



# Magazines in digital format

**Bringing digital magazines to eBækur.is**

Guðmundur Siemsen Sigurðarson

Davíð Einarsson

Ívar Örn Halldórsson

2012

BSc in Computer Science

Instructor: Birgir Kristmannsson

Examiner: Björn Jónsson

## Table of contents

Introduction .....	3
Project description .....	3
Project requirements .....	4
Original requirements .....	4
Requirements as they stand .....	4
Project tools and documentation .....	5
Environment .....	5
Tools and software .....	5
Documentation .....	6
Programming rules.....	6
Function names and global variables .....	6
Commenting .....	6
Testing .....	7
Unit testing .....	7
System testing.....	7
Project analysis .....	7
Methodology (Scrum).....	7
Work schedule .....	9
Total work hours finished .....	9
Risk management.....	10
Sprint progression.....	11
Retrospective for sprint 0 .....	12
Retrospectives for sprint 1 and 2 .....	12
Retrospective for sprint 3 .....	13
Retrospective for sprint 4 .....	13
Retrospective for sprint 5 .....	14
Retrospective for sprint 6 .....	15
Retrospective for sprint 7 .....	16
Conclusion .....	17
Future of the project.....	17

## Introduction

This report details the account of our final project for Reykjavík University, which is being done for the company D3 Miðlar ehf. The point of the project is for D3 to be able to integrate it with their webpage that sells digital books (<http://www.ebaekur.is/>) so that they can sell digital magazines both individually and through subscriptions.

## Project description

This project has two parts: one is to create a website that will allow D3 to sell magazines, both individually and as subscriptions, while the other part is to allow D3 to become a magazine content provider. This means that content owners can upload their magazines to D3 and D3 will hold onto it and can thus handle content distribution (basically becoming an aggregated content provider). The first part will be integrated with the eBækur.is website (the project that D3 has recently finished) while the second part will be the backend for that system.

The screenshot shows the eBækur.is website interface. At the top, there's a navigation bar with 'Listar', 'Flokkar', and 'Höfundar' dropdowns, and a search bar. Below the navigation bar, the main content area is titled 'Bíó Myndir Mánaðarins' (Movie of the Month) for 'Desember 2012'. The featured movie is 'The Hobbit: An Unexpected Journey'. To the right of the movie poster, there's a description in Icelandic and a 'PDF' download icon. Below the movie section, there are social media sharing buttons for Facebook, Google+, and Twitter, along with a 'Gerast áskrifandi' (Become a subscriber) button. At the bottom, there's a section titled 'Sýnishorn úr tímariti' (Sample from the magazine) showing a grid of various digital magazines available for purchase, including 'PINDERBOL', 'FLIGHT', 'ASTFANGINN', 'EKKERT ANNAD EN STRÍÐ!', and 'LES MISERABLES'.

## **Project requirements**

### **Original requirements**

The administrator role can create publishers, third party members (who want to sell the content D3 provides), view sales reports for the entire system and each publisher. While the publisher role can insert content with minimum price, place release date on content, choose if they want their content locked with DRM and view sales report for their content. Third party role can view a list of magazines that their system is allowed to sell and gain access to fulfillment links. Buyer role can use their fulfillment links to gain access to their magazines, use the eBooks app to read their magazines, buy magazines and sign up for subscriptions.

### **Requirements as they stand**

As it stands, all the requirements that the buyer originally wanted have been fulfilled and then some. As the project progressed additional requirements were added. The ability to import from the backend system to the frontend system (eBækur.is site) has been added, using an importing program that reads metadata from the backend. Subscription related work is also done by its own program, which is scheduled to run at a set time, adding new magazines to a subscriber's magazine list. Rudimentary messaging was also added to the backend system so that users can email system admins, also an event logging system was added and the ability to view those records on its own page so that system admins can see errors and events within the backend system.

## **Project tools and documentation**

### **Environment**

This project uses Visual Studio 2010 as IDE and C# for behind the scene work (database work, classes and such) while the pages themselves use ASP.NET (with MVC 4.0 and Entity framework) and MS-SQL for the database. D3 provided the group with facilities at their offices where they supplied computers, the software needed to do the project and so on.

### **Tools and software**

POCO is used to control how the data layer is created with regards to the Entity Framework. Rhino Mock is used to mock the data layer so that it can be unit tested more easily.

Twitter Bootstrap is used to create a standardized look for all the pages and LESS is used to extend CSS to allow for more control over the CSS and thus how the sites look.

Two programs are used to do the image previewing when magazines are uploaded. ImageMagick is the tool itself that creates preview images given a PDF file and Ghostscript is what ImageMagick uses to do that. The project also uses a local class that D3 created to do some image manipulation that is important but not big enough to justify using ImageMagick for it.

The project also includes support for easy integration with Adobe Content Server for DRM locking options.

## Documentation

Google docs hold all documents pertaining to the project, like the product backlog, sprint backlog, burndown charts and schedule of each group members. When group members are not able to be at D3 then MSN is used to communicate with each other.

GitHub is used for source control, which is to say for source code, test data and third-party programs needed to run the system.

## Programming rules

### Function names and global variables

Function names are upper camel case i.e. it starts with capital letter and each successive word within the function name starts with a capital letter. Variable names are lower camel case and if the variable is in the global scope of a class then it is written `_variableName`.

### Commenting

The template for commenting functions is:

```
/// <summary>
/// summary of function here
/// </summary>
/// <param name="variable name">explanation of variable</param>
/// <returns>what the function returns</returns>
```

# Testing

## Unit testing

All major services and functions of the project have at least some unit testing in some way and every time before a build is committed with the latest changes, the unit tests are run so a broken build is not committed to the source control.

## System testing

If at any point a team member found himself with some downtime during a sprint then that team member would usually test the project as a whole. This happened a lot over the course of the project and many errors were found and fixed this way. This also ensured that the system as a whole was being tested.

# Project analysis

## Methodology (Scrum)

The group decided on using the Scrum method of agile software development. So each member served a specific Scrum role during the development of the project.

**These are the assigned roles during development:**

- Scrum Master
  - Davíð Einarsson - [davide09@ru.is](mailto:davide09@ru.is)
- Product owner
  - Gísli Þór Guðmundsson - [gisli@d3.is](mailto:gisli@d3.is) - (Works for D3)
- Programmers
  - Guðmundur Siemsen Sigurðarson - [gudmundurs10@ru.is](mailto:gudmundurs10@ru.is)
  - Ívar Örn Halldórsson - [ivar07@ru.is](mailto:ivar07@ru.is)

The project started with sprint 0, its major purpose was to get development started as soon as possible. So it started by creating the product backlog, a rough database, setting up the software needed to develop and so on. The product backlog started with all the stories the project owner said he wanted to see from this product, but as the development went along, more stories were added. The stories were assigned story points and this was done by having everyone in the group write down a number and if members had different numbers

written down for that particular story it would be discussed until a number could be decided upon (if the group was deadlocked then two out of three people agreeing would be enough). The story points had to belong to the Fibonacci sequence, since that allowed for a better way to differentiate between two stories. It was also decided that each sprint would last two weeks (except sprint 0 and 2 since they were both anomalies), so that came to a total of 8 sprints.

The group had weekly meetings, trying to meet on Tuesdays but if someone couldn't make it the rest did the meeting and informed the absent party of what happened so they could give input if needed. There were also unofficial daily meetings, so if someone had finished a story he informed the rest of the group it was done and they talked about what they were going to do next. If a major problem came up it was discussed or at least brought to the attention of the rest of the group, since someone could have had the same problem and had already solved it, also if others knew of the problem they could work around it while it was being fixed.

The group met the instructor, Birgir Kristmannsson, weekly (except when status meetings were in the week, since they could serve the same purpose) to get advice from and to inform him of the status of the project. Usually sprints ended with doing sprint retrospectives, this wasn't done for all the sprints since sprint 0 was prepping, sprint 1 was getting in the groove of things and sprint 2 was abnormal since the group was short of one member.

To see what was done in each sprint and the project as a whole please refer to the additional documents "Sprint Backlog" and "Product Backlog" respectively that come with this report.



### Work schedule

Every team members schedule was different, some could work all day while others had classes with strange arrangements of free periods, but the group tried to average at least 22 hours a week per member for 16 weeks so that comes out to about 350 hours per member.

### Total work hours finished

	Total work hours	Work hour goals
Davíð	363	350
Guðmundur	307.5	350
Ívar	358.5	350
<b>Total</b>	1029	1050

## Risk management

The point of the risk analysis was to identify the risks that might be encountered during the development of the project and to at least acknowledge them or find a solution to them before they happened.

Although there were problems setting up the DRM server it did finally come through, so in the end the content could be protected (if the publisher wants to). One problem the group did encounter was when phones tried to load magazines; the pictures were just too big. And since the app was made by D3 we couldn't do much about it, although in the future they might solve this by themselves.

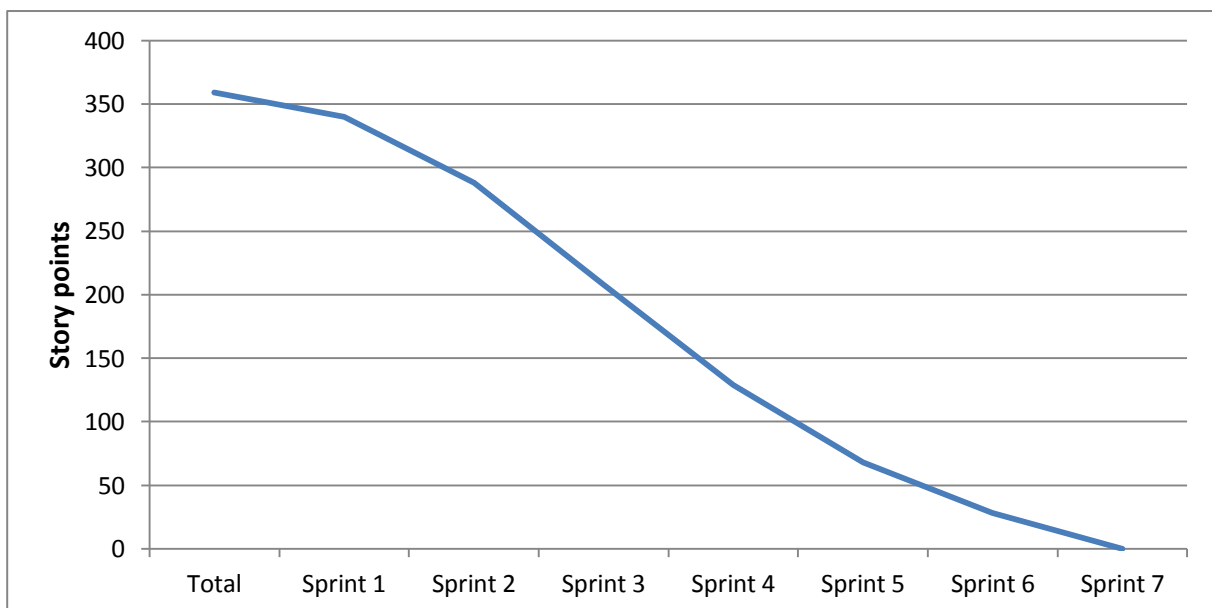
Description	Probability (1-5)	Effects	Response
There could be a problem with the content protection/DRM	2	There will be no content protection	Try to find a new content protection system
The magazines could be too big in size for the users to download through phones (has mostly to do with pictures)	3	The buyer will have to wait a long time for his magazine	Try to compress the magazines, or get better compressed magazines from the publisher
A group member can't work as much as planned due to illness or other personal issues	2	Delays work	Work more the next week in evenings and weekends
Technology issues such as computer malfunction, network malfunction, problems with source code, etc.	4	Delays work	Fix the issue, restart service
Phones take too long to load magazines since they are too big	5	The buyer reads quicker than the reader app can load	Try to find a way to compress it properly

## Sprint progression

The total number of sprints (including sprint 0) is 8, where each sprint is two weeks long (except sprint 0 and 2, which are anomalies). There was a general problem of always taking one story too many into each sprint, meaning the group never managed to complete an entire sprint (except in sprint 5 and 7) but managed to come very close each time. The only exception to this is sprint 6 where the difficulty of subscription story was severely misjudged, leading to three stories not even being started on in the sprint.

A more detailed description of how each sprint went can be seen in the document “Sprint Backlog”. The sprint burndown charts below show worked hours and those hours include both working on stories and working on other things like refactoring, commenting and bug fixing. To see a more detailed description of how the hours were spent, please see the “Team Schedule” document. The chart below shows the progression of story points over the course of the project, at sprint 7 we have finished all stories.

Release burndown chart



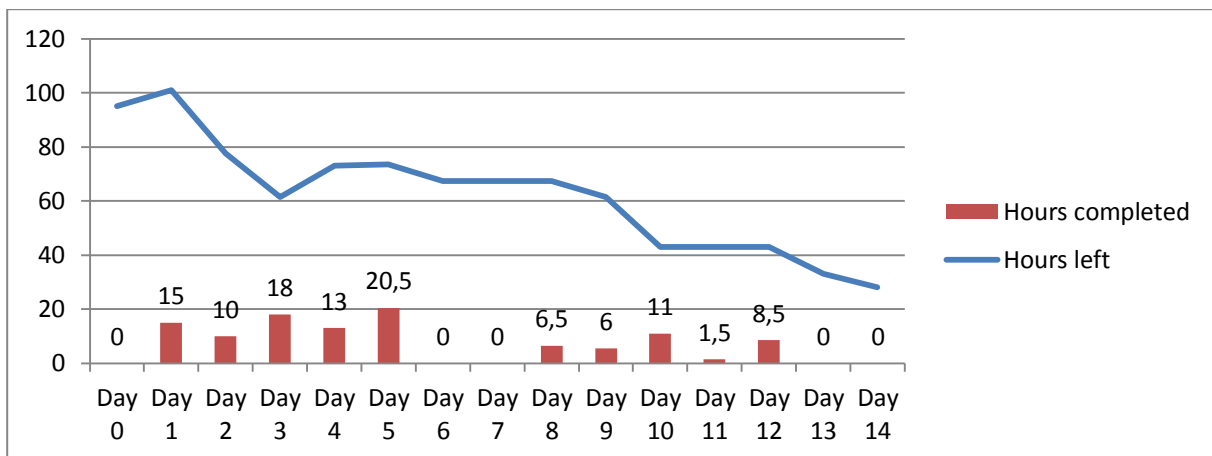
## Retrospective for sprint 0

The group finished setting up almost everything it needed to start the project; the only thing missing was proper documentation. The group also created a rough database, corresponding to how the group thought the project would be like and finished creating the stories for the backlog.

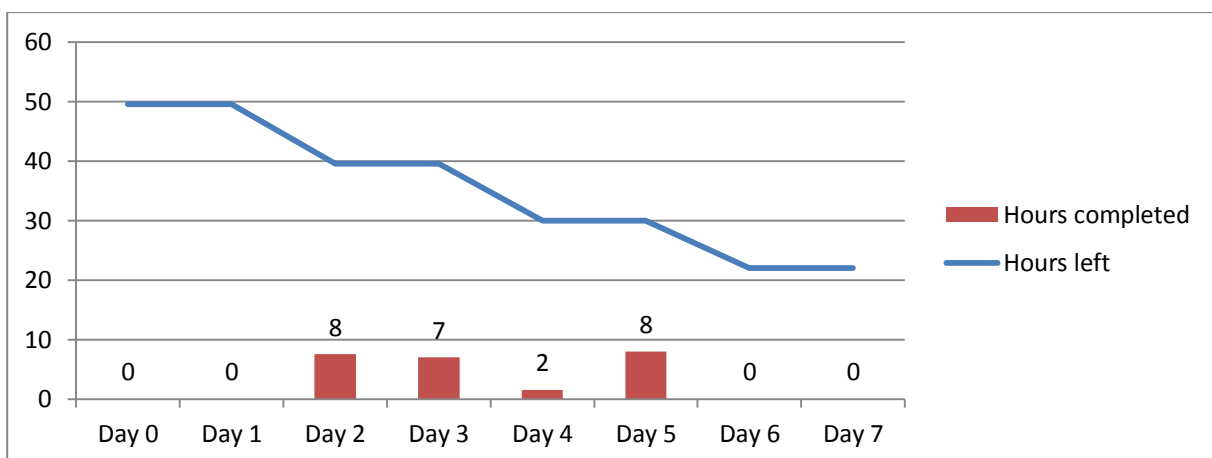
## Retrospectives for sprint 1 and 2

There were no retrospectives for these sprints since they were outliers, sprint 1 was the group getting into things and in sprint 2 the group was short of one member.

Sprint burndown chart 1



Sprint burndown chart 2

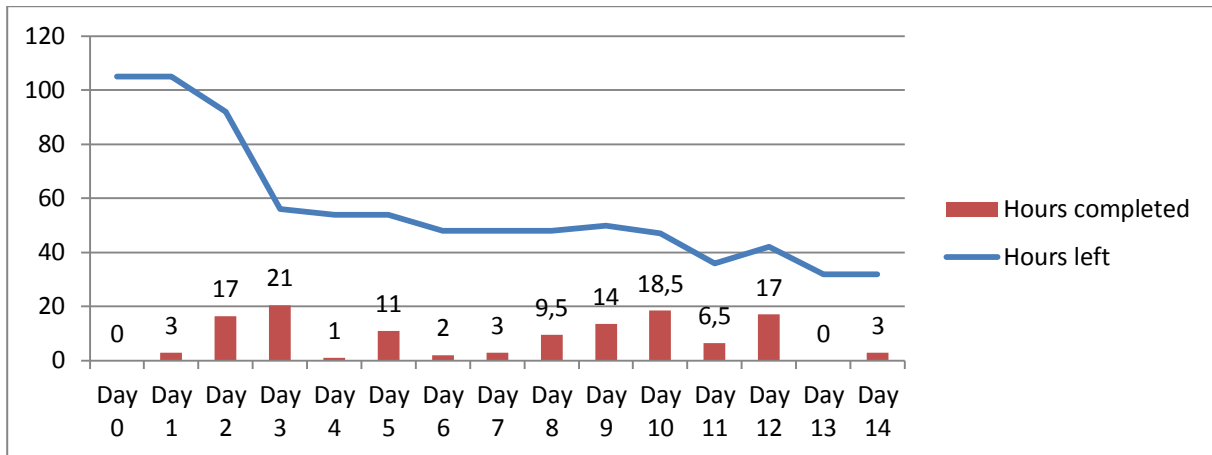


### Retrospective for sprint 3

The group had to plan the order in which it took stories into sprint better, for instance: the DRM story had to be done before the story about the fulfillment link.

The group still had not managed to finish all the stories that were assigned to a sprint and the plan was to put fewer stories into the next sprint.

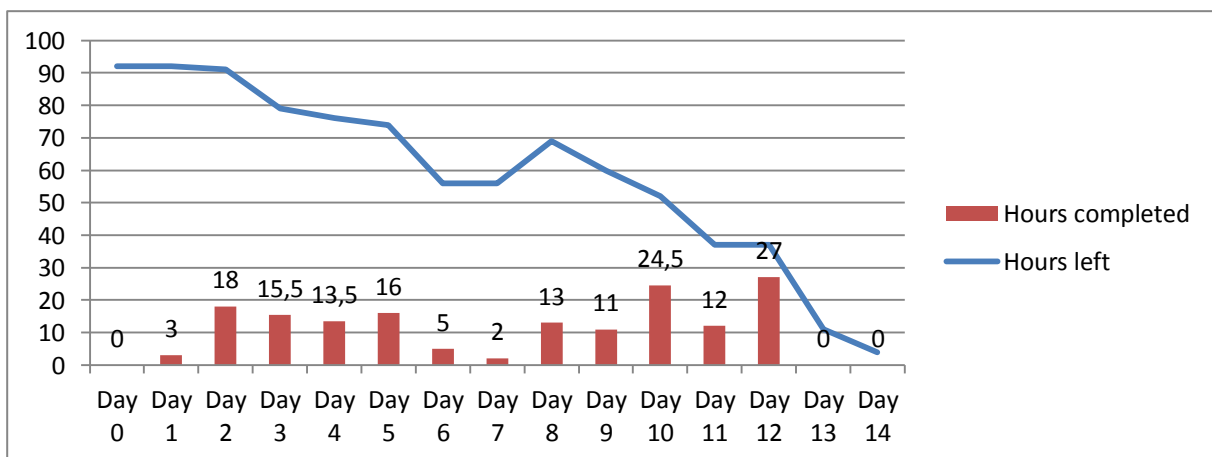
Sprint burndown chart 3



### Retrospective for sprint 4

The group had started to feel the effect of the story point's estimation that was done at the start of the project. Since then the group started to define the project better so the earlier estimation of the story points has either been too big or too low. And the group also had not managed to finish a sprint (story points wise) however there were only a few tasks left so estimation was getting better.

Sprint burndown chart 4

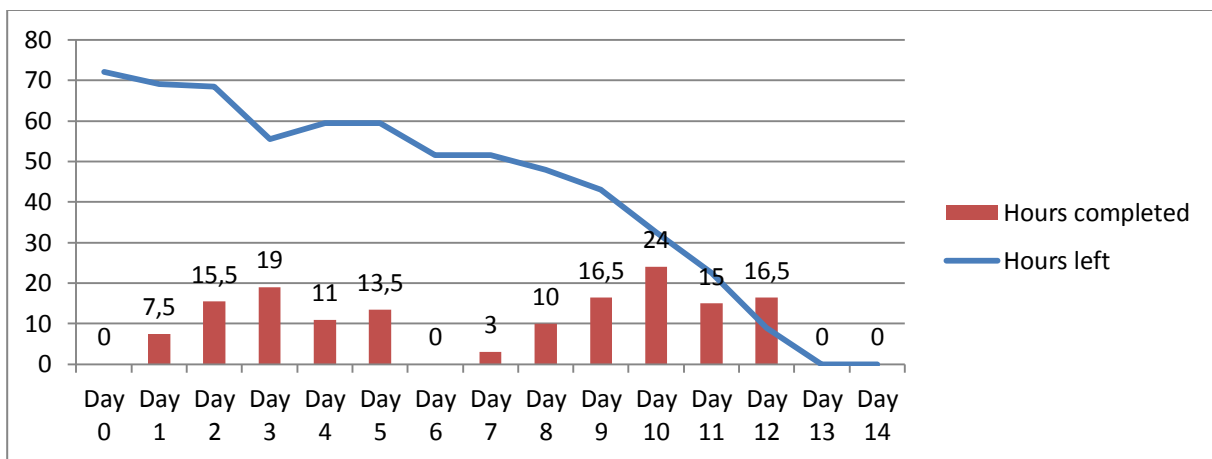


## Retrospective for sprint 5

All the stories for this sprint were finished, so that went well, also access was provided to eBook code base and it had been going well on integrating the group ideas into it.

The group was still working on the time estimation skill, which was getting better and better but could still do with some improvement (the time estimations were mostly overshoot rather than undershot but not as much as at the start of the project).

Sprint burndown chart 5

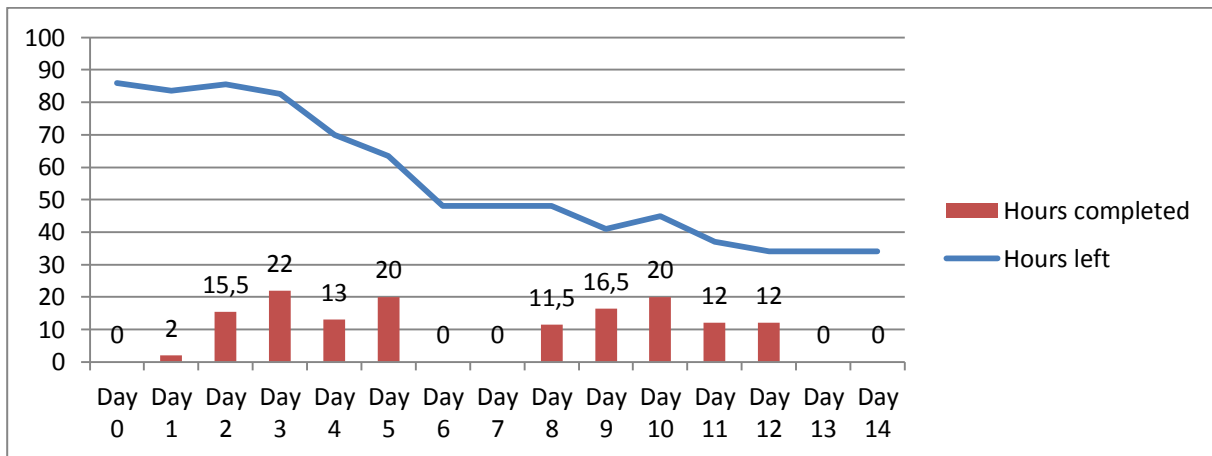


## Retrospective for sprint 6

The group didn't manage to finish all the stories for this sprint because the group had underestimated how complicated and long story 17 (Signing up for subscription) would be. This in turn meant that two people in the group (Ívar and Davíð) had to work out the logistic with this story, since Davíð was needed to work out if there could be any repercussions to any decision made (since this was being integrating into the eBook website).

The group did manage to isolate some stories that seemed to be interlocked with each other, story 17 and 15, so that Guðmundur could work on 15 while Davíð and Ívar worked on 17. So that went better than the group had thought it would go.

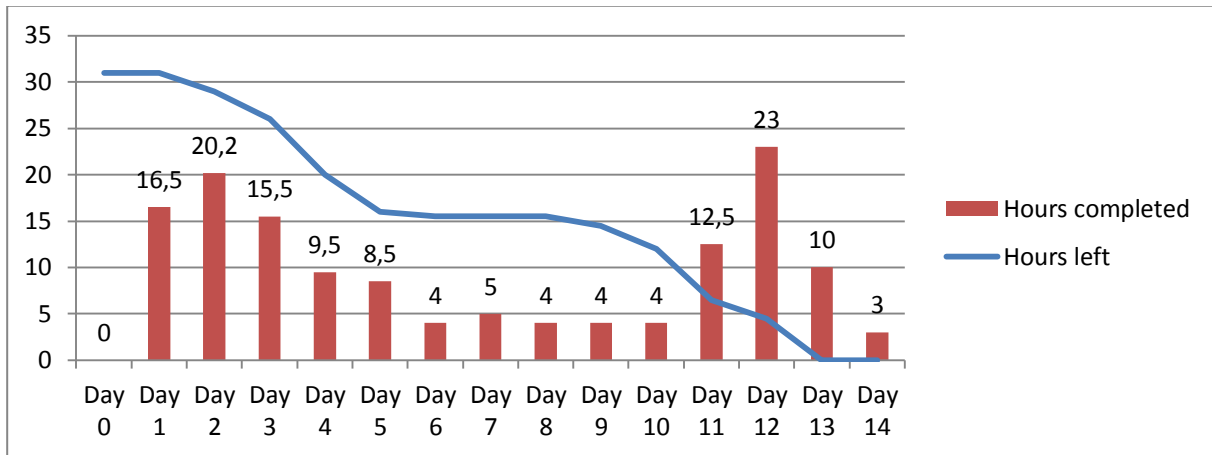
Sprint burndown chart 6



## Retrospective for sprint 7

Overall sprint 7 went well with all the stories from the product backlog completed, but the biggest problem that the group managed to overcome was the subscription part, which was left over from sprint 6. It needed a major refactoring to work properly with database transaction. Also a lot of time went into bug fixing, commenting, unit testing and refactoring.

Sprint burndown chart 7





## **Conclusion**

The end product fulfilled the original buyer's requirements that were laid out at the start of the semester and even some extra things that were not thought of at first (importer for magazines and subscriptions, preview image handling and so on). The product owner was pleased with the project, which is one of the main points.

There are of course things that could have gone better, such as assessing how big a story was going to be (i.e. the story points). This can be seen when looking over our sprint backlog where most of the time we were close to finish everything in a sprint, but only managed once to finish a sprint completely. Our ability to assess the stories got better and better as the project went on (in our opinion). We could also have done some user center testing to get a better view of how well the system could be used by end users, but by the time we thought of that it was too late to implement in such a way it would be helpful rather than detrimental.

Things that went well was for instance our adaption to SCRUM methodology, how well the group handled problems that arose during development, how as a group we were able to tell what stories were going to be a problem and what needed to be done to solve them. Also everyone within the group could voice their concerns about certain aspect of stories or how the project was going and get feedback from the rest of the group. Twitter Bootstrap allowed for easy creation of a symmetrical look for our backend portion and also easier manipulation of the feel and look.

## **Future additions to the project**

A nice addition would be apps specifically designed to read magazines instead of the eBækur apps that are designed around reading books mainly in the ePub format.

More choices for the publisher role on how the subscription deals are implemented, for instance a complimentary magazine with a subscription or the first subscription at a discount.