discussed.
2 Method

This section describes how the work of the project was completed, including a description of the Scrum method, how it was applied on this project, the development process and the development environment.

2.1 Scrum

The agile software development method Scrum is described, both the background of Scrum and how it theoretically should be applied on a software development project, and how it was applied on the project discussed in this report.

2.1.1 Background

The word Scrum has its origins in the Rugby Union, where it describes the team’s aim of moving forward as a unit by passing the ball back and forth (Takeuchi & Nonaka, 1986). In 1980, the term was adopted as a strategy name used for managing complex projects in web-based software development companies (Pope-Ruark, 2012; Hallin & Gustavsson, 2012). Today it is one of the most common agile methods (Hallin & Gustavsson, 2012), where describing key words are transparency, inspection and adaptation. These keywords describe the importance of the entire Scrum process being visible to all responsible for the outcome (described in chapter 2.1.2.2 Scrum Team), detecting undesirable variances and adjusting those as soon as possible. According to the founders Scrum is: “Lightweight, simple to understand and difficult to master”. (Sutherland & Schwaber, 2013)

The agile methods all have their base in The Agile Manifesto defined by a group of independent thinkers called The Agile Alliance. The aim of agile methods is to develop software efficiently and quickly while having close interaction and good communication between the developers and the customers (Bose, 2008). According to The Manifesto for Agile Software Development, things to value during an agile development process are:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan,

This has been adapted by Scrum. (Sutherland & Schwaber, 2001; Sutherland & Schwaber, 2013)

2.1.2 Definition of Scrum

Ken Schwaber and Jeff Sutherland developed the Scrum method in the early 1990’s (Sutherland & Schwaber, 2013), where the idea was to repeat development cycles until the product is complete (Pope-Ruark, 2012). These cycles are called sprints and contain some specific activities, which will be described later in this report. Scrum artifacts, such as the product and sprint backlog, represents value or work to make the project transparent, and that will also be addressed.

Scrum is not meant to be a tool for the technology behind the product development, but rather a process framework for managing the development (Sutherland & Schwaber, 2013); it is a project managing method (Hallin & Gustavsson, 2012).
2.1.2.1 Product Backlog

All Scrum processes are built around an ordered To-Do list called the product backlog. It contains all the requirements, functions, features and enhancements to be made to the product. The product backlog is put together by the team in the beginning of the process but is constantly changing as the project itself is changing. The items in the backlog are prioritized after importance for the product and the project. (Sutherland & Schwaber, 2013) The items in the product backlog are commonly referred to as user stories. A user story is a feature or function written in, as the name implies, a user-centric perspective and is written in the form of a story. A common pattern for a user story could look as follows: ‘As a <role> I want to <action> so that <result>’. A user story will sometimes contain criteria called acceptance criteria that describe different actions and what the result should be since an item may sometimes need to manage several different actions. The user story cannot be considered done unless it satisfies these acceptance criteria. The advantage of using user stories in an agile development process is to encourage a rich dialog between stakeholders and developers. It gives the stakeholder a better understanding of what is done while giving the developers more flexibility as well as avoiding premature specifications of the item’s implementation. (O’hEocha & Conboy, 2011)

2.1.2.2 Scrum Team

Every Scrum project team consists of a Scrum master, a product owner and a development team. The product owner is the one who is responsible for the product backlog. This includes making sure the items are clearly expressed and in order. He or she also has to make sure the backlog is transparent, understandable and visible to everyone involved. (Sutherland & Schwaber, 2013) The product owner often represents the customer or final user (Hallin & Gustavsson, 2012).

The development team is a group of three to nine people, working on creating the actual product. The development team is self-organized and cross-functional. This means that the team themselves decide how they will best work to develop the product. Being cross-functional means the team members have all the necessary skills to create the product and that no outside consultation is needed. The members of the development team have no special titles or roles. The focus is on what the team can perform, not the individual developer.

The last role in a Scrum team is the Scrum master. The Scrum master’s most important responsibilities are to plan and carry out the Scrum events, to make sure everyone adopts the Scrum method and to ensure everyone involved understands the theory and practice. The Scrum master is also the link between the Scrum team, the product owner and the organization or customer. If there is more than one development team working on the product, it is the Scrum master’s job to coordinate with the other groups. (Sutherland & Schwaber, 2013)

2.1.2.3 Scrum Events

In Scrum there are a number of events that should be carried out to achieve the best possible outcome. Common for all events are that they are time-boxed, meaning they have a maximum duration. The longest Scrum event is called a sprint, and has time duration of maximum one month. All other Scrum events are a part of a sprint. (Sutherland & Schwaber, 2013)
In the beginning of a process you do not know how many sprints will be needed to complete the project. However, the duration of each sprint is always decided in advance. (Hallin & Gustavsson, 2012)

Each sprint begins with a sprint planning where the entire Scrum team attends. During the planning the team decides on a sprint backlog, a sprint goal and how the sprint goal will be achieved. In other words, how the work will be done. (Sutherland & Schwaber, 2013)

The sprint backlog consists of the product backlog items with the highest priority, the items to be achieved during the sprint (Hallin & Gustavsson, 2012). After the team has decided on the sprint backlog the sprint goal is set. It sums up all the items in the sprint backlog and is supposed to be an objective for the team to make them work towards a common goal rather than on separate items.

To keep the Scrum project transparent and the development team up to speed, each day in the process begins with a daily Scrum meeting. It is a 15-minute time boxed event in which the entire development team and the Scrum master participate. (Sutherland & Schwaber, 2013) During the meeting everyone in the team answers the following questions:

- What did I do yesterday?
- What will I do today?
- Do I see any challenges that I might need help with or that prevent the team from reaching the sprint goal?

It is very important that everyone speaks and that everyone gets his or her time during the daily Scrum meeting. Discussion and additional comments should be reserved for later. (Sutherland & Schwaber, 2013; Pope-Ruark, 2012) The Daily Scrum meeting is a tool for quickly detecting obstacles in the project and solving them quickly. It also reduces the amount of other meetings, which saves time for the project.

At the end of every sprint a sprint review is held. It is an informal meeting that the Scrum team and the key stakeholders attend. The motive of the sprint review is for the team to show their customers what they have done during the previous sprint and to get their input on what to do next. The outcome of a sprint review is usually a preliminary sprint backlog for the next sprint, and some valuable input to take to the next sprint planning.

After the sprint review the Scrum team holds a sprint retrospective. The retrospective is a chance for the team to inspect themselves and plan improvements for the next sprint. The aim is for the team to develop and keep improving their work. It also increases the Scrum focus on inspection and adaptation. (Sutherland & Schwaber, 2013)

2.1.3 How Scrum Was Applied in This Project

During the course “TDDD83 Computer Engineering – Bachelor Project”, given at Linköping University, where a web-based e-commerce system was developed, Scrum was used as project managing tool. Since the objective for this project was for students to learn, no real customer was involved which made this project differ a bit from ordinary Scrum development. However, as much Scrum practice as possible was used and the following text describes how the team developing Eldflugan applied Scrum to their work.
2.1.3.1 Scrum Team

The team developing the product consisted of eight people, which formed the development team. In the beginning of the project one of them was chosen to be the Scrum master. Seeing that no actual customer was involved in this project, the role of product owner was not appointed. Instead the development team created the product backlog in the beginning of the process. The team also took upon themselves to make sure the backlog fulfilled all necessary requirements, such as being transparent and visible to all, which normally is the product owner’s responsibility.

The development team at Eldflugan fulfilled the requirements of being both cross-functional and self-organized. Except for the Scrum master no one had any official titles. The team was allowed to carry out their work, as they found suitable, as long as they kept to the Scrum practices, which they did.

There was only one development team working on the product backlog, which decreased the work for the Scrum master, as no coordination with other development teams was required. However, the Scrum master was responsible for all the Scrum events the group held (See 2.1.3.3 for more details about the teams Scrum events).

2.1.3.2 Product Backlog

The product backlog consisted of user stories describing features and functions the team found necessary and desirable for the product. Depending on importance, the team prioritized among the stories after adding them to the backlog.

To keep track of the status of each story, the web-based collaboration tool trello was used. In trello the user stories were written on “cards” that could be moved between different lists depending on their development state. Names of the lists used were: Project backlog, Product backlog, Sprint-X backlog, In progress, Doing, Awaiting testing, Testing, Tested, Done and Blocked. While working on a story, the team members signed up on associated cards to enable other team members to keep track of the other team members’ work. This increased project visibility and transparency.

2.1.3.3 Scrum Events

In this project, opposite to what is said in official Scrum practice, the number of sprints to carry out was decided when the project began because the project had a limited time frame. The number of sprints chosen was four, with different objectives for each one (for details about sprint objectives see chapter 2.2 The Development Process).

Each sprint started with a sprint planning where, according to Scrum practices, the team decided on a sprint backlog and how best to carry out the work to reach the sprint goal.

Since none of the team members were working on the project full time it was decided that the daily Scrum meetings only would be held three times a week; Monday, Wednesday and Friday. This was thought to correspond to a full time project having daily Scrum meetings every day. During those meetings all team members where standing up while they, one at the time, said what they had done since the last meeting and what they where going to do next. If any problems have occurred those where addressed too.

The definition of done, i.e. how to see if a user story is done, used in the project worked as follows. When the functionality explained in a user story had been developed and tested according to the
acceptance criteria by the developer, a team member that did not develop the function tested it. If the team member judged that it fulfilled the acceptance criteria it also was considered done and moved to the list Tested. During the following sprint meeting, when the whole team was present, it was moved to the list Done in the trello board if the whole team agreed on it being done.

As previously mentioned, no real customer existed in this project, which removed the need of a sprint review. However, a sprint presentation, where the team presented their work to the course examiners and fellow students, was held at the end of each sprint. After the sprint review the team held a sprint retrospective where they reviewed their work and discussed how to best improve. An example of a thing that was addressed during the retrospective was how the team held their meetings. In the beginning of the project the team members where not thorough enough during the daily Scrum, but after addressing it during a retrospective this was changed, which lead to more giving Scrum meetings. A full summary of the results from the retrospectives is found in Appendix D: Retrospective.

2.2 The Development Process

The objective of the software development process was to learn the scrum methodology while developing a web-based e-commerce system. The development process was divided into four sprints; sprint zero, sprint one, sprint two and sprint three. The team has been working individually with the support from an advisor.

2.2.1 Sprint Zero

Sprint zero was an initial phase with the objective to start up the development process and to let the team get to know each other. The sprint consisted of several steps. To be able to work effectively as a group throughout the software development process, the first step was for team members to get to know each other by presenting each other’s journey lines. The team also held social activities with the same purpose.

The next step was to develop a project plan consisting of group contract, communication plan, project organization, milestones, vision for the product and objectives of the project (see Appendix F). In the group contract details such as timekeeping and schedule of Scrum meetings were included. The communication plan decided what communication tools to use, i.e. how to communicate through email, text messages and physical meetings. The project organization described who the Scrum master was and which member of the team that was responsible for different parts of the software development process. The objective of the project consisted of objectives for the product, as well as for the team and each team member. Personal goals for the team members were to learn Scrum and improve the knowledge of agile development, to improve web-programming skills and learn the development environment and tools required for the project, and to be able to create other web-applications in the future.
2.2.1.1 Concept Development Process

Based on the vision for the product a prototype was developed in several steps. The first step was brainwriting, a method where each person in the group write down ideas on a sheet of paper, and then the paper is passed to the next person, who develop the ideas of the first person, or come up with completely new ones. The papers are sent to a new person a couple of times. The advantage of brainwriting is that it usually yields more ideas in less time than brainstorming would have done. (Wilson, 2013) This resulted in all functions for the product that the team could come up with, with the vision in mind, being written down.

The second step was to do a function analysis, where the ideas collected in the first step were ranked to be required, desirable or unnecessary by the team. This was done to be able to decide what functions were prioritized for the first iteration of the development process. The functional analysis worked as a base for the product backlog.

The third step was to do a concept divergence, where each member of the team picked one of the required functions and made a sketch of the e-commerce system, letting the picked function be the head function. After that, as a fourth step, a concept evaluation was made, where all team members presented their idea and the team agreed on one concept to visualize with a prototype. The fifth step was to make the prototype. After completing the prototype it was tested on several potential users and updated considering the feedback given by the potential user. For some of the feedback collected, see Appendix F. Making a prototype enables the team to make small changes and correct errors before everything are fully resolved, which leads to faster organizational learning (Coughlan, Fulton Suri & Canales, 2007).

2.2.1.2 Planning the Product Backlog

After completing the prototype a product backlog and user stories were created. The product backlog consisted of all functions required and desirable for the product. The required functions were prioritized and placed in the backlog for the first software development sprint. Time estimation was done through using Planning Poker. Planning Poker is a tool for estimating the time it will take to complete the development of a story. It is a way to get consensus of how long it will take to develop, avoiding discussion. Each player pick a card with a given time estimation, show it to the rest of the group and then an average is calculated and used as an estimation. (Grenning, 2002)

2.2.2 Sprint One to Sprint Three

After completing the initial phase, the following three sprints were dedicated to develop the product. It was done following the Scrum methodology, iterating through the three sprints. After the first sprint simple functionality had been implemented, after the second sprint all the functionality was finished and during the third sprint no more functionality was added. Instead the team was working with
refactoring. Refactoring is to reconstruct an existing body of code, without changing the external behavior of the code, and is done during the development process (McDonnell, 2013). During the development of Eldflugan, the refactoring was not done during the development process, but afterwards. The purpose of refactoring is to improve maintainability, quality and clarity of the code (McDonnell, 2013).

Most of the development process consisted of the team sitting together, but much of the actual work was done individually.

2.2.2.1 Testing

During the software development process all created functionality was tested. To make sure it worked well together with the rest of the application it was tested frequently during development. When a team member had completed developing functionality, the tool trello was used to move the story describing the functionality to the list Awaiting Testing. When the story had been placed in Awaiting Testing one of the other team members was assigned to test the functionality. If it passed the test the story was placed in Tested, and if it did not pass a comment on what went wrong was left, and the item was returned to a member of the team to solve the problem. During testing the item was tested for whether it matched the story and its acceptance criteria, and if it worked well with other parts of the application.

Another part of the testing was the testing done on potential users of the product. Throughout the development process the development team asked potential users to test the functionality on the website. The feedback given from the users was discussed in the team and if the team decided to change the functionality to match the expectations of the customers, it was added as a user story to the trello.

2.3 Development Environment

The section about development environment includes a description of the development environment, where the project was hosted and the graphical framework used in the project.

2.3.1 Integrated Development Environment

During the development process the team used an IDE (Integrated Development Environment) called PyCharm, which has syntax highlighting, code completion, debug-mode and other useful features for development in Python. PyCharm works on multiple operating systems, namely Windows, OSX and Linux. It has support for the micro framework Flask and the programming languages Python, JavaScript, HTML and CSS. PyCharm also have the option to run the code locally, creating a local web application that operated the way it would on the web server. This allowed the team members to test changes before they got pushed to the git repository.

2.3.2 Hosting

In the beginning of the development, a cloud-based hosting service called OpenShift was chosen for the developers to use. On OpenShift a git repository, for version control, was hosted and when the developers pushed their changes to OpenShift a deployment script made the updated version available on the web server.
During the development process an alternative platform was set up, because of difficulties with OpenShift. The alternative platform only hosted the web server, and the git repository was moved to GitHub. A deployment script was also set up and it was called by GitHub on push.

Some gains of working with a third party platform like OpenShift is that configuration and support also can be outsourced to the service provider and the team can focus on development. On the other hand, the code is not primarily stored by the developers and they become more dependent on an external factor, which cannot be controlled. This is something that could be avoided on the alternative platform, which had to be configured and kept the code closer to the developers.

When the code was moved to GitHub and the alternative platform was set up, another git branch was created. After that point there were two versions of the code: one branch to keep the latest working main version and one branch to keep the code with newly developed functions. By separating the code in this way the developers were able to have two functioning websites, one for development and one for production. This was not possible with OpenShift and was an argument against OpenShift. At the end of the project the developers had to move the code back to OpenShift because of external requirements and then there only was one version of the website again.

### 2.3.3 Graphical Frameworks

To help make the application responsive and scalable to different screen sizes and to speed up the development process the front-end web development framework Bootstrap has been used. Bootstrap was created by a designer and developer at Twitter and initially served as a style guide for their internal tools development, and has since its release to the public become one of the most popular front-end frameworks and open source projects in the world. With its column system, Bootstrap has allowed the developers to place objects on the page and to change content depending on what device the site was shown on. It also has built in solutions for commonly used elements, e.g. dropdown menus and carousels for showing promotion pictures. (Bootstrap, 2014)

Using this graphical framework also gave the webpage a uniform look despite being made by a team with varying knowledge of web design.

### 2.3.4 Continuous Deployment and Integration

During development the team worked on the same branch of the code. The developers continuously merged the updated code from the git repository with the code they had on their local machine. When a team member had added a new functionality the code was pushed to the git repository where the deployment script would make it go live on the web server, at this stage the updated code was available to the other team members. This way the team could continuously merge their own code with the newest version available as well as making the changes live when new code was pushed to the git repository.

Early in the development process the team standardized how to use the tools given for continuous deployment and integration. Before a team member was allowed to push a change to the git repository the team member had to test the code locally. If it worked the team member could pull down and merge their code with the live version. After the merge the code had to go through another round of testing before it could be pushed to the git repository. This routine was set to prevent faulty code from going live and potentially causing the web application to go down.
3 System Overview

The system overview includes a description of the product backlog, and a technical description of the current functionality of the web application.

3.1 Product Backlog

This is a description of the product backlog. All features contained within the product backlog can be overviewed in Appendix A: User Stories.

3.1.1 Features

Before the website development started, the product development team was tasked to map out concepts for software features that the customer would find useful and desirable. The developers had two clients; the product owner and the broader targeted customer group. The functions required by these customers branched out into two separate types of functions; administrative and user type functions. Administrative functions give the product owner the possibility to dynamically change the content on the web, while user functions are either desirable or necessary for any potential customers. For instance, administrative functions could include adding and removing products from the web page. Administrative functions work as an interface for the product owner who might not have any prior knowledge of programming, but still need to manipulate data presented by the site. Contradictory to the administrative functions the user type functions are meant for the common users of the web page and not meant for the product owner. These functions are then split into several categories, like my pages, products and shopping cart.

During the creation of the concept the functionality and design of some existing interior design and lamp websites was analyzed and used as an inspiration. For example, the functionality of the filter was inspired by the price comparison engine prisjakt.nu, the dual navigation bars (see Figure E.2 in Appendix E) from the interior design site rum21.se and the product grid from the lamp site lampan.se.

When the features had been mapped out the team started working on a prototype of the website which is shown in Appendix E: Prototype.

3.1.2 Bug Fixes

After sprint one several administrative and user type functions were finished, yet in most cases the new functions were not completely bug free. Thus, the product backlog grew to containing items in order to deal with these issues, referred to under the tag BUG in the product backlog.

3.1.2 Non-functional Requirements

Non-functional requirements or the qualities of the system are divided into two main categories; performance qualities, such as security, or evolution qualities, such as testability, maintainability, extensibility and scalability. Though the backlog has no specific tags noted non-functional qualities, it was often implied in several of the items in the backlog.
3.2 Technical Description

What follows is a full recount of the project's current functionality, much like a manual. User story descriptions can be found in Appendix A: Product Backlog, and will be referred to in the text.

3.2.1 User Type Functions

User type functions are the functions utilized by the end user, some available to registered members and some to everybody.

3.2.1.1 Homepage

Upon arrival at the site, the customer is greeted by a carousel containing promotion for various products (Figure 3.1, notation 5; Appendix A, user story O3). Below is a grid containing promotions or offers, each containing a blue button to redirect any potential customer towards what they would like to see (Figure 3.1, notation 6; Appendix A, user stories O2-3 and A1-4). All of these can be edited by administrative functions (as described in Appendix A, user stories A3-4). In the top left corner of the screen, there is a login link and contact link (Figure 3.1, notation 1; Appendix A, user story Mp2). Each of these links reloads the main content of the page to access these functions respectively.

At the top right corner of the screen are one button and one link (Figure 3.1, notations 3 and 4). Notation three contains a dropdown menu showing all items having been selected for purchase (Appendix A, user stories Sc1-5). A recount of the number of products currently in shopping cart can be seen and by clicking on the dropdown menu a calculated cost will appear (Appendix A, user stories Sc1 and Sc5). This along with a green button redirecting the user to the checkout counter and a small blue button permitting to store a shopping cart (Appendix A, user stories Sc1 and Sc4). In the footer there are links regarding contact information, about Eldflugan, how to find the stores and social media (Appendix A, user story O5). Each of the links redirects the user to a page containing information regarding the suggested topic. Below the logo, on the left side of the screen, there are links to the multi product pages containing either indoor-/outdoor products or accessories (Figure 3.1, notation 7). These are each possible dropdown menus, which can redirect the user to view all products having that criterion (Appendix A, user stories P2-3). Beside the indoor link is a small house linking back to the
home page. To the right of the accessories link the user can utilize a quick search function (Figure 3.1, notation 8; Appendix A, user stories S1-4). By typing in a minimum of three letters long string, the user can quickly find any product containing those letters in its name or certain attributes. The products matching the query will appear in a dropdown menu just below with a purchase link to the right and a picture, the name and the price to the right (Appendix A, user story S3).

The functions seen in Figure 3.2 below remain present at all times as long as the user remains at the webpage (Appendix A, user stories O2 and O6).

Figure 3.2: Navbar

In addition so does the information in the footer mentioned above.

3.2.1.2 Login

Figure 3.3: Login view

After having pressed the login link in the top left corner of the screen the user will be redirected to what can be referred to as the login page. Here the user can use the input forms (Figure 3.3, notation 2; Appendix A, user story Mp2) to login to an existing user account or click the register button to register a new account (Figure 3.3, notation 1; Appendix A, user story Mp5). If the user for any reason has forgotten their login information they may have a new password sent out to the email they are using by pressing the Forgot password button and then filling in their registered email address (Figure 3.3, notation 4; Appendix A, user story Mp9). Logging in or registering will not empty the current shopping cart so if the user registers or logs on after items have been added to the cart the products remain in the cart and the user can continue the purchase as a logged in user.
3.2.1.3 My Pages

Upon logging in the user will enter what is called *My Pages*. From here the user can view saved shopping carts, previous orders and delivery addresses (Figure 3.4, notations 2 and 3; Appendix A, user stories Mp4 and Mp7). The user can also change his password, email and/or name (Figure 3.4, notations 4 and 5; Appendix A, user story Mp10). When logged in the login link in the upper left corner is exchanged for a dropdown menu. Containing a link to logout the user or redirect him or her back to his or her pages (Figure 3.4, notation 1; Appendix A, user story Mp1). The user can also add various delivery addresses, set a default shipping address or edit an existing delivery address (Figure 3.4, notations 6, 7 and 8; Appendix A, user stories Mp8, Mp10 and O2).
3.2.1.4 Multi Product Page

By clicking on either the indoor or the outdoor links, arguably even the accessories link, the user will be redirected to the multi product page (Figure 3.5). From here the user can view all products upholding the set criteria selected (Appendix A, user story P2). Further criteria can be used to filter the products by clicking any of the checkboxes and moving the price slider before clicking the green filter button (Figure 3.5, notation 2; Appendix A, user story P3). Using the reset button, the checked checkboxes can be reset (Figure 3.5, notation 2; Appendix A, user story P3 and O2). By clicking on the breadcrumbs in the upper left side of the screen the parameters of products shown can be changed (Figure 3.5, notation 1; Appendix A, user story Mp2 and Mp3). Above the product grid to the right, and below the price slider, there are two buttons allowing the user to change the way that the products are viewed into either list view or grid view (Figure 3.5, notation 3; Appendix A, user story P4). By clicking on any product the user is redirected to the single product page (Figure 3.5, function 4; Appendix A, user story P8) and by clicking on the green button at the bottom right corner of the product caption, an item can be added to the shopping cart (Figure 3.5, notation 5; Appendix A, user stories Sc2-5).
3.2.1.5 Single Product Page

By clicking on a product through the quick search or the multi product grid or possibly through a promotion at the homepage, the user will be redirected to the single product page (Appendix A, user stories P8 and O2). From here a large picture of the product which on click will show a larger image (Appendix A, user stories P7-8) can be seen to the left and product information and inventory status to the right (Figure 3.6, notations 5 and 4; Appendix A, user story P8). Below the inventory status the user can type in the number of items they wish to buy and then press the green purchase button (Figure 3.6, notation 3; Appendix A, user stories Sc5 and P5). If nothing is typed in the default setting is one thus allowing the customer to add a single item to their shopping cart should they press the purchase button (Figure 3.6, notation 3; Appendix A, user stories Sc1-5 and P2-8). Below this a logged in user can leave a rating on a product or change an existing one (Figure 3.6, notation 2; Appendix A, user story P5). A user who in turn is not logged in can see the current average rating displayed on the stars but cannot directly interact with it. In the far right bottom of the screen is a discussion forum accessible for any user logged in or not (Figure 3.6, notation 1; Appendix A, user story P1 and P5).

3.2.2 Admin Functions

As explained earlier the product owner will have access to all admin type functions. These functions are accessed by logging in with a certain username and password, either by clicking the login button in the upper left corner of the screen and typing the correct information into the form, or by adding “/admin” to the URL and submitting the correct information to the form. If correct information is entered the user will be redirected to the admin view. From there the administrator have the ability to manipulate the content of the website. After entering the admin view the administrator will have two links below a navigational bar. Clicking the direct links will either redirect the user to the start page or...
recreate the original database structure. The text will ensure that it is obvious which link does what. At
the top of the page there is a menu that shown in Figure 3.7 below.

Figure 3.7: Admin menu

Each button allows the administrator access to functions related to the text. (Appendix A, user stories A1-4)

3.2.2.1 Product Management

By clicking the button called Produkthantering a dropdown menu appears which allows the user to
either directly add a product or overview all existing products in the database. Just below the
administrator menu four tabs have appeared, the one active by default showing how many products are
currently in the database, and the other tabs are Create, Add Filter and With Selected. (Figure 3.8,
notations 1,2,3 and 5) Below the tabs all products and their attributes will appear in a list (Figure 3.8,
notation 7), each headed column signifying certain attributes the product has. The remaining two
columns being functions, the second column containing the image of a pencil and a trashcan allows for
manipulation of the specific product (Figure 3.8, notations 4 and 6; Appendix A, user stories A1-4). By
clicking the trashcan the product on the same row can be deleted and by clicking the pencil the product
can be modified, that is any attribute can be changed (Appendix A, user stories A3-4). If the user
wishes to remove a product they can also check the checkbox in the first column and then click the
With Selected tab on the top left below the administrators menu and then click delete in the dropdown
menu that appears. (Figure 3.8, notations 6 and 5)

Figure 3.9: Admin Filtering

By clicking the Add Filter dropdown menu among the tabs the administrator can sort the product
overview by a combination of criteria. When the user clicks Add Filter he or she can filter by the
following product attributes: Name, Brand or Lamp Type. Clicking any of these will cause a textbox,
an extra dropdown menu and two buttons to appear below the tabs but above the product overview
(Figure 3.9). Clicking the grey button marked with whatever was selected from the dropdown menu to
the far left will remove this interface. In the dropdown menu four categories can be selected equals,
not equals, contains and not contains as the text suggests it will determine how the filter will use the text written in the textbox (Figure 3.9). So for instance applying a filter with not equals and “Lux” having previously pressed Name will filter away all products with the name “Lux”. The filter is applied either by clicking the blue Apply button to the right or by pressing enter on your keyboard. (Appendix A, user stories O2, A3-4 and P3) When the filter has been applied the product overview list will become shorter allowing the user to find what they are looking for. When a filter is applied the user will also note that an extra button has appeared alongside the Apply button or if the Add Filter is not being used by itself to the right. This grey button named Reset filter will allow the user to reset the filter set up by the user. Several filters can be active at the same time on multiple categories allowing the administrator to filter using and criteria i.e. filter names that equal “Lux” and “Carpatica”. If the reset filter button is pressed all filters will be removed, not just the latest. (Appendix A, user stories O2, A3-4 and P3)

![Figure 3.10: Adding product](image)

By clicking the Add product button in the dropdown a form will appear allowing the admin to fill in information for any potential new product (Figure 3.10). Each input have the attribute description next to it making it easy to understand what to write where. Upon completion the user can press the Submit button to complete the procedure. (Appendix A, user stories A3 and A4)

### 3.2.2.2 Frontpage

![Figure 3.11: Promotion management](image)

By clicking the Förstasidan button on the admin menu the administrator can handle what promotions are made to the customer upon first entering the site. Upon clicking the button Förstasidan the user can select either Carousel or Promo (Figure 3.7) both linked to the homepage. Similar to the Product
Management page the default view is a list containing all saved promotions regardless of which of the option is clicked. There are also tabs in the top left just below the administrator menu as well as the list containing attributes of promotions, which is different from that of the products. The first two columns are exactly the same in functionality and look as the Product Management, allowing the user to delete or edit a row. The middle tab carrying the name Create has a similar functionality as that of its Product Management counterpart. Upon clicking opens a form in which information regarding a new promotion can be added. Due to text written to the left of the textboxes it should be fairly obvious what should be inputted where in order to create a new promotion. (Figure 3.11; Appendix A, user stories A3 and O1)

3.2.2.3 Inventory

Figure 3.12: Inventory management

Much like its two predecessors the inventory’s default view shows a list of all existing products and just like with its predecessors the first two columns allow the administrator to edit or delete data (Figure 3.12, notations 2, 3 and 4). Specifically how large of a supply there is of the certain item (Figure 3.12, notations 6 and 2). (Appendix A, user story A3)

3.2.2.4 Orders

Figure 3.13: Order management

In the administrative view of orders current orders are viewed in a list (Figure 3.13, notation 1; Appendix A, user story A4). When an order has been processed the admin can click the cogwheel belonging to the correct order number and change the orders status (Figure 3.13, notation 2; Appendix A, user story A4). This way a customer can follow where their products are. Changing an order will also trigger in sending an email to notify the customer of the change (Appendix A, user story O4).
3.2.2.5 Upload File

Since pictures for lamps must exist on the server, image files can be uploaded onto the server and ordered in directories (Figure 3.14, notations 4 and 5). Like the first three administrative menu options the layout of the columns and rows is similar. By clicking the pencil icon the filename can be changed and by clicking the cross icon the file or directory can be removed (Figure 3.14, notation 2; Appendix A, user stories A3 and P2-8). By clicking the button named Upload File a new file can be uploaded to the server (Figure 3.14, notation 4; Appendix A, user story A3). By clicking the button named Create Directory a new directory can be created (Figure 3.14, notation 5). In order to help with the placement of directories it is shown in the upper left corner above the rows but below the menu (Figure 3.14, notation 1).
4 System Specification

The system specification describes the database, graphical user interface (GUI), and the logic including AJAX, the framework and how it communicates with the database. An overview of the system specified in this section is given in figure 4.1.

![Figure 4.1: System Overview](image)

4.1 Database

This is an explanation about the database. It describes how it is designed and implemented, what software that was used, what choices were made and why.

A database is a collection of related data that represents some aspect of the real world; the data should be logically coherent and have an inherent meaning. Furthermore, the database should be designed, built and filled with data for a specific purpose. (Elmasri & Navathe, 2010) Knowing this, the first step in the creation of the database was the design. Based on the user stories the team brainstormed to get different entities and from the entities the different relationships. Focusing on the customer and the products, the team drew up an ER-diagram (Entity Relationship diagram) to visualize how the entities should interact with each other and which ones that was necessary. The first draft of the database prototype consisted of 10 entities and 11 relationships. When implementing the database it was decided that the member who first picked a user story requiring a certain entity would be responsible for adding it to the code. Since it was possible that the team had not thought of every entity or attribute needed, every member of the team was free to add more if they needed to. This proved to be true and during development more tables were added and in the end there were 14 entities and 13 relationships (see Appendix C: ER diagram). Carousel and promotions do not have relationships to other entities in the database.

To create and maintain a database, a database management system (DBMS) is required. The DBMS makes it possible to query the database for information and manipulating the data (Elmasri & Navathe, 2010). For the database of Eldflugan MySQL was used as the DBMS and SQLAlchemy was used as an object-relational mapper. SQLAlchemy makes it possible to write the database queries in python code, without considering which database type is used. When the team was testing changes locally SQLite was used instead of MySQL. This was mainly because of the convenience of not having to host a database server.
The strengths in using a relational DBMS is that the database system has a self-describing nature since it stores metadata about the different tables. This metadata describes type, storage format and other various constraints on the data. One other advantage is the sharing of data and multi-user transaction processing, which means that the DMBS ensures that if several people try to update the same data it will not cause any errors in the database. (Elmasri & Navathe, 2010) On a webshop there may be several people trying to purchase the same product simultaneously, and if there is only one item left the DBMS makes sure that only one purchase goes through. This is a possibility at Eldflugan; therefore using a DBMS was a good choice.

During development several problems appeared, the team lacked knowledge of how to implement a database and that caused it to be delayed. At times there were also bugs that stopped the database from generating the tables and table contents properly. If the tables failed to generate, many areas of the web application would cease to function or not display the content properly. The error logs for these bugs did not give enough information for the team to solve and they remain unsolved, the bugs could not be reproduced and they seemed to happen for no apparent reason. To solve these bugs a temporary fix was to delete and recreate the whole web application.

4.2 Graphical User Interface

This section gives a definition of the graphical user interface, or GUI as it will be referred to, describes the views of the web application, and a description of design principles and how it was used in the project.

4.2.1 Definition of the Graphical User Interface

The GUI is made out of several independent views. For the purpose of this report the team choose to define a view as a basic building block for a user interface. Each page on Eldflugan is made up of several views, some of which are reused throughout many pages. Depending on the screen size and the device used to access Eldflugan, the views are arranged in different combinations and may appear differently. Eldflugan have two types of users: administrators and customers. Depending on which category the user belongs to he or she will use the website in different ways. On that basis the GUI is designed differently for the two types of users.

4.2.2 Views

To navigate around the website there is a menu view. This menu is in turn made up of three sub views, a meta menu, a product menu and a search field. The meta menu contains hyperlinks related to non-product categories. Graphically the meta menu is a horizontal bar that is always located at the top of the screen.

Under the meta menu the product menu can be found. This view is used to navigate to specific product categories, sub categories or specific products. Contained within the horizontal bar that make up the product menu the search view can be found. The search view is a text field that will expand to show miniature product thumbnails as suggestions when typed in. Thumbnails are designed to display a large amount of pictures on a screen of limited size. Given that the web page is accessed through a small resolution screen the meta- and product menu will be merged into one expandable dropdown menu. The menu view will appear throughout all pages.
Figure 4.2: Menu

On the homepage there are two main views: on top a display carousel and under it a grid of promotion thumbnails. The carousel is a container that is filled with various promotion pictures that interchange each other at a set interval. Users can also navigate between the pictures by clicking on arrow icons at either end of the picture. To inform the user of which picture in the set he or she is currently at there are small circles at the bottom of the container representing the picture set, and the one with a filled circle is the one that is currently shown. Under the carousel reside the promotion thumbnails. These are designed to promote special offers. Depending on the screen size the grid that displays these thumbnails will have more or fewer columns. The menu view will load when the user first enter the site and will not reload after that. This is accomplished by using asynchronous input and output.

Figure 4.3: Carousel

Figure 4.4: Promotion Thumbnail

The easiest way to browse through the assortment of products on Eldflugan is the product category page, which is made up of two view components, the filter view and the product grid view. The filter view is designed to allow the user to specify the types of products that should be displayed on the product category page. Within the view there are four fields, of which the three first contain check boxes where you can decide which room, brand, and color you want to filter on. The fourth field is a bar where you can choose a price range to filter on. Under the filter the product thumbnails will appear. Each thumbnail contains a picture of the product it is representing, the name of the product, the price of the product and a buy button. Depending on the screen size the grid that displays these thumbnails will have more or fewer columns. It is also possible to choose to display the products in a row if a user should prefer this.
Figure 4.5: Filter

Figure 4.6: Product Grid

Should a user require more detailed information about a product there is a product page. The product page contains four views, a product info view, a product picture view, an alternative color view and a rating/discussion view. Information about the product is displayed in the product info view, which is a simple table of information, with each row of information on top of the other. The view lists contains information about brand, color, size, price and stock. A visualization of the product is shown in the product picture view, which simply displays a picture of the product. If clicked the picture will expand and the rest of the webpage will receive a darker tone. Under the product picture view alternative colors are displayed if available. It consists of a row of small thumbnail pictures, each representing an alternative color. At most four thumbnails will lay in one row; if more alternative colors are available there will be several rows.

**Alternativa färger:**

![Krom](image1.png)  ![Guld](image2.png)

Figure 4.7: Alternative Color
A view that is important for the user is the shopping cart. This view expands from the meta menu on click. It contains a representation of all the products that have been put into it. The user can manipulate the products in a number of ways. It is possible to adjust the number of products by pressing plus- and minus buttons as well as removing them completely by clicking the trashcan button. When ready to make a purchase there is a button to redirect you to the checkout page.
The checkout page is made up of two views. The first one displays all the products you are about to buy. This view has the same functionality as the shopping cart. The next view is a form where the user will fill in the relevant information for the webshop to be able to complete the order. The form is made up of text fields, radio buttons and an order acceptance button.
4.2.3 Design Principles

A principle to keep in mind when designing a user interface is Zipf distribution, which states, “A model of menu performance must accommodate the fact that the frequency of item selection is non-uniform.” In other words, few menu items amount for a large amount of user interactions. (Thimbleby, 2007) In the design of Eldflugan consideration have been taken to this principle. Frequent user interaction such as browsing products, filtering and making a purchase have been made easily accessible. Less frequently used functions such as cancelling an order and changing a users default shipping address have been hidden away.

Another reason for hiding away less frequently used features is the Hick-Hyman Law. It states that the time it takes to reach a decision goes up as the number of choices goes up. The formula for calculating the time $T$ to make a decision can be described as following: “The Hick-Hyman Law, then, states that the time $T$ to choose an item is proportional to its information content, giving $T=a+b\times H$, where $a$ and $b$ are empirically derived constants. When the user chooses between $C$ equally probable alternatives, the Hick-Hyman Law can be rewritten as $T=a+b\times \log_2(C)$.” $H$ is the amount of information content. Therefore, by hiding some of the less frequently used features the user can make a decision faster when first entering Eldflugan. (Cockburn, Gutwin & Greenberg, 2007) See figure 4.13.

Considering the limited screen real estate available to display products the user can find it is difficult to overview what data is available. To counter this Eldflugan is designed around Ben Shneiderman’s Visual Information Seeking Mantra: “Overview first, zoom and filter, then details on demand” (Shneiderman, 1996). See figure 4.5 and 4.6.
Drawing navigation is a great way to create a flatter navigation experience for the user. The idea of this type of navigation is to have a “hidden drawer” which is always available for the user. Eldflugan make use of this type of navigation to create a consistent navigation experience for the user. Another good reason for using this menu layout is Hicks’s Law, which states that the time it takes to reach a decision goes up as the number of choices go up. By hiding some of the less frequently used features I make it easier for the user to make a decision (Cockburn, Gutwin, Greenberg, 2007). This also corresponds with Pareto’s principle which states that 20% of the features are used 80% of the time, by hiding the other 80% of features the user experience is made simpler (Newman, 2006).

Figure 4.13: Mobile navigation

By hiding the menu in a drawer when accessing the website through a mobile device the signal to noise ratio is optimized. Signal to noise ratio is the ratio between relevant content and navigation features. A successful GUI should display as much relevant content (signal) as possible without compromising the ability to navigate. (Marsden, 2013)
4.3 Logic

The Logic section describes how calls between client and server are handled and describes the framework Flask. The paragraphs about AJAX describes how a part of the content of the web application can be reloaded, not having to reload the entire web application.

4.3.1 AJAX

Throughout the development of the website a speedy and seamless experience for the customer has been an important focus. Because of this focus many parts of the website were loaded using asynchronous request triggered and handled by JavaScript in the customers’ web browser, also known as AJAX (Asynchronous JavaScript and XML). AJAX requests were sent to load only selected parts of the website. One example is the customers previously added shipping addresses, which were loaded with AJAX and reloaded when a customer added another one.

AJAX incorporates several technologies including; HTML and CSS for presentation, dynamic display and interaction using the DOM (Document Object Model), XML and XSLT for data interchange and manipulation, XMLHTTPRequest for asynchronous calls and JavaScript to bring it all together. As seen in Figure 4.14, a user using a traditional web application needs to wait for the response when sending a request to the server. When using a web application with AJAX the requests are sent asynchronously and the user can interact with the web application during the time the request is handled by the server. Therefore AJAX is an approach to make regular websites more like software applications. More technically this is accomplished by loading an AJAX engine instead of loading a web page at the start of a session. Every user action that would generate a HTTP request instead takes the form of a JavaScript call to the AJAX engine instead, e.g. as seen in Figure 4.15. The AJAX engine can handle many simpler tasks on its own, and if it needs data from the server to respond the data retrieval is asynchronous and does not stall the user’s interaction with the application. (Garrett, 2005)

A plugin called PJAX was added to help make the implementation of AJAX easier. This plugin combined AJAX requests with a JavaScript feature for rewriting the URL in the web browser called pushState. PJAX sends a request to the server and adds an http header, which indicates that it is a PJAX request. On the server side a Flask plugin called Flask-PJAX was added to handle the PJAX requests. The server side plugin returns only the main content of the website when a PJAX request is received and leaves out the header, footer and navigation. This is very useful because PJAX requests...
are only sent from a current visitor of the website who already have the header, footer and navigation loaded.

```javascript
$.ajax({
  url: '/api/cart/get',
  type: 'post',
  cache: false,
  dataType: 'json',
  success: function (data) {
    var cart = data;
    updateCart(cart);
  }
});
```

Figure 4.15: AJAX call to handle a customer's cart

### 4.3.2 Framework

**Flask** is a micro framework for **python**, which aims to keep the core simple and at the same time add options to add extensions to give further usability. One of the uses in **Flask** is the routing; it gives the ability to add functions to URLs. (Flask, 2014) Using the routing feature when a customer gets to the homepage of **Eldflugan** a function will be called that queries the database for the necessary data and then renders the appropriate HTML template. Extensions were added to give the team more tools to work with such as **Flask-Admin** that the admin pages are primarily built with.

Using **Jinja2**, a templating language for **python** that is included in **Flask**, data could be sent along when PJAX requests were made and **Jinja2** code and variables could be used in the templates. This is often used when the HTML templates need data from the database, for example when navigating to the products page the database will be queried for all the products and send this data to be displayed in the main content. In the HTML code it will loop through all the products and output a HTML template with correct syntax and data. This makes it possible to have an arbitrary amount of products while still maintaining the design of the web page.

```html
{% extends pjax("masterpage.html") %}
{% block title %}{{product_data.name}}{% endblock %}
{% block body %}
<div class="container">
  <p>{{buy_status}}</p>
  <h1>{{product_data.name}}</h1>
</div>

{% for product in products %}
{% include "single_product_grid.html" %}
{% endfor %}
```

Figure 4.16: Extending templates and using variables

Figure 4.17: Looping over content
4.3.3 Database Communication

Communication with the database was conducted by using *SQLAlchemy* (for more information see 4.1 Database). With *SQLAlchemy* entities were defined as *python* classes, as seen in Figure 4.18 and Figure 4.19, and to a large extent database operations was run in methods on said classes or instance objects of these classes. This helped make the code more readable and helped to structure the code, when all code related to database communication could be gathered in one file and imported to other files through this file.

```python
class Product(db.Model):
    __tablename__ = 'products'
    id = Column(Integer, primary_key=True)
    name = Column(String(50))
    brand = Column(String(50))
    description = Column(String(200))
    price = Column(Integer)
    image_address = Column(String(100))
    color = Column(String(50))
    dimension = Column(String(50))
    lamp_type = Column(String(50))
    room = Column(String(100))
    is_accessory = Column(Boolean)
```

**Figure 4.18: Database python class**

```python
Product(request.form['name'], request.form['brand'], request.form['description'],
        int(request.form['price']), filename, request.form['color'],
        request.form['dimension'], request.form['type'], request.form['room'])
```

**Figure 4.19: Adding a product to the database**

4.4 Refactoring

The system has been optimized through code enhancements that the team implemented during the refactoring part. The team worked with several categories of code refactoring. General code refactoring, generalizing of code, security, and general method were among the types of refactoring that the team implemented.

In the general code refactoring the team exchanged variable names to names that were more fitting to the variable. This was done in order to get a code that is easier to read and understand. Code that had been developed was exchanged for pre existing plugins. The plugins were thought to provide better functionality and would be easier to work with for future developers.

Security was believed to be an important aspect in the development of the site. To increase the security, the code was modified to only allow users with privileges, such as administrators or logged in customers, to access certain pages that could contain sensitive information.

To optimize the code the team tried to generalize it where possible. Functions were made to work with an arbitrary amount of elements instead of a static amount. An example of this is the index page carousel that previously held a fixed amount of pictures. After the refactoring it was possible to display an arbitrary amount of pictures.
To avoid recurring code and code duplications the team developed a general method for sending out mails to customers. This implementation also made the code more coherent and easier to read for current and future developers.
5 Marketing Plan

5.1 Background

In order to prepare Eldflugan for entering the e-commerce business a compact marketing plan including an environmental scanning, a SWOT analysis, a STP (Segmentation, Targeting and Positioning) and marketing mix have been formed. Through knowledge of external factors Eldflugan can adapt a strategy that suits its vision and business concept.

5.2 Environmental Scanning

To identify different factors that could affect the launch of Eldflugan’s webshop an environmental scanning has been done. These factors are presented through a PEST analysis and a brief mapping of potential competitors.

5.2.1 PEST

To recognize external factors in Eldflugan’s macro environment a PEST analysis has been done. To be aware of political (P), economical (E), sociocultural (S) and technological (T) factors will help the company to plan its future strategies correctly in order to maximize opportunities and minimize threats. (HIA, 2011)

5.2.1.1 Political


5.2.1.2 Economical

The Swedish e-commerce market has during the last ten years grown by 655% in general and the turnover of the interior and furniture section has grown by 19% during 2013. During 2013, regular retailing increased by 1.9% compared to e-commerce’s 17% increase. (PostNord, Svensk Digital Handel & HUI Research, 2013) These numbers may have been supported by the increase in buying power of the Swedish population. Since 2005 the real net pay has increased by 23% when cleared for inflation. The increase can be explained by tax relieves and salary increases. (Ekonomifakta, 2013)

5.2.1.3 Sociocultural

According to an annual survey done by the Swedish Statistics Bureau, SCB, 91% of the Swedish population among 16-85 years old has access to the Internet. During the period April 2012 to March 2013 as much as 68% of the population had made at least one purchase in form of a physical product or a service. The most common group to do online shopping on a continuous basis is the ones in an age
range of 16-54 years. (Statistiska Centralbyrå, 2013) This group is considered to feel comfortable when using the Internet and comfortable level is stated to be one of the factors that have a positive influence on online shopping frequency. (Dai, Zhang & Zhou, 2007)

Comparison Shopping Engines (CSEs) are growing in popularity and have become an important tool to get customers for e-commerce businesses. CSEs allow customers to compare prices on products across various shops and act as a connector to new customers and as an equalizer for smaller retailers. For example it is stated that up to 20% of the web traffic on a shop could be due to CSEs and consumers viewing a product via a CSE are twice more likely to buy the product than consumers viewing through another method. (Miller, 2011; Miller, 2013)

According to Anderson, Ponnavolu and Srinivasan (2002) product discounts, delivery rebates and other incitements should be provided on a regular basis in order to get loyal customers. Cultivation, as they call it, creates the deep relationship needed to avoid vanishing customers.

Results from studies made in the USA show that both webrooming and showrooming are pretty common among consumers. Showrooming is when a customer takes a look at a product in a physical store and then does the purchase online wherever it is cheapest. Webrooming is pretty much the opposite of showrooming. In the poll made by Accenture, 65% said that they would do webrooming while 62% could think about partaking in showrooming. (Tuttle, 2013)

5.2.1.4 Technological

It is stated that business revenue can get influenced negatively if a web page initially takes more than five seconds to load. Even a single second matters for the customer satisfaction level and the number of views on the page. Companies that are conscious and understand the essentiality about the end users experience and further adapt and optimize their webshops are more likely to get satisfied customers, a good brand reputation and higher revenues. (Poepsel, 2008) These kinds of adaptions also include small screen devices that are used in an increased extent. 83-86% of the Swedish population in the age range 16-44 has used the Internet through their smartphone during the first quarter 2013. (Statistiska Centralbyrå, 2013)

There is an increase in algorithms that recommend products related to visited or purchased products, and these algorithms have become more precise in order to make the webshopping as personal as possible. The e-commerce market has applied these algorithm techniques through what is called cross selling. (Leavitt, 2006)

5.2.2 Competitor Analysis

There are several competitors in the Swedish lamp e-commerce market for Eldflugan, both those who only offer lamps and light bulbs and those who offer a more complete interior concept. Since Eldflugan, at least initially, is going to be one of the smaller niche players in the market, a comparison will be done only towards equal companies. This comparison excludes the major part of the market; the concepts of the players that today are big and successful are not noticed. But the focus for Eldflugan is to gain an initial market share and therefore it has been an active choice to compare with certain potential competitors.

What can be observed is that a significant part of them have started their e-commerce or even their business in general during some of the past three years. Another useful fact is that actually all of them have increased their turnover steadily over the years. This corresponds well to the facts stated above
about the rapid development in the Swedish e-commerce business during the recent years. Turnover numbers for six potential competitors are presented in the table below. Due to a variety in dates for fiscal years some companies have not reported their sales for 2013 yet. (Retriever Business, 2014)

<table>
<thead>
<tr>
<th>Year</th>
<th>Belysningsdesign</th>
<th>Elstore</th>
<th>Lampan</th>
<th>Ljustema</th>
<th>Rendl</th>
<th>Uttaget</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>0</td>
<td>2,000</td>
<td>3,154</td>
<td>1,739</td>
<td>3,134</td>
<td>1,739</td>
</tr>
<tr>
<td>2011</td>
<td>3,134</td>
<td>3,154</td>
<td>12,300</td>
<td>11,366</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td>6,694</td>
<td>11,366</td>
<td>12,300</td>
<td>11,366</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>12,300</td>
<td>12,300</td>
<td>12,300</td>
<td>12,300</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chart 5.1: Turnover in kSEK for potential competitors of Eldflugan

Since omni channel marketing and selling, i.e. reaching customers through all available shopping channels as mobile, web and physical stores, become more common and also provide a competitive advantage (Brynjolfsson, Hu & Rahman, 2013), a comparison of the competitors’ webshops from a mobile device has been done. This comparison showed that four out of six did not have a small screen device adaption at all. Not having a mobile version can lead to poor performance and slow loading, and this can according to Strangeloop (2013) lead to customers not returning to the site and/or going to a competitor’s site instead.

When browsing the potential competitors’ webshops some differences can be noticed concerning concept and assortment. A majority is offering premium brand products and a large assortment to satisfy all kind of needs while just a few have tried to niche their concept within, for example, LED-light products. About a half of the actors are combining their webshops with a physical store.
5.3 SWOT

The analysis tool SWOT has been used to determine *Eldflugan’s* internal strengths (S) and weaknesses (W) as well as its external opportunities (O) and threats (T). This is a foundation for the market implementation that will be done when launching the webshop. (Kotler, Armstrong & Parment, 2011)

### Figure 5.1: SWOT

#### 5.3.1 Strengths

As mentioned before fast loading times result in more page views, higher revenues, brand recognition and higher customer satisfaction. Thus, as a web application *Eldflugan* has an advantage against some of its competitors, as loading times are lower.

The mobile version of *Eldflugan* provides a competitive advantage as a mobile version can keep many different kinds of customers on the page. Customers using a small screen device can get decreased loading times and still complete the purchase process due to the compact layout of the webshop that *Eldflugan* offers. As mentioned, the extent of Internet using through a smartphone is pretty high and *Eldflugan* is one of the actors that can satisfy the needs of such a user.

Not only by offering a responsive website that gives *Eldflugan* the ability to offer mobile users what they want, the company also offers a channel that some of the competitors cannot which are the physical stores. By having an omni channel network and competitive prices *Eldflugan* makes it possible to develop and keep a strong brand.

#### 5.3.2 Weaknesses

As mentioned, cross selling is becoming an increasingly important method to increase sales and as *Eldflugan* lacks this feature this is a loss of a big opportunity. *Eldflugan* also lack incitements to motivate and create strong relationships with its customers.
Eldflugan’s vision is to offer design interior at a cheap price. Their way to implement the vision is through having a collection of high quality lamps that are not the trendiest brands. This could confuse customers that are more stuck at the brand names than at the actual design specifications of the products.

Another feature not offered by Eldflugan is the opportunity to choose the number of products shown on one page in the product grid when browsing. This could, if the potential customer has a slow Internet connection or if Eldflugan has a bad server host, reduce the number of complete purchases. According to a performance test a page with 54 products took 128% more time to load than one with 5 products (see Appendix B: Webtest).

5.3.3 Opportunities

As mentioned before the e-commerce market in Sweden is in an expansive phase right now. This includes the interior and furniture sector where several new, small players have appeared during the most recent years. If Eldflugan has a successful launch of its webshop they can get a share of the growing market. With the great extent of people using smartphones to reach Internet and that consumers buy products from several channels Eldflugan has a chance to participate in the growth of all channels.

Even though there are no big growth rates in the physical retail market it seems to be useful for the e-commerce and vice versa as this opens up for show- or webrooming. This is something Eldflugan can use to their advantage with their competitive prices and physical stores.

With a vision to sell inexpensive lamps to design interested customers, Eldflugan will be one of the players in the market that can take an advantage of the opportunity that CSEs really influence the consumer behavior. Through this Eldflugan can build a network of potential customers and further get a loyal customer base.

5.3.4 Threats

Major actors in interior decorating have strong brands and can use economies of scales to out-compete smaller actors like Eldflugan. These actors are also more likely to have a strong following of loyal customers who promote the brand.

Eldflugan will be launched a few years after several competitors and this could be a big threat as the competitors have had a chance to develop strong brands and loyal customers, and stealing customers from them could prove hard and expensive.
5.4 TOWS

To take advantage of strengths and opportunities as well as avoiding weaknesses and minimize threats, the tool TOWS has been used to convert a SWOT into strategies (Kindström, 2013).

<table>
<thead>
<tr>
<th>Internal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strengths</td>
</tr>
<tr>
<td>Using Strengths to Maximize Opportunities</td>
</tr>
<tr>
<td>• Omni Channel – New Business Channel</td>
</tr>
<tr>
<td>• Small Screen Device – Expansive Market</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>External</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opportunities</td>
</tr>
<tr>
<td>Using Strengths to Minimize Threats</td>
</tr>
<tr>
<td>• Fast – Market Timing</td>
</tr>
<tr>
<td>• Competitive Prices and Omni Channel – Market Timing</td>
</tr>
</tbody>
</table>

Table 5.1: TOWS

5.4.1 Using Strengths to Maximize Opportunities

By offering multiple shopping channels to its customers, Eldflugan can be one of the players that take the most advantage of the show- and webrooming trend. If a customer finds a product on the webshop interesting the customer should be able to visit the physical store in order to get a better perception of the product. Eldflugan should also be able to steal market shares, for example when people are looking for interior products in a competitors physical store and think that the premium brands are too expensive, they instead go online to Eldflugan’s webshop to find a similar product that is cheaper.

With a rapid growth rate in the Swedish e-commerce business, a high ratio of smartphone users online and the tendency that people shop through many different channels, Eldflugan has an opportunity to gain market shares through omni channeling. Especially if consumers tend to shop more on mobile devices, Eldflugan has an edge comparing to its competitors.

5.4.2 Using Strengths to Minimize Threats

Eldflugan’s omni channel network and competitive pricing can help strengthen Eldflugan’s brand by advertising in the right channels, and thus help steal customers from already established competitors.
By offering a superior web site and shopping experience *Eldflugan* has the possibility to steal customers from big, established sites that have less optimization in loading times and screen sizes. *Eldflugan's* opportunity to compare and learn from established competitors has made their webshop more adapted for today's customers.

### 5.4.3 Minimizing Weaknesses by Taking Advantage of Opportunities

While *Eldflugan's* brands may not be as trendy as the some of its competitors, this allows them to focus on the looks and price of a product. This can lead to price sensitive customers finding *Eldflugan* through a CSE, which is an increasingly common way for customers to find products and websites.

### 5.4.4 Minimize Weaknesses and Avoid Threats

When not offering any cross selling or loyalty program for its customers, *Eldflugan* most likely will not be able to compete against players having economies of scales. It can be positive for *Eldflugan* to avoid such competitors and instead aim for customers who are more interested in a slim concept with low prices than a lot of extra bonuses and discounts.

### 5.4.5 Strategies from TOWS

With TOWS as a starting point, well-adapted strategies can be chosen for *Eldflugan*. Out of the chosen strategies the marketing mix will propose how the company should act in order to succeed.

*Eldflugan* should focus on highlighting the competitive prices of its products as well as the flexibility it has with offering shopping through several channels. With competitive prices it is likely that *Eldflugan’s* products will be available through the increasingly important CSEs. Having two physical stores, a webshop and a well-adapted small screen adaption the company can participate in the show- and webrooming trend. *Eldflugan* should also take advantage of its omni channel edge due to the expansive market in combination with the tendency that people more often use their small screen devices for shopping.

### 5.5 STP

A STP is used to guide *Eldflugan’s* development and implementation of an appropriate marketing mix by highlighting possible target markets. Through segmentation (S) and targeting (T) a certain customer group can be selected, combining this with positioning (P) value can be created for the selected customer group. (Kotler *et al.*, 2011)

#### 5.5.1 Segmentation

Relevant aspects for *Eldflugan’s* segmentation are income, location, interior design interest and age. Income is divided in low-, middle- and high income, where middle is between 180,000 and 400,000 SEK. Location is divided in distance from *Eldflugan’s* two physical stores, which is where the support and brand knowledge is highest. The interior design interest is an important aspect for a customer of *Eldflugan* as the site is designed to encourage a lot of browsing. This aspect is a bit diffuse, and it can be hard to determine which level a certain customer belongs to. It is more of an informal aspect where
the levels are decided by Eldflugan based on perceptions. The segmentation will also cover the age spans 18-34, 35-54 and 55+. Figure 5.2 shows these aspects.

![Figure 5.2: Segmentation](image)

### 5.5.2 Targeting

From the segmentation, the most attractive segments are chosen based on their size, attractiveness and growth potential.

As Eldflugan is focused on price sensitive customers low- and middle income takers are chosen. Because Eldflugan already is established with two stores in the Linköping area, Linköping and Östergötland will be the focus segment. Eldflugan is designed for customers who like to browse and shop around, and an interest in interior design is believed to be crucial. Lastly, as younger people tend to shop more online ages 18-34 will be targeted. An overview of the selected segments is shown in Figure 5.3.

![Figure 5.3: Targeting](image)
5.5.3 Positioning

_Eldflugan_ offers the same functionality and quality of products as its competitors to a lower price. The positioning strategy is therefore “same for less”, i.e. offering the same value, but for a lower price. The niche is to offer high quality products but to avoid premium brands. The competitors offering a very large assortment of premium products could be categorized as “more for same” while, for example, the more niched company offering a large assortment of LED-lamps has a positioning strategy like “more for more”.

5.6 Marketing Mix

The marketing mix describes how _Eldflugan_ should be positioned considering the four Ps, which are the attributes of the product, the price and place, and how it should be promoted. (Kotler et al., 2011)

5.6.1 Product

_Eldflugan_ offers a large selection of varying interior products, which are made accessible for the consumers through offering a possibility to browse with _Eldflugan_'s powerful multi filter tool. This enhance the chance that a consumer finds what its looking for which hopefully increase the number of complete purchases. The multi filter tool also enables the interior design interested consumer who knows what he or she is looking for to quickly find the right item. A combination of a simple payment process, fast deliveries and high quality products will make customers satisfied and likely to return to _Eldflugan_.

5.6.2 Price

Since _Eldflugan_ is trying to gain a share of an expanding market and not trying to steal market shares from competitors, or at least not initially, neutral pricing is a good strategy for the company. This allows the company to get a sufficient margin level in order to get a stabilized profit as soon as possible. The neutral pricing is set in relation to the value of underlying interior products, which means that _Eldflugan_ will offer lower prices than its competitors. The competitors that offer all the premium brand products will simply be more expensive than _Eldflugan_.

5.6.3 Place

_Eldflugan_ has two physical stores and a stock to send products from. When shopping from the online store customers can choose from collecting their products at one of the stores or to send products by mail or express mail.

5.6.4 Promotion

_Eldflugan_’s focus will initially be to use cost-efficient ways of promotion as Search Engine Optimization and _Google AdWords_ to gain a wide customer base. _Eldflugan_ will also be active in social media, as _Facebook_, _Twitter_ and _Instagram_. A suggestion is that in the future, customers who like or follow _Eldflugan_ will receive special offers and promotions through these channels.
6 Ethical Aspects

Online privacy is a key ethical issue for online operators (Stanaland, May & Miyazaki, 2011). The lack of trust is one of the most common reasons of consumers hesitating to shop online, and there is a correlation between the perceived ethical performance of a shopping website and consumer trust (Ming-Hsien, Chandlrees, Binshan & Hung-Yi, 2009). Main reasons for this are that transactions are more impersonal, anonymous and automated. Trust in business to consumer e-commerce, as in the case of the website this report is based on, tend to be more difficult to attain than trust in business-to-business e-commerce. The reasons for this are that business to consumer is usually shorter in term, and more transaction orientated. (Maleki & Akbarzadeh, 2012)

6.1 Legal Aspects

According to Personuppgiftslagen (Swedish Personal Data Act), it is permitted to collect personal details, such as name and personal id, from a person as long as this person agrees to it. The personal data should always be treated correctly and only with the purpose of its collection, and it should not be stored longer than required for the purpose. (PUL, 1998:204) At the website mentioned in this report, this agreement is made as a part of the registration. It could be claimed it should be easy for the customers to complete their next purchase as an incentive for storing the information. However, there have been cases when this is not incentive enough, for example the company Ellos got criticized by Datainspektionen (Swedish Data Inspection Board) for storing information too long (Sveriges Radio, 2013). The ethical aspects of handling personal information existing in addition to these judicial issues are discussed further in following pieces.

6.2 Handle User Information

As stated before, trust is an important factor in e-commerce business. There are several ethical challenges in e-commerce activities (Maleki & Akbarzadeh Pasha, 2012), but customers who are less concerned about their privacy at a website are more willing to share their personal information, also for marketing purposes (Nam, Song, Lee & Park, 2006). Preibusch, Kübler & Beresford (2013) concluded that when customers have to choose from a website where they had to give out less information, or a cheaper website where they had to give out more, the majority picked the latter. On the other hand the company with a stronger privacy policy do have a higher profit per sold unit and more satisfied customers. (Preibusch, Kübler & Beresford, 2013) These facts indicate the authors could collect all the relevant and usable information about the customers, as long as the secure storing is communicated to customers to build up trust among them.

During the process of developing the website mentioned in this report ethical aspects of handling the personal information were thought through. Personal information can be accessed only through entering password and email address at the website this report is based on. If the customer has forgot his or her password, an email with a temporary key can be sent to create a new one. When the customer is registered in the first place, its self-chosen password string is encrypted together with a salt key that all customers have in common and a personal, randomly generated user salt. The encryption is done through joining the three strings together and converting them to a 32 digit hexadecimal number, using the hash function MD5 (Message Digest algorithm 5). This is a one-way encryption, which
means that the authors cannot extract the actual password string that the customer typed during registration. Instead a comparison of the encrypted hashes is done during the login process or when a customer changes its password in my pages. This procedure prevents unauthorized users from stealing the customers’ private information. (Fernandez, 2014) The authors asked several potential users to test the website. In internal documents as well as presentations during the project they were referred to as testers, and their identities were not mentioned. They also agreed to share the feedback they gave. In case they, during the process of testing the website, tried to buy products their information was stored in the database and could be accessed by the authors.

6.3 Communicating Ethical Strategy

There are several possible actions to be taken into consideration in the marketing plan for the release of the website discussed in this paper. To establish the website’s ethical image and strong consumer trust, an effective ethical strategy with matching plans and actions should be developed. This should be accomplished through having a clear privacy policy at the website, disclose information about the website clearly and trustfully, tell consumers how their information will be used and perform good business ethics, e.g. by conducting ethical practices. (Ming-Hsien, Chandlrees, Binshan & Hung-Yi, 2009) These actions may be implemented in several ways at the website discussed in this paper. A clear privacy policy could be linked to at the point of approval of storing the personal information. This policy could also tell the consumers how their information will be used. More information about the owners of the website and history could disclose information about the website clearly and trustfully, and in a way showing the performance of good business ethics.

Nam, Song, Lee and Park (2006) also point out some constructs of trust on a website: convenience of a website, the reputation of the website and a third party certificate. These would be important aspects to keep in mind when launching and marketing the website. To have a third party certificate, or a privacy trust mark, also has the potential to result in lower privacy concerns, higher perceptions of advertiser ethics and trust and higher willingness to recommend the website to others (Stanaland, May & Miyazaki, 2011). One Swedish privacy trust mark that could be used for the website discussed in this paper is Trygg E–Handel (authors’ translation; Safe E-commerce), aiming to make it safer for the customer to purchase online (Trygg E-Handel, 2014).

6.4 Use of Personal Information for Marketing Purposes

The ethical issue of online privacy is a key issue for advertising online (Stanaland, May & Miyazaki, 2011). Collecting data on consumers’ buying patterns, preferences and personal information make a internet marketing strategy that focus on a long-term relationship and a series of business transactions with a customer possible (Nam, Song, Lee & Park, 2006). To abstain immoral behavior may be a way to differentiate itself from competitors (Ming-Hsien, Chandlrees, Binshan & Hung-Yi, 2009). At an early stage in the development process of the website mentioned in this report, the authors were planning to customize the My Pages section for each customer based on past purchases. There would be special offers based on earlier purchases to tempt customers to buy more.
7 Summary

The purpose of the summary is to discuss different aspects, and to summarize and draw conclusions about the project.

7.1 Discussion

In this section code and functionality are discussed in a future development perspective, as well as the development process.

7.1.1 Functionality in the Future

In the market plan for Eldflugan, weaknesses are pointed out in the SWOT analysis. If new functionality is added, these should be considered, and functionality that reduces these weaknesses should be implemented to increase the profit of the business. One of the weaknesses mentioned is the lack of cross selling opportunities. The database of the website today collects information about the purchases and the website has space available at my pages for customization and special offers. Developing the website further and implementing functionality for cross selling would add value and should be considered in future implementation. Another weakness is that Eldflugan does not offer the opportunity to choose the number of products shown in the product grid. This should also be considered when developing the website further. As the market plan indicates these relatively small changes will have a big positive impact on the experience of the website and motivate customer loyalty. Furthermore, possible functionality in the future should be aligned with the vision. One possible feature that would align well with the vision to allow fast and simple purchases is a one-click-purchase button. This would allow users to purchase a product with one click, without having to enter address and credit card information thereby lowering the barriers to order a product from Eldflugan.

7.1.2 Analysis of the Development Process

Throughout the project Scrum was used as a managing development tool, having both advantages and disadvantages. This being an agile development process, the Scrum methodology seemed appropriate to use. However, some of the main advantages of Scrum methodology were not fully utilized due to the project’s structure. One aspect where the project did not follow Scrum methodology in the classical sense was the lack of a product owner and instead having the development team being responsible for the product owner’s tasks. The product owner’s main tasks are to maximize product value and to manage the product backlog. Because no real product owner existed in the project, these functions may have been lost. An extension of this was the use of user stories. User stories are mainly designed to be equally understandable for the stakeholders and the developers, but in this case, when the developers themselves designed them, it was viewed more as unnecessary work than an asset to the project’s development. Since user stories are not a necessity in Scrum, the development team might have found it more productive to design the backlog items in a way they found suitable. However, the user stories also put the focus on the user perspective when developing functionality, which is very important to keep in mind. Having said that, with a learning perspective being this project’s main aim, both the Scrum methodology and the applying of user stories was very rewarding.
The last sprint in the project was labeled as a refactoring sprint; the team did not develop any further functionality but focused on improving the current code base. In traditional Scrum development, refactoring is done continuously during the entire project. By moving all refactoring activities to one sprint the team could focus entirely on improving the existing code. In the aspect of having an agile development process it might have been beneficial to follow traditional Scrum practices. This would have allowed the team to iterate over previous improvements. Another aspect to be considered is the team’s knowledge of how to refactor code. Initially the team had limited knowledge of how to do this. Therefore, by doing all of the refactoring activities simultaneously significant knowledge and experience was gained compared to what would have been achieved if refactoring had been done continuously through the project. Another part of the refactoring was replacing code created by the team for ready-made solutions. Doing this can help future development from a generalization perspective and can also save a lot of time. This is something the team has learned and will remember in future development projects.

As mentioned, daily Scrum meetings were held three times a week and because the team members did not work on the project full time this were considered enough. The importance of these meetings have changed during the course of the project; during sprint zero they were considered unnecessary as no development took place, but in sprints one to three the daily scrums were considered a useful tool for keeping track of what the team members were doing and what difficulties the project faced.

Before each sprint the team performed a sprint planning during which the team got to decide what the sprint would consist of and how the team should reach the goal for the sprint. These planning sessions have been useful for the team as they provided an opportunity to decide on what to do with items that were delayed and prioritize items in the product backlog. This should have been the responsibility of the product owner, and since no product owner existed the sprint planning were used in a way that helped the team with the vision.

After each sprint a sprint retrospective was performed. The retrospective helped the team find and avoid weaknesses and to improve the team’s strengths. During the retrospectives several ways of improving the team’s work have been found, many of which have been implemented and used and have helped the team boost their productivity. Still, there were some weaknesses that were found in all sprints that the team did not solve, which implies that the actions of these were not followed properly. Not being in time for meetings was a problem that came up in every retrospective, even though the team added incentive to not being late the problem persisted throughout every sprint.

To approximate the time consumption for the development of various features on the site the team played a game of planning poker during sprint zero. During sprint one the actual time consumption of the development was nowhere near the estimations that had been done during the previous sprint. Therefore it was decided that the group did not have enough experience or knowledge to do accurate time estimations. In the following sprints the group focused on completing the existing user stories, without taking consideration to the existing time estimations.

Technically the team was satisfied with the IDE. An important feature of the chosen IDE was that it worked on multiple operating systems. This allowed the team members to work in the same environment regardless which operating system they used and thus the team could more easily standardize they way they worked with the code.
### 7.1.3 Analysis of the Code in a Future Development Perspective

The development of the Admin page had the main purpose to make it easy for the owners of Eldflugan to maintain their website; to make changes on their own and develop it further in the future. These things were thought through, and the team used the Flask-Admin tool that has several advantages. At first, it is easy for the administrator of the website to change its contents and range of products. The website could be delivered to a person outside the development team, and this person could administrate all the important parts of the website. Second, as Flask-Admin is used, the admin page could easily be extended with more functionality in the future, based on existing code and classes from the database.

Throughout the development there was a focus on making the code look good, mainly by following the existing style guide for python. The file structure has been restructured multiple times to make it as logical and intuitive as possible. The improved file structure also helped to decrease that amount of imports in most files, for example almost all pure database communication exists in one file instead of multiple as it did at the beginning of the project. Some of the refactored code from sprint three was related to these aspects and that sprint played a great part in the overall look of the code. These factors may be important if the website would be further developed in the future, because they would help a developer to understand and overview the code.

The website is developed in a way that should not create any major problems if it had to be scaled up in the future to meet demands of an increasing amount of customer. This is an important factor for a growing business and securing the scalability is a key aspect to increase the chances of keeping up with demand.

In the existing code, the protection of the password is implemented and handled by the development team. It is encrypted to protect the information. Due to ethical aspects as mentioned earlier the protection of the password is important to give a secure impression and ensure further sales. The password is encrypted as well as managed by the development team, but the existing code may not easily be developed to become more secure in the future, which limits the future customer management. Instead, to ensure the protection of the customers in the future, it would have been a better idea to outsource the protection of the password to a third party.

### 7.1.4 Difficulties during Implementation

During the project the team faced many technical difficulties. In the beginning of the project the first problem that appeared was how to implement the database in the best possible way. A Python plugin called SQLAlchemy was chosen. This required extra knowledge from the team, and this knowledge had to be acquired. SQLAlchemy helped save time in the implementation by not having to adapt the database queries to other database technologies.

When the MySQL database was set up on OpenShift another problem came up; how would the local computers react when testing connection to the database server on the server? Looking up an OpenShift specific system environment variable containing the database address to connect to solved this problem. If said environment variable did not exist an SQLite file was created instead, because during development the website was most likely to run on a local computer. This problem was not solved fast enough because before the SQLite solution was created the team’s OpenShift server went down for half a day. During this time no programming could be tested because the team had relied on
OpenShift and tested the code online directly since the local database was not set up. The problem with OpenShift was related to an update they had done on some of their servers, including the one used by the team. The problem with the update was resolved by OpenShift, but meant that the team had lost approximately 21 hours of work.

Later in the project another problem appeared which was not as easily identifiable. The problem was noticed when pushing updates to the OpenShift server. During pushes an error message was printed, saying something attempted to be installed in a directory where the team’s account did not have the right permissions. An effort to solve the problem was conducted by switching to another virtual environment for python. The attempt failed but deleting the server and creating a new one resolved the problem. The same problem appeared again later in the project.

When the problem with OpenShift appeared the second time the team decided to move to another platform. The new platform was similarly to the server at OpenShift a virtual machine hosted on another server. The server was set up and configured by a team member and deployment scripts were acquired. The source code was moved to GitHub since a graphical interface seemed like a nice feature. This set up worked very well and had more possibilities than OpenShift, or at least the free version of OpenShift. With this new platform the possibility to have multiple versions of the website up and running at the same time was easy. This was great for having multiple branches of the website, one for development and one for production.

All was well until it appeared that keeping the website on OpenShift was a requirement. At that point these previously mentioned remarks were presented and later the website was moved to a new server at OpenShift. This move caused some issues with the database that were unresolved because the team ran out of time.

During the development the team had some problems with third party frameworks, both JavaScript on the frontend and python on the backend. Notably when implementing the PJAX script on the frontend there were some difficulties. The problem was related to the nature of PJAX; loading partial web pages instead of loading everything, which meant that if the web page had pre-existing functions that had been triggered on the load event. These functions had later also to be triggered on the PJAX specific success event, i.e. triggered when a web page loaded with PJAX had been successfully loaded. This adaption required some experimentation because some functions were malfunctioning when run more than once in the same session. The problem was solved by only running the malfunctioning functions when the website first was loaded. One way to minimize this problem would have been to implement PJAX earlier so that it would have been fixed in many small steps instead of one massive step.

Most of these problems could have been prevented if the team had assessed potential problems at the beginning of the project. For future projects this will be remembered and highly valued by the team members based on the experiences in this project. The team really liked the graphical interface and functionality of GitHub and it is likely that this or similar tools will be used by the team members in future projects. Because of the difficulties experienced with OpenShift the team members will probably evaluate the alternatives if faced with a choice between OpenShift and similar alternatives in the future.

7.2 Conclusion

There are several conclusions to be drawn from the project and the discussion above. The Scrum methodology was applicable in a straightforward way on the project. If there were a product owner, the product backlog would probably be even more important and urgent for the process. The team did
however, although some parts were not as usable as others, learn how to use Scrum in a software development project and did get a deeper understanding for the benefits of Scrum. An important insight drawn from the market plan is how functionality might affect market aspects and the willingness to make a purchase among the customers. Another aspect is how strengths and weaknesses in the code may affect future implementation. Different implementations may give the same functionality, but different opportunities for the future. An admin tool that make it easy for the administrator to maintain the website, and code that can be developed without changing the fundamental structure, will facilitate future development. To prevent problems during the implementation, it would have been beneficial to make a risk analysis. In this risk analysis all thinkable problems or risks should be evaluated and their impact assessed. If the impact and the likelihood of the problem are greater than a selected threshold a plan to solve the problem must be created. It would also be a good way to start thinking of ways to prevent some risks.

To sum up, the team learned about Scrum and the working process, as well as overcome technical challenges and how to solve problems that might come up during software development.
References


Appendices

Appendix A: User Stories

In the text the number and letter beginning each new story refer to user stories. If all stories are referenced it will be by the name inside the brackets. Beneath some of the user stories are noted acceptance tests to clearly indicate what is required of the function. The functions lacking them are because the developers have found that the function description should suffice.

Admin-type functions

(A1) [Administration] as administrator I would like the ability to recreate the original database structure in case something goes wrong. Sprint 1

(A2) [Administration] as administrator I would like the Admin to be password protected so that no unauthorized people can access it. Sprint 2

(A3) [Administration] as administrator I want to be able to manage the range of products. Sprint 1 complete 2
   a. Using a form the user should be able to add a product and all surrounding details.
   b. If necessary the user should be able to edit or remove data on a product or remove the product entirely.
   c. Product management should be easy and intuitive.
   d. The user should also be able to choose which products are to be promoted.

(A4) [Administration] as administrator I want to be able to manage orders so that products can be sent out etc. Sprint 2
   a. Given that an order has been placed the user will be able to manage the order status.
   b. Making changes should be easy and intuitive.

User-type functions

My Pages

(Mp1) [My pages] as a customer I want to be able to log out so that I can avoid other people using my account. Sprint 1
   a. Logging out should always be easily accessible.

(Mp2) [My pages] as a customer I wish to be able to login so that I can access my customer information etc. Sprint 1
   a. Given that the information the user uses is correct he/she will be logged in.
   b. Given that the information the user uses is incorrect he/she will be notified via alertify.
   c. Logging in should always be easily accessible.

(Mp3) [My pages] as a customer I want to be able to edit and save my customer information so that purchased items are always shipped correct. [14] Sprint 1 complete 2
   a. Given that all necessary fields in the form has reasonable data and the user clicks save addresses should be updated.
   b. Given that all necessary fields in the form do not have reasonable data and the user clicks save he/she will be informed of the error.
c. The form clearly indicates what should be written where so the user is never confused.

(Mp4) [My pages] as a customer I want to be able to see my purchase history so that I can see what I have bought. Sprint 1 complete 2
   a. Given that the user is logged in, he/she should be able to see all previous and ongoing purchases and their related data.

(Mp5) [My pages] as a customer I want to be able to **register** an account so that I can store and easily access customer information. Sprint 1 complete 2
   a. Given that the information entered is valid and the user is not already registered he/she the user will be added to the database.
   b. Given that the information is not valid or the user is already registered he/she will receive an appropriate message.
   c. It should be obvious what information is required to register an account.

(Mp6) [My pages] as a customer I want to be able to access my wish list so that I can resume purchase later on. Sprint 1
   a. Provided the user adds products to the cart, but for some reason does not immediately wish to purchase these items. The shopping cart can be stored so that the customer can continue shopping at a later time.
   b. The stored shopping cart can be accessed whenever logged in from the my pages interface.

(Mp7) [My pages] as a customer I want to be able to return an order and see that on my order status on my pages. Sprint 1 complete 2

(Mp8) [My pages/Shopping cart] as a customer I want to be able to store multiple addresses so that I can more quickly perform purchases in the future. Sprint 1 or 2
   a. Given that the information is correctly inputted when the form is submitted the address information will be stored.
   b. Given that the information is incorrect when the form is submitted the user will be notified properly.

(Mp9) [My pages] as a customer I want to be able to generate a new password in case I forgot my old one. Sprint 1 or 2

(Mp10) [My pages] as a customer I want to be able to edit my password and other personal information. Sprint 2
   a. Given that the user for any reason need to change data stored when registering i.e. password, e-mail and name. It should be easily accessible from the “my pages” interface and perform just as well as upon registering.

Shopping Cart

(Sc1) [Shopping cart] as a customer I would like to go from my shopping cart directly to the checkout counter so that I can complete my purchase. Sprint 1 complete 2
   a. Given that the shopping cart is open and the user presses the “checkout” button he/she will be redirected to the checkout counter.
   b. Given that the shopping cart is open the user has access to press the “checkout” button.

(Sc2) [Shopping cart] as a customer I want to change the number of products in the shopping cart so that I do not have to go back to the product page. Sprint 1 complete 2
   a. Given that a product is in the shopping cart and the user via button or text field adds or removes products to be purchased, the amount is updated.
(Sc3) [Shopping cart] as a customer I want to be able to go to the product pages for the products I have in my shopping cart so that I can go back to where I found them. Sprint 1 complete 2
   a. Given that there are products in the shopping cart the user will be redirected to the products individual product page upon clicking the product.

(Sc4) [Shopping cart] as a customer I want to be able save my shopping cart as a wish list so that I easily can add items later. Sprint 1

(Sc5) [Shopping cart] as a customer I want to store multiple items in the shopping cart so that I can buy all the things I want at once. Sprint 1 complete 2
   a. Given that the desired product is in storage, when the customer clicks the buy button the product will be added to the cart.
   b. Given that the desired product is not in storage, the customer will be able to track that product and receive information when it will be in storage again.

Checkout

(C1) [Checkout] as a customer I would like to choose between different payment and delivery options so that I do not have to adjust myself after the site. Sprint 1
   a. Payment method is updated on click.
   b. It is easily recognizable which option is which.

(C2) [Checkout] as a customer I would like to change/edit my address information in the checkout counter so that I can be certain my products are shipped to the correct address. Sprint 1 complete 2
   a. Given that the customer is in the checkout counter he/she can see all stored address information and using the same functionality as described in user story Mp3 edit any existing information without leaving the checkout counter.

(C3) [Checkout] as a customer I want to be able to place an order so that I can so that I can receive the products I want. Sprint 1

(C4) [Checkout/Email] as a customer I would like to receive email confirmation after I have placed my order so that I feel certain everything is all right. Sprint 2

(C5) [Checkout] as a customer I would like to perform a purchase without having to register to the site. Sprint 2

Products

(P1) [Products] as a customer I would like to share products via social media so that my friends can see what I like and am interested in. Sprint 2
   a. Facebook
   b. Twitter
   c. Google+

(P2) [Products] as a customer I want to be able to see multiple products so that I can easily overview them and determine which one I like. Sprint 1 completed 2
   a. Given that more products than one exists in a category and the customer has selected that specific category he/she can easily survey all existing products in a grid.

(P3) [Products] as a customer I want to be able to filter products by attribute and criteria so that I easily can find something that fits what I am looking for. Sprint 1 completed 2
   a. Given that the customer has set multiple criteria and clicked the “filter” button all products fitting those criteria will appear.
b. The customer should be able to filter on all useful attributes any product has in the
database.

(P4) [Products] as a customer I want to be able to choose to view the products in a list or a grid so
that I can view the products in the way I want.

(P5) [Products] as a customer I would like to see and leave ratings on products so that others can
enjoy my opinion. Sprint 1 complete 2
   a. Given that a user is logged in they can leave their opinion of a product in the form of a
      rating.
   b. Given that a user has left a rating it can be changed
   c. Any user can see the average rating of a product

(P6) [Products] as a customer I want to be able to choose quantity so that I do not have to press the
buy button several times. Sprint 1 complete 2

(P7) [Products] as a customer I want to be able to see a picture of the products so that I know what it
looks like. Sprint 1 complete 2

(P8) [Products] as a customer I want to easily overview a single product and all its related data so
that I know I am choosing the right product. Sprint 1 complete 2
   a. Given that the user is in the product page he/she can see all necessary data in an
      intuitive manner.
   b. Given that the product is in storage the user should be able to add it to his/her shopping
cart.
   c. Given that the product is available in different colors the user should be able to easily
      view all color variants.

Search

(S1) [Search] as a customer I would like to receive suggestions on products when I write something
in the search field. Sprint 1 complete 2
   a. Given that matching text exists in the database suggestions will appear as the user
      writes.

(S2) [Search] as a customer I would like to quickly search for something using free text so that I
more quickly can find the product I am looking for. Sprint 1 complete 2
   a. Given that matching text exists in the database, when something is written in the search
      field.

(S3) [Search] as a customer I would like my suggestions to come with a short description I.E.
picture and price when I am searching for something. Sprint 1

(S4) [Search] As a customer I would like to be able to quick search for products fitting a certain
attribute like color or room. Sprint 1 or 2
   a. Given that matching text exists in attributes such as room or color suggestions will
      appear based on that.

Other

(O1) [Sprint 1] make the design for the master page sprint 0/1
   a. Make a basic design for the master page that fits with the desired functionality.

(O2) [Navigation] as a user I would like to have good menus so that everybody can find everything
easily at all times. Sprint 1
(O3) [Front page] as a user I want the first page to give information about what the site has to offer so that my interest is peaked. Sprint 1 completed 2
   a. The administrator can change what is promoted on the product page.
   b. Suggestions should appear in an intuitive manner.

(O4) [Email] as a customer I want to receive emails when something happens with an order I have placed. Sprint 2

(O5) [Meta] as a customer I want to easily find the companies contact information so that I can contact them if necessary. Sprint 1 complete 2

(O6) [Mobile] as a customer I want the page to look ok regardless of my screen size. Sprint 1 complete 2

Blocked

The following functions were blocked either because they were considered unnecessary or the development team saw fit for some reason to exclude them from implementation.

(P9) [Products] As a customer I want to choose the number of products viewed per page so that I can see the products I wish.

(Sc6) [Shopping cart] as a customer I want to use rebate codes so that I can make cheaper purchases.

(Sc7) [Shopping cart] as a customer I want to receive suggestions on products that are related to those in my cart so that I can make an informed purchase.

(O7) [E-mail] as a customer I want to monitor products and receive e-mail when they are once again in storage so that I do not personally have to check up on it.

(C7) [Checkout] as a customer I want to fetch my address based on my social security number so that I more quickly can make a purchase.

(A6) [Administration] as administrator I want to answer reviews so that customer experience can be increased.

(P10) [Products/Checkout] as a customer I want to know shipping costs for the product I buy so I know how much to pay.

(A7) [Administration/Navigation] as administrator I want to edit the data contained in the menus so that I can adjust it after trends and seasons.

(A8) [Administration/Meta] as administrator I want to be able to create webpages so that I can create more content than products.

(A9) [Administration] as administrator I want to write a blog so that the customers can take part in my insights.

(Mp11) [My pages/Meta] as a customer I want a public profile on the page so people can see what I like.

(Mp12) [My pages/Meta] as a customer I want to share my wish lists so that my friends can see what I want and am interested in.

(C8) [Checkout/E-mail] as a customer I want to be able to monitor products and if it is not in storage I want to receive an e-mail notification when it is so that I can purchase it then.
Bug fixes

(B1) [BUG] Price slider is not set after maximum-minimum possible prices on products.
(B2) [BUG] Shopping cart cannot be seen at checkout counter
(B3) [BUG] Alertifies on major changes.
(B4) [BUG] All links on main page is incorrect
(B5) [BUG] Not able to remove products using ‘-’
(B6) [BUG] Validated text areas are larger
(B7) [BUG] Cannot access products containing criteria with åäö
(B8) [BUG] Is not logged out when logout button is pressed
(B9) [BUG] Mobile navigation bar is not folded up as I press login/logout
Appendix B: Webtest

<table>
<thead>
<tr>
<th>Load Time</th>
<th>First Byte</th>
<th>Start Render</th>
<th>Speed Index</th>
<th>DOM Elements</th>
<th>Document Complete</th>
<th>Fully Loaded</th>
</tr>
</thead>
<tbody>
<tr>
<td>First View</td>
<td>12.486s</td>
<td>5.116s</td>
<td>0.760s</td>
<td>8073</td>
<td>12.486s</td>
<td>80</td>
</tr>
<tr>
<td>Repeat View</td>
<td>1.710s</td>
<td>0.588s</td>
<td>1.710s</td>
<td>1825</td>
<td>1.710s</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure B.1: Loading times 54 products

Figure B.2: Content breakdown of requests and bytes 54 products
## TDDD83 Computer Engineering – Bachelor Project
Linköping University Spring Semester 2014

### Figure B.3: Loading times 5 products

<table>
<thead>
<tr>
<th>Load Time</th>
<th>First Byte</th>
<th>Start Render</th>
<th>Speed Index</th>
<th>DOM Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>First View</td>
<td>6.163s</td>
<td>2.033s</td>
<td>3.684s</td>
<td>3772</td>
</tr>
<tr>
<td>Repeat View</td>
<td>1.857s</td>
<td>1.047s</td>
<td>1.976s</td>
<td>2936</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document Complete</th>
<th>Fully Loaded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Requests</td>
</tr>
<tr>
<td>5.163s</td>
<td>36</td>
</tr>
<tr>
<td>1.857s</td>
<td>5</td>
</tr>
</tbody>
</table>

### Figure B.4: Content breakdown of requests and bytes 5 products

<table>
<thead>
<tr>
<th>Requests</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>52.4%</td>
</tr>
<tr>
<td>js</td>
<td>6.3%</td>
</tr>
<tr>
<td>css</td>
<td>30.1%</td>
</tr>
<tr>
<td>image</td>
<td>3.1%</td>
</tr>
<tr>
<td>other</td>
<td>10.1%</td>
</tr>
</tbody>
</table>

| HTML      | 60.9% |
| js        | 11.9% |
| css       | 10.9% |
| image     | 4.9%  |
| other     | 2.4%  |
Appendix C: ER Diagram

Figure C.1: ER Diagramme
## Appendix D: Retrospective

### Table D.1: Retrospective Sprint zero

<table>
<thead>
<tr>
<th>Positive feedback</th>
<th>Things to improve</th>
<th>Insights</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Atmosphere and Social</strong></td>
<td>• Positive atmosphere, we’re having fun and we’re motivated</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cooperation</strong></td>
<td>• All team members are involved and we help each other</td>
<td>• In lack of communication, we’ve done some things twice</td>
<td>• Remind each other about group contract</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Make decisions</td>
<td>• Assign on Trello, check it continuously</td>
</tr>
<tr>
<td><strong>Development Work</strong></td>
<td></td>
<td>• Report time spent and scrum document</td>
<td>• Need to learn Scrum better, should watch Scrum Video</td>
</tr>
</tbody>
</table>
| **Meetings, Structure and Communication** | • Follow group contract  
• Planning & structure of meetings  
• Great focus during meetings  
• Active listening and open “forum” | • Use Trello  
• Knowledge of Scrum  
• Could use the Supervisor in a more effective way  
• Being on time | • Need to prepare supervisor meetings  
• Read group contract before February 17th  
• Plan use if Trello, have Trello and Scrum document on display during Scrum meetings  
• Watch Scrum Video  
• Go through questions together in the team before supervisor-meeting |

### Table D.2: Retrospective Sprint one

<table>
<thead>
<tr>
<th>Positive feedback</th>
<th>Things to improve</th>
<th>Insights</th>
<th>Actions</th>
</tr>
</thead>
</table>
| **Atmosphere and Social**          | • Great atmosphere  
• Group dynamics                                                                           |                                                                           |                                                                         |
| **Cooperation**                    | • Problem solution  
• Help each other  
• Efficacy                                                                                  | • Doing the same thing twice (use Trello)  
• Overall understanding  
• Focus on meetings | • Person responsible for testing  
• A team member is responsible for a story until someone takes over |
| **Development Work**               | • Good at continuous refactoring                                                   | • Git/version handling                                                   | • Meeting at beginning of sprint 2 to go through code  
• More careful when merging  
• Break down trello-stories  
• More comments  
• Read merge dialogue | • Have meeting to go through code  
• Break down trello-stories when accurate  
• Go through merging routines  
•                                                                                     |
| **Meetings, Structure and Communication** | • Being on time  
• Discuss too little problems at scrum meetings  
• Should stand up and use the Trello more at meetings  
• Being on time, where is the “fika”? | • Complete Scrum document  
• Add problem-part in scrum document  
• Remove Scrum document  
• Discuss not only what but also how  
• “fika” interest rate, go through at Scrum meetings | • Go through Trello at meetings  
• Add problem-part in Scrum document  
• Go through Trello at all Scrum meetings  
• Add problem-part in Scrum document, write more detailed |
Table D.3: Retrospective Sprint two

<table>
<thead>
<tr>
<th>Positive feedback</th>
<th>Things to improve</th>
<th>Insights</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atmosphere and Social</td>
<td>Giving each other positive feedback</td>
<td>Say your opinion</td>
<td>Need more social events</td>
</tr>
<tr>
<td></td>
<td>Team spirit</td>
<td>Lack of social events</td>
<td></td>
</tr>
<tr>
<td>Cooperation</td>
<td>Good atmosphere for questions</td>
<td>Trello</td>
<td>Communicate who is doing what</td>
</tr>
<tr>
<td></td>
<td>Team programming</td>
<td>Focus on meetings</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Good cooperation and communication</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group process progressing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development Work</td>
<td>Testing</td>
<td>Git</td>
<td>More strict Git-routine</td>
</tr>
<tr>
<td></td>
<td>Assign on stories</td>
<td>Openshift</td>
<td>Remove code not in use (i.e. test code)</td>
</tr>
<tr>
<td></td>
<td>Use of Trello</td>
<td>Messy code and difficult to jump on different part</td>
<td>Comment functions</td>
</tr>
<tr>
<td>Meetings, Structure and Communication</td>
<td>More focus than before</td>
<td>Not using time with supervisor as well as could do</td>
<td>Clear definition of when a meeting starts and ends, no use of computers or mobile devices during meetings if not necessary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Do not describe work in detail at scrum meetings</td>
<td>Clarify when meeting is for the whole group and when it is for a few individuals</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Go through former retrospectives</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Prepare for supervisor time</td>
</tr>
</tbody>
</table>

Table D.4: Retrospective Sprint three

<table>
<thead>
<tr>
<th>Positive feedback</th>
<th>Things to improve</th>
<th>Insights</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atmosphere and Social</td>
<td>Great motivation and moral in group</td>
<td>Inefficient group work and lack of focus</td>
<td>Need more focus for individual tasks</td>
</tr>
<tr>
<td></td>
<td>We are having fun</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Team spirit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cooperation</td>
<td>Good distribution of work</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development Work</td>
<td>Less double work</td>
<td>Need more goals and internal deadlines</td>
<td>Clarify deadlines and goals for report</td>
</tr>
<tr>
<td></td>
<td>Good at assigning tasks</td>
<td>We didn’t follow up the last retrospective</td>
<td>Each team member can work where most efficient</td>
</tr>
<tr>
<td>Meetings, Structure and Communication</td>
<td>Joint responsibility for spending the time needed</td>
<td>Being on time</td>
<td>Clarify when there is a meeting and not</td>
</tr>
<tr>
<td></td>
<td>We do use supervisor time more efficiently</td>
<td>Unclear what each team member is doing, do not go through that very well at Scrum meetings</td>
<td></td>
</tr>
</tbody>
</table>

61
Appendix E: Prototype

Figure E.1: Index page
Figure E.2: Index page - scrolled down

Figure E.3: Dropdown navigation
Takkampor

FÄRG
- Röd
- Blå
- Silver
- Magenta

VARUMÄRKE
- Kartell
- Watt & Veke
- Zero
- Fatboy

RUM
- Kök
- Vardagsrum
- Badrum
- Sovrum

PRIS

Filtrera
Återställ

Figure E.4: Filter

Figure E.5: Product grid
Lampa

Detta är en lampa som passar i ditt sovrum!

Mått: 30 cm hög

Färg:

Leveransinfo: Finns i lager

Pris: 500:-

Köp

Figure E.6: Product info

Figure E.7: Cart
**Figure E.8: Checkout**

**Figure E.9: My pages**
Appendix F: Project Plan

Introduction

This is a project by eight students studying I-Data. The goal of the project is to develop a working webshop that sells lamps of different type and form. The team chose to develop this webshop for lamps because lamps are a specific topic that all the members have similar knowledge and picture of, and this will reduce the risk of the members of the team working toward different goals.
Vision

Eldflugan will make it easier for price sensitive and interior interested individuals to find lamps and accessories in a large product range. They will be able to locate desired products through great search and filtering possibilities. Eldflugan offers cheap and stylish lamps through a user friendly website which is designed to get customers to complete a purchase both swiftly and easily.

The time frame for the development of Eldflugan is that the webshop should be finished in week 16 with at least 2912 hours of work. The project has no budget.
Goals

Team Goals

All members of the team should understand all aspects of the product.

All members of the team will understand what has been done in the project.

Members ahead will take responsibility for learning those behind, and those behind will take responsibility for learning more.

The team will have a working product at the end of the course.

Product Goals

We want to create an intuitive, aesthetic and functional webshop. The products and their properties will be stored in a well-structured database. The webshop will have functions such as search, checkout, shopping cart, recommended products, my pages and a filtering function. The website will motivate the customers to be intelligent shoppers. Our goal is that the website can differentiate and compete against similar websites.

Individual Goals

Anna:

My goals for this course are to learn the best way to work using Scrum in a development project and to learn as much as possible about web development. To achieve these goals, I will actively participate in all aspects and do my best throughout the process, and contribute to the group’s common development.

Claes:

My goal for the course is to learn the different languages and tools to be a productive member of the team and actually be a productive member of the group. I plan to achieve these goals by learning the languages and tools while I work on the project to achieve both goals jointly.

Erik:

At the end of the project, I am not unhappy if we do not have a working prototype, but I want to be able to say that I know how to design and develop a website. I want to emphasize that I can code in the languages HTML, CSS, jQuery, Python, essentially all languages covered by the course. In five years, I want to understand how I and the other group members have done, and why we have written what we have written for the result to turn out as it did. I want to construct a database that works and that has an ordered structure. I would of course also carry out an approved work. I also want to confirm that I work well or at least acceptably in a team where I have no prior relationships with the other members.

Gustav:

My goals for this course are to learn how to in a good way to work with development projects in programming, to expand my knowledge of Python and to learn Scrum. The goals will be achieved by actively focusing on using the skills that are taught, during the course, in the project and to conduct self-learning patterns in the problem solving in the project.
Julia:
My goal for the course is primarily to become sufficiently skilled at programming to be able to create a website on my own. I plan to achieve my goal by putting in a lot of time and by making sure that I understand what I am doing, and to learn as much as possible from the team members.

Kalle:
My goal for this course is to actively be a part of a larger project with a successful result. I also want to develop a basic knowledge of web programming and python. I will achieve this by participating in scheduled activities related to the course and by spending as much time as needed on my own.

Pontus:
My goal for the course is to learn as much as possible about web development. I plan to achieve this by actively participating and doing my best throughout the process.

Tim:
I have two main goals with this course: to develop my skills in web programming and to learn the project methodology scrum and agile development. I addition to acquiring these skills, I want to contribute to the project by delivering good code and also by helping to structure the work and contribute to a positive work environment where everyone on the team works effectively.
Group Contract

Times

Scheduled work times are meant for all members of the group to work with the project. In addition to these times, more time will be scheduled for the project.

Timekeeping

All members of the team promise to try to be on time to the best of ability. Team members will announce late arrival via SMS group. Delayed team members are expected to find out what they have missed themselves. The team will not wait for delayed team members and starts working on time.

Calendar

The team has a shared Google Calendar where all group meetings and deadlines are posted. Team members are responsible for entering planned absence from the calendar as soon as possible. The Scrum master are responsible for entering meetings, deadlines etc.

New activities should be booked well in advance before the event occurs.

Time Reporting

Time reporting is done in a spreadsheet on Google Drive. Team members will report time the same day as the work was done.

Scrum Meetings

The team will have Scrum meetings at least three times a week. The Scrum master is responsible for notifying the time and place of meetings in advance of the meeting. The Scrum master is also responsible for entering the meetings in Google Calendar.

Standing meetings are tentative:

- Monday 8.15 - 8.30
- Wednesday 8.15 - 8.30
- Friday 8.15 - 8.30

Should these times not work the meeting is primarily moved to scheduled work times. Members shall as soon as possible notify the other members if they do not have the opportunity to participate in a meeting.

Agenda for the meetings

1. What did I do yesterday?
2. What will I do today?
3. Do I see any challenges that I might need help with or that prevent the team from reaching the sprint goal?

All team members are responsible for filling in the agenda before the meeting. We take a lap in the group before other points are raised.
Communication

SMS Group

The SMS group is used when quick and short information is to be communicated to the whole group.

Email

All members are responsible for reading their mail every day.

If a question is asked by mail, this will be answered within 24h, or faster if possible.

Mail that is relevant for the whole team will be sent to the whole team via lampaonline@googlegroups.com. Reply to the entire group when replying to these emails.

Feedback/Problems

If a problem/conflict between two team members arise, the interested parties must deal with this immediately. Feedback should be constructive. Bring up positive feedback too.

Decision Making

Decisions are taken by majority. The adviser will be consulted in a four against four.

The team also has a possibility to defer a decision if there is a majority for this.

A team member may invoke stretch in the debate. After this each member of the team say one more thing and then a decision is made by voting.

When a team member is absent, the team may make decisions unless the decision is directly related to that team member. Should this be the case, the concerned team member will be contacted before the decision is made.

Definition of Done

A story is moved to Done in a group meeting if no objections are made. If the story is a function it also has to have been tested before it can be moved to Done.

Responsibilities

Scrum Master – Julia

Other responsibilities are determined during the project, if necessary.
Project Organization

Competencies

Anna:
Have taken courses in Ada and Java on the university. Has previously worked in projects both in school and in work and voluntary sector.

Claes:
Knowledgeable in Ada and Java. Have worked on projects before. I have had jobs where I had to be very self propelled so I have learned self-discipline.

Erik:
Can code in Java and Ada. Have worked as substitute economist at a scale industries and thereby have base economic knowledge. Has been captain of my own youth team and shared the responsibility for a younger youth team. Have thus exhibited some form of leadership qualities.

Gustav:
Have knowledge of web design and programming in several languages from high school. Has since taken courses in Ada and Java at the university. Also has experience from working in a group focusing on the development of a website.

Julia:
Previously programmed in Ada and Java. Has also worked in several groups in which I held different roles.

Kalle:
Programmed in Ada and Java. Worked in school projects and other groups in different roles.

Pontus:
Have taken courses in Ada and Java.

Tim:
Have taken courses in Java and Ada at Linköping University and at the University of Cape Town. Having previously run my own webshop I hence have some experience from online marketing, AdWords etc.
Prototype Testing

To receive feedback from outside the team, four persons outside the group tested the prototype. The spontaneous reactions to how the users felt were:

- Intuitive and easily maneuvered page where it is easy to find what you are looking for
- The menu bar at the top of the prototype looks as if it lies on top of the other parts. It would have been prettier with just buttons
- On the index page it would have been better if prices and offers were more clearly displayed to attract more purchases

We will take the feedback into account in our future work.
Appendix G: Experience Summaries

The following Experience Summaries consist of personal reflections and experiences from the project, written in the perspective of each team member.

Tim Andersson

Going into this project I had two goals set for myself. The first goal was to learn how to apply scrum methodology and agile development processes in a software development project. The second goal was to improve my technical- and programming skills to the level were I could comfortably contribute to the development to both frontend and backend parts of a web application. My plan was to achieve these goals by actively working with the other team members as well as acquiring knowledge on my own when necessary. During the duration of this project I have tried to keep these goals in mind.

Scrum is a very effective methodology when developing software in a small team like ours. It allows you to focus on features that actually are requested by the user by working with user stories. The user stories are also a good way to break down a big and complex problem into smaller pieces that seems more manageable. I worked on several user stories, both those concerning frontend development and those concerning backend development. Backend tasks often turned out to be the hardest challenges. These where also the most rewarding as you could develop quite sophisticated functionality. I am particularly proud of the mail functionality in which I had a big part. It gave me insights on how to combine ready-made packets with my own code.

A limitation in Scrum, and an area it does not address is group dynamics and team building. Sense it did not exist any Scrum activities with the purpose to promote team building we organized our own events. These events where important for the group and connected the individual members. Having a team where the team members feel comfortable with expressing their opinions and giving feedback to other team members is an important success factor in project like ours. Another limitation with scrum that our team ran into is that is does not provide any processes to assess risks that can occur during the project. In our particular project we had several major issues that could have been dealt with in a more effective manner if we would have been aware of the risks earlier on.

One thing that I would like to have done differently is the process of development of a concept for the website. Since we did not do a proper marketing plan until after the webpage was complete it is hard to know if our website really fits our targeted customer base. A way to develop our concept further could have been to use The Lean Start Up methodology. Then we would have tried our concept on possible consumers early on in the project to get feedback that would have allowed us to determent if our idea was good or not. With the use of The Lean Startup method you can minimize the amount of code you write and actually develop products and services that are useful to consumers (Blank, 2013). A measure we did take to test our concept was to do user tests at the end of sprint one. This provided us with inputs on how to improve the current functions. However, I did not test if customers would be willing to pay for our product, which was a flaw. The reason why the team and I ended up developing our concept this way was because the focus of this project was to fulfill the course objectives and not to sell our product.

When looking back on the project I feel like I have achieved my personal goals. I have participated in and acquired knowledge and experience of all the scrum activities and I would know how to apply these if I were to work in another scrum project. Regarding my goal to improve my technical skills, I feel that my skills have improved immensely, and I have gained knowledge in both frontend and
backend development. Overall I am really proud of what the team and I have achieved during this project, in terms of the website we developed and in terms of personal development and learning.

Gustav Arnesson

At the beginning of the project I set my personal goals to learning about Scrum methodology, software development in a team and programming in python. Now, at the end of the project, these goals have been reached and I have an increased knowledge in all these three regards.

Scrum methodology seems like the perfect way to work according when working with software development in a small team. May be a more correct phrase would be that agile software development is the perfect way to develop software. Since no one in our team had previous knowledge of Scrum we started off slowly. It was hard at the start, but we helped each other getting started. This also means that we worked perhaps more according to the overall agile software development methodologies than strictly with Scrum. Although we did have our Scrum events like daily Scrum meetings and retrospectives after each sprint, but most importantly we worked in an agile way that was most fitting to our purpose.

In hindsight I would like to get more tools to help us with our work. At the beginning, when we were getting started with Scrum, we were trying to use a burn down chart. This seemed like an excellent tool for keeping track of the time spent and the time left in each sprint based on the time spent and the estimated time on the items in the sprint backlog. This idea was scrapped because there were few free versions, which were easy to set up, but the free one that was found required us to report our spent time on the Trello board and this meant a lot of extra work for us. At the time we had already created a good time report spreadsheet where every person reported the time spent after each meeting, programming session or other activity. If a burn down chart would have been created from the established time report spreadsheet, we would have been forced to move the sprint backlog from Trello to the spreadsheet. This would have most likely resulted in more struggles than positive gains. We stuck to this detailed time report spreadsheet throughout the whole project and it gave us a close look at what time we had spent and had left on the whole project, not the sprint like a burn down chart.

During the project the whole team worked on many different parts of the development in accordance with the idea of multifunctional learning which Takeuchi and Nonaka (1986) suggests. This meant that a person who developed a login function could also design the login form. To keep up a good workflow we often sat together at the same place and time with the whole group. If questions aroused at these sessions, an answer was often one person away. This way we could eliminate time-consuming text correspondence and also help to further improve our team spirit.

The technical aspects of the project worked quite well. I would have liked to see a greater selection of hosting platforms, like Google App Engine or Heroku, to choose from, since OpenShift failed to impress me much. Worth noticing is that we do not have the same experience with these cloud platform as some of the persons responsible for choosing this platform for us. Reasons why OpenShift was chosen may include that it has a free version, is easy to set up and requires little configuration. How it compares to other platforms is hard for me to say, but based on my expectations it could have worked better. I would have liked to be able to make the choice myself, but sometimes external factors gives an unwanted requirement, which have to be met.

Concerning the language and framework I have found python to be a likable language, not as strict as I would like it. In comparison to Java where variable types must be defined, I might choose Java just
because of that. Using Java would have also meant less time spent learning the language, since all team members have previous knowledge in Java. Still, the way python delimiter code with indentation and also if the official style guide is followed then, the code looks really nice, which made me really like python too. Flask as a web framework was easy to work with and had helpful documentation. After this project I would prefer to work more with python and less with for example php.

To sum it up; I am very happy and pleased with this project, the team I have shared the journey and burdens with, and the technical aspects with a few changes.

**Pontus Brengdahl**

At the beginning of the project the team had a joint goal for every member to understand every part of the product and what has been done in the project. To accomplish this, it is the responsibility of those who are ahead to teach those who are behind, and those who are behind has to take responsibility for their own learning. The team also had a goal to have a working product when the course was over. My personal goal was to learn as much as possible about web development.

During the project I have been focused mostly on development of different functions and have tried to help my teammates when needed. As I had the opportunity to work on many different parts and functions I feel that I have a good idea of how the different languages work, what they should be used for and what they can do. I feel that I master the different languages and that I accomplished my personal goal. However, my opinion is that we did not quite reach our team goals as some parts of the code/project are not understood by all members of the group. I think that this is due to lack of time and focusing too much on functionality/features from the product backlog instead of slowing down and making sure that all members of the group have some understanding of all the parts of the product.

In order to reach the team goals I think that the team would have needed to focus less on functionality and more on keeping all members of the team up to speed. As this was a course and not a real project with a customer, this could have been a good idea since the goal of the project was to learn about all parts of the development process.

At the start of this project I had no experience from web development, and now I have a good understanding of how to use the different languages and what to use them for to make future problem solving easier.

The development process has been very instructive, both because this is the first bigger project I am a part of and because working with Scrum is all new to me. During this course I have gained a better understanding of how a “real” development project works and which difficulties they may face. We have had a lot of difficulties that were more or less unfixable by us as the web host sometimes did not work as we thought it would, and this has taught us to try planning to avoid or minimize the effects of obstacles of that type. I think that this is a big difference from a course and a real project as the course often use environments that are designed to allow the students to learn the content of the course instead of trouble shooting, whereas a real project may not have that extra support.

One thing missing from this project is a real product owner. This made some of the important parts of Scrum, as the product backlog with user stories, less relevant for us as we decided what the backlog contained and how the user stories were defined. In a real life development process the product owner would have managed the product backlog and the items would have been an important way to get the users thoughts and expectations to specifications for the developers, and to make sure that the items were performed in the right order (Sutherland & Schwaber, 2013). In a real project the goal or vision
of what the product should be might change over time, and this would be reflected during a sprint review (Sutherland & Schwaber, 2013). This made the process slightly unrealistic and may have lead to a worse result than what could have been achieved if the developers would have been guided by the visions and goals of the product owner.

Anna Ekelund

My goals for the project were to learn how to work with Scrum in the best way possible in a software development project, and to learn as much as possible about web programming. To do this I said I would participate in all activities and do my best throughout the process, and contribute to the joint development of the group. I was also very interested in how the group process would advance, as this was the largest project in respect of time so far at the University.

To learn Scrum was a goal that I believe would be fulfilled by participating in the working process as well as possible, as the Scrum events is a major part of Scrum. In the beginning the participation in Scrum meetings wasn’t very rewarding, as we did a lot of activities together and I already knew what the rest of the team was doing. During sprint one and two I did however start to understand the importance of iteration and what made the Scrum meeting worth completing. As we were eight people in the team it was almost impossible to keep track on what everyone was doing, and the Scrum meetings were very helpful in this matter. I also started to understand why user stories were relevant, as it doesn’t matter how good the code seem to be if it doesn’t meet the expectation of the customer. It helped me to focus on the goal of the implementation and the vision of the product. I think the interaction between customer needs and technical challenges was one of the most interesting aspects of the projects, and demonstrated well in the user stories. It was interesting because it challenged me to think both in a customer experience and business perspective and a technical perspective; which in a way also is the aim of my University studies.

As mentioned in the beginning I was expectant about the group process and how it would advance. Wheelan (2013) points out that a group will develop through five steps; Dependency & Inclusion, Counterdependency & Fight, Trust & Structure, Work and Termination. Due to a personal interest of group processes and how people interact in groups I did keep the theory in mind during the project. Indicators of the first phase; Dependency & Inclusion, are dependency on a leader and concern about safety (Wheelan, 2013). At the beginning of the project, in sprint zero, the group did have many questions about the structure of the project and did rely a lot on instructions, which I believe is related to the fact that the group was in the first phase of the group process. I felt that I wanted to get clear instructions of what our assignment was. The atmosphere in the group was polite and joyful. The second phase of the group process; Counter dependency & Fight; is indicated by that development members disagree about objectives and procedures, but is also necessary for the establishment of trust and an open atmosphere (Wheelan, 2013). In sprint zero the team had to agree about objectives of the project, and during sprint zero we had some difficulties with OpenShift. These things I believe pushed the team into the second phase of the group process. The third phase of the group process; Trust & Structure, is characterized by commitment to the group and willingness to cooperate (Wheelan, 2013). These things have distinguished the group all through the project, but clearly shown in sprint two when many problems were solved through cooperating in the group. I felt like we were started to have a flow, and I realized how important the team was for my development. I dared to give more detailed feedback to other team members. The fourth phase of the group process; Work, is distinguished by productivity and efficiency (Wheelan, 2013). After the second phase I believe the group has been
moving forward, and during the writing of this report I think we’re about to enter the fourth phase. As the group didn’t finish their work yet, the fifth phase is still to come. As the work is more efficient later on in the group process, it would have been preferable to enter the latter phases earlier. I believe group processes take time, but that maybe we could have done some things to push it forward. We could have given each other feedback more frequently, and collaborated even more during development to provoke a faster progress.

At the beginning of the project we had an ambition as a team to share knowledge within the group, to make sure we all knew all parts in detail. This was done after the sprint one but after that we didn’t do it as profoundly as the first time. As the project became more complex, it was getting more difficult to understand all parts of the project. This made it time consuming to create functionality that included several parts of the project, but also forced me to understand more parts of the project. A factor Takeuchi and Nonaka (1986) describe in their holistic approach of product development distinguish a team is the multilevel learning, where the learning is on an individual level, as well as a group level. I think that if we’d been working even harder on learning on a group level we could have learned even more. A reflection around this is that I should have asked the other members of the team more about their parts, to better understand the whole picture.

Most of the technical platform I’d never used before, which made it interesting but also difficult to adapt to all the new language and technique. I valued the opportunity to combine different programming languages and tools, it made me see the whole picture of web programming in a completely new way. I also appreciate that from having the experience of only programming simple assignments, I in this project got to program the whole web application and get a new level of understanding the World Wide Web. I thought the fact that almost everything had to be developed considering the database, and at the same time adding AJAX was challenging. An example is when I added the star rating, I had to take into consideration that it had to be added as an attribute to the product as well as an attribute to the customer who was rating it.

My goal to learn to work with Scrum, and interest to see how the group process was developing, I believe were fulfilled quite successfully. I do know how to work with Scrum in a way I didn’t know before. I’ve experienced the development of the group. My other goal, to learn as much as possible about web programming, is difficult to measure, but I believe that I do have a much better understanding for web development than I did before the project, and I’m a better programmer than before.

Claes Kallström

Five months have passed. During these five months a team of eight people have developed a working webshop while also developing as a team. To succeed in this individual goals as well as team goals were defined. My personal goal was to learn the necessary programming languages and skills to be an active and productive member of the team, a goal I feel that I have reached. One of the team's goals was to have everyone understand everything that is in the code in the finished product, a goal I don't feel we have accomplished. In hindsight I think this was an unrealistic goal when comparing to how high we set the bar for the quality of the webshop, during development different people grasped different areas better and could more easily produce good code quickly. But by having a tight knit group that worked together and had frequent meetings the knowledge flowed between the members so that everyone got a basic knowledge of everything and knew where to turn should they need help with something similar. After Sprint 1’s retrospective we all sat down and went through the code, explaining
to each other what the code did, this was an attempt to have everyone get a deeper knowledge of how the web application worked. In later sprints we had too much code to have this as a feasible option, even though the variables and functions have explanatory names I still find it difficult to understand what every line of code do so I have to trust that my teammates know what they did and that they did it well.

The way we worked was based on the Scrum Method (Sutherland & Schwaber, 2013). Using this method proved to be very effective and educative, everyone knew what was going on with the webshop and if problems arose they could be swiftly dealt with. A difference in our way of using Scrum was that we only had Scrum Meetings on Mondays, Wednesdays and Fridays. In the Scrum Meetings everyone got a chance to talk uninterrupted about their work and this was something I liked, it didn't matter if you were an extrovert or a shy person, you were forced to speak about your work and problems that had appeared giving everyone an equal chance of voicing their opinions. In Sprint 1 we were not thorough during the Scrum Meetings, only vaguely saying what we had done and this caused us to have an approximate understanding of the work that had been done. This was an issue that was brought to light in Sprint 1’s Retrospective and after that we talked more about what we had done and also how we had done it. Though everyone in the group sat around the same table and worked, most of the time individually but the close proximity to each other made it easy to get feedback and help when required so by having the meetings every other day was no issue to us. Our development was divided up into four Sprints, three of those sprints focused on coding with the last of these three focusing on improving code already written. During the last sprint we had produced so much code that it was hard to know where to improve and how, it was simply too much to delve into and unravel. This is an area where I wished that we could have had more time to sit down after every sprint and review the code, improving where we saw room for it. The part I liked the most was the Sprint Retrospective, this gave an opportunity to bring up the errors we made as a team so that we could improve productivity. An example of this was that team members were often late to Scrum Meetings, only knowing what a few of the team was doing seemed problematic so we set up a new rule: Should a person be late he/she would have to bring a snack to the team. This rule made people get to the meetings on time as well as the punishment was something we could bond over and let us get a little relaxation during the day.

The team developed as well alongside the webshop, on the first day where we got to meet each other I knew only one name out of the seven I was supposed to work with. At the start it was a bit tense but in the end it almost feels like a group of friends working on a project. This could be noted in the Sprint Retrospectives, in the last and second to last Retrospectives there was an honesty in the feedback and no one was afraid to point out the team’s flaws. Something that I value high and it gave us a chance to improve the team even more. Many laughs were had as well and the people that were punished for being late brought the most delicious snacks, cakes and muffins. This rule to bring a snack in case you were late resulted in less tardiness and a great way to bond through the common interest of coffee and cake. Without the feeling of being a team I doubt that the end product would have been as good as it is now.

As for the webshop itself, our idea was that we should be able to give this webshop to someone without prior knowledge of coding and that person would still be able to do changes to the site. We developed an Admin page that could alter many areas of the webshop, products could be added, deleted or changed just to give an example. We, the team, were the ones who came up with all the different functionalities that were added to the webshop, if we were to work with a real company it would be them who would give us functionality to add but instead we ourselves had to pretend to be
the company and try to look at this from their perspective. Adding the Admin page seemed like something a company without programming knowledge would ask for, everyone in the team had the knowledge to change in the database through coding but by doing it this way it gave us a greater satisfaction since it came closer to an assignment given from a real company. I doubt that many people have this knowledge should we work with a real company, which is why I feel that producing a webshop that everyone with basic computer knowledge can manage should have been the focus all along. With us pretending to be the company we had the power to skip parts that proved too hard and we may have missed parts that may be very important to a company since we are looking at this from a programmers point of view.

All in all I enjoyed this project and the Scrum Method. I can also say that I am proud of our achievements and the end product.

Kalle Olsson

Before this project course started I had no experience of web programming or any software development methodology. Therefore obvious goals for me were to get satisfying knowledge within these areas. But considering the size of the project, the most important goal for me was to contribute as much as I could in a large project with a good outcome in form of a great product. This is what I thought would be most useful when I have completed my whole education at Linköping University.

Without any experience of software development it was a bit confusing initially to understand how we would proceed during the creation of the product. I just expected that it would be very time consuming. Scrum was, according to me, a pretty good process tool to complete this project. According to Schwaber and Sutherland (2013) Scrum is easy to understand but hard to master. I really agree with that. The Scrum concept was a bit hard to embrace in the beginning because of the lack of competence within the subject that we had in the team. We did not really know how much we could accomplish during a sprint and how to split up functions of the webshop into parts in a product backlog. This resulted in a backlog that maybe was a bit too big for the team during sprint one. Due to the leftover functionality from sprint one, the sprint two backlog was adapted to make it possible to finish on time. I think we really learned a lesson here related to software development. Things tend to take longer time than expected due to problems along the way.

Schwaber and Sutherland (2013) claim that a team gets transparency through the daily Scrum meetings. This transparency has been a positive force within the team. My perception is that everyone has needed to perform since the rest could see the status of each individual’s current task. The meeting tool helped us to recognize if someone was struggling and maybe needed assistance. It was also useful to prevent us from doing work twice, someone who had solved a problem that another member said was challenging at the moment could alert the struggling member at the Scrum meeting.

My role in the group was pretty much to work back-end in python. Mostly I have been handling forms, passwords and validation processes. Now afterwards I am pretty satisfied with that since I have realized that I am not a very big fan of the front-end programming. There are several reasons why I am not so fond of front-end programming. Firstly I would say that I am not born with a design mindset, writing CSS code and scripts require fantasy during the development process. Secondly I think that when you do this kind of programming you never get satisfied with the result, my perception is that you can go on forever designing a form or some kind of buttons. Therefore my knowledge among the different parts is a bit unequal. But I must say that I have really improved my programming skills and I
really like python as a language. It is very intuitive and the compact syntax for me as a semi experienced user. I base my positive python opinion on my earlier experiences from programming in ADA and Java. Comparing to these languages python is straighter forward when for example declaring variables and I am also very fond of the indentation system. But without experience from programming with the other languages I think it would have been much harder to appreciate python. I really needed the basic knowledge before programming in python

I think that I have succeeded with the primary goal that I set up before the project. I really feel like I have been contributing to our finished product as much as I was capable of. Due to the lack of competence I had before my expectations were a bit diffuse, but I assumed that the project would be very time consuming and now afterwards I would say that was correct. I have learnt to work according to Scrum in an increased extent during the sprints. This is definitely a framework I would like to use again when it is appropriate to a project. Lastly I want to say that I am very pleased with the webshop and the team I have been working with.

Julia Thudén

Before we started the project, all the team members defined individual and common goals with the project. One of my own hopes was that I, when the project was completed, would have enough understanding of all the parts of the project to be able to redo them on my own. Now, when the project is done, I understand that to succeed in a project of this size, everyone can’t be a part of everything, meaning I don’t have deep understandings of all the parts in the process. Instead I have been able to learn more about the parts I have focused on which hindsight probably have been personally more instructive.

The part I put most time and effort into was to make the webpage mobile accessible. Bootstrap is a front-end web development framework (Bootstrap, 2014) we have worked a lot with while developing our product. Bootstrap components usually have built in solutions for being mobile accessible which should make this task fairly easy. However, our team had a few ideas on how we wanted the website to look on a small screen, which did not correspond to the bootstrap solution. For example, I had to build a special navbar shown only in mobile view and the filter was also rebuilt to fit a small screen better. Bootstrap is very rewarding to work with due to its ability to make the user create fairly complex components in a very easy way. This might be one of the reasons why I found it so pleasant and interesting to work with.

During this project process I have had the role as Scrum master in the team, meaning I was responsible for all Scrum events our team held. I was also responsible for all the team members adopting the Scrum rules and understanding the theory and practice. (Sutherland & Schwaber, 2013) Since neither I, nor anyone else in my team had ever used Scrum practice before, I found making sure everyone understood the theory and practice quiet difficult. Prior being chosen as Scrum master I had only attended one lecture about Scrum, which I didn’t found enough to understand the reasons behind all Scrum events being so important. Perhaps some more reading on my own would have been useful. However, I felt I had enough information to plan and carry out all the Scrum events, and the longer the process went on, the more understanding of the Scrum practice my team and I got. Now, when the software development is done and the team no longer applies the Scrum practice, I feel that I fully understand all its advantages. One example of a Scrum event importance I have now understood is the need of a Scrum retrospective by the end of each sprint (part of the project process). The purpose of a retrospective is to evaluate the work being done and come up with improvements for the next period.
(Sutherland & Schwaber, 2013). Our retrospectives began with every team member getting a bunch of post-its and started to write down things that had been good and things that had been less good since the last retrospective. When that was done, everyone presented, one at the time, his or her thoughts to the team. The post-its were put together in different categories before carrying on to the next part being writing down suggestions for improvements. Also this was done individually on post-its before presented to the group. Finally the team together decided which improvements to focus on for the next sprint. The first time we did this, I found it a bit silly and unnecessary, but after having finished our fourth and last retrospective I realized how few of those things we thought was less good during the first retrospectives that was still on the list. Examples of improvements we made after addressing the problems at the retrospective were how we handled meetings. We realized that in the beginning we were not specific enough during our daily Scrum meetings, but after we discussed it at the retrospective, it got a lot better. We also decided that we had to be clearer about when we where having a meeting and when we didn’t. Sometimes when it wasn’t clear, not everyone paid attention, which later made us have to repeat what we had said. If we hadn’t had the retrospective I believe that those problems would not have been addressed and our work hadn’t been as good as it was. The retrospectives helped us as a team to improve more than I could ever have imagined.

A final important thing I take with me from this project is the ability to work in a team that I haven’t picked myself. Most of my team members I didn’t know from the beginning and many of them I would probably never have got to know if it wasn’t for this project. When we started, I never thought I would be so comfortable and having so much fun with my team as I did. I can’t put my finger on the reason for this but I am confident that one of the reasons is that in the beginning of the project we had to sit together and agree on rules regarding for example how to work and keep in contact and so on. This was also one thing I thought to be a bit silly when we did it, but I now understand the importance of. And I think this sums up my most important lesson from this project; the importance of communication and common agreements are the most important thing while working on a project.

Erik Wallvik

So here I am at the end of the road. When I think back to the beginning of the project I’m not sure exactly what my expectations were, but I know I had high hopes to really learn a lot about web development. A goal I clearly shared with the rest of my team members. In order to reach this goal one of the things I did was, to the best of my abilities keep to the contract we formalized as a group and I think I managed to uphold it with the one exception of showing up on time were I have to be harsh and say I failed, but I think my team members were okay with. As a team we stipulated most goals around learning as much as possible, to share knowledge within the team gaining experience in all languages. A goal which I think we achieved, everybody worked with all programming languages gathering knowledge along the way and of course some picked it up faster than others, the ones who knew more took their time to make sure that I understood the new plugin they had installed or the function they had written and I’m grateful that the team had that kind of dialogue amidst its members. I’m not going to say that I know everything like the teams ambition was in the beginning, but in hindsight I think it was an unrealistic expectation for the entire group to have. The project evolved so quickly that such a demand was unfair to have in the end. Still I think that the goal of knowing all pieces of the code was the right one to set even though as a team I don’t think we reached all the way there, but it forced us to try and resulted in us knowing far more than we would if we had not had that goal.
I can't say that the fault lies with the team entirely either, because that isn't true. I myself am also at fault. The contract clearly stated that in order to reach our goal those with less knowledge were just as obligated to seek it as those who had it were obligated to share and I, as the project grew was unable to learn in the same pace. Which in retrospect I believe is ok due to the massive amount of information that was to be processed. This most certainly would have slowed down and perhaps caused us to fail our goal of having a functioning site upon project completion.

I now know some about everything and though I might not be able to come up with some of the clever solutions presented in the project on my own. As a whole I can look back at the code and given a little time understand what most of it does. I know that though I may not be the sharpest programmer in the team I feel I can say that I know have a fair amount of experience in how to program in all the languages that we have encountered throughout the course, which individually was my main goal.

One of the things that I was really worried about when we started was my capability to contribute to my team. I had no real experience in the start and really started from scratch, but I think in retrospect most of us did. I usually don't have much problem fitting into a new group of people when I feel like I have something to contribute. That resulted in the feeling that working with this group would be challenging for me and it is hard to say whether or not I have proven to work well within the team, because everybody in the group has been so great that for me it hasn't really felt like a challenge.

I had never worked with the SCRUM-method before we started and in the beginning of the project we were introduced to Tuckmans concept of forming, storming, norming and performing which seemed strange to me as it is a concept from 1965 (Hallin, Karlsson Gustavsson 2012). However it has gradually become clear that even though it is an old theory it still holds value. I learned it was a common way to work in web development and I think because of the enormous independence it has suited me well so far. Granted I haven't taken on much responsibility within the group when it has come to planning and scheduling and such, which means I have enjoyed most of the benefits with a flat organizational structure and work program and few of the drawbacks. Yet I feel that I am comfortable with the structure in case it will be a part of any future work I’ll do. What was great about our team specifically was that we smoothly dealt with our storming phase and the group quickly passed it and moved into the norming phase. Unfortunately it also leaves at the individual level wanting a proven capability to deal with this particular phase, which is important in order to reach stages of real productivity.

Perhaps the most important lesson apart from the coding knowledge I’ve acquired is that from this project I’ve learned to work with different development environments. As a team we had immense trouble with the cloud-based hosting service called OpenShift and even though I may not have spearheaded the efforts to resolve those issues I feel as if the experience we gained troubleshooting has real life applications and as a student that carries great value. It would of course have been fun to implement some of the functions that we didn’t manage to complete instead of trying to solve server issues, but it is an invaluable experience to better learn how to set up these web servers and because we had such extensive issues with this we also tried other solutions which in the end was not accepted gave us the opportunity to learn more about this subject. To me this seems like a bad requirement in contrast to just using another application and it seems odd that it would be a requirement to use a specific cloud based host. I think one of the main issues with using this specific one however was the lack of proper troubleshooting. With Github I received an answer within an hour after sending a troubleshooting question and OpenShift offered no such assistance. The answers we received from the designated server assistance the course offered could only be reached certain days and could not
always give us a straight answer on why an error occurred. However I choose to see the positive side effects it has uniquely granted my team and me personally and like a wise man once said:

“We learn wisdom from failure much more than from success. We often discover what will do, by finding out what will not do; and probably he who never made a mistake never made a discovery” - Samuel Smiles

Experience Summary – References


På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns de lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnad som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© [Tim Andersson, Gustav Arnesson, Pontus Brengdahl, Anna Ekelund, Claes Kallström, Kalle Olsson, Julia Thudén, Erik Wallvik]
Tim Andersson, Gustav Arnesson, Pontus Brengdahl, Anna Ekelund, Claes Kallström, Kalle Olsson, Julia Thudén, Erik Wallvik
Experiences from the Development of a Webshop Using Scrum Methodology 14/028