HAAGA-HELIA
University of Applied Sciences

Diary Thesis

Jonni Harjamäki

Bachelors Degree Thesis

March 2016

HAAGA-HELIA
University of Applied Sciences

Abstract

Date of presentation

Degree programme

Author or authors Jonni Harjamäki	**Group or year of entry** 2013
Title of report Diary Thesis	**Number of report pages and attachment pages** 30 + 0
Teacher(s) or supervisor(s) Amir Dirin	

Abstract

This thesis is result of following my daily work. In this report I discuss my daily activities and try to give some grounds for some of the solutions I have made.

During this process I worked on a single product. I am part of a development team that is working on the said product. When I started writing the thesis I had already worked for few months. This thesis is a look on my personal development as a developer. It will provide insight on how my skills have progressed.

As a result of this thesis I was able to take a look on my progression as an developer. I can surely say that I have grown as a developer. My skills in acquiring and using information have progressed immensely.

Keywords

Table of contents

1 Introduction

I will write this diary thesis starting from 21.3.2016 and continue for 13 weeks. In this diary I will discuss my daily duties as a Junior Developer. I will try to analyze my professional progress as a individual developer and as a part of team and a company. I will try to discuss my solutions and give grounds to them

1.1 Background

My current day to day duties include development, planning, bug fixing and deploying builds to our testing environment. We are a Scrum team consisting of Scrum master, The developers, Product owner and testers. Most of the work is usually finding bugs and developing new features. The application is Financial Management system. Our team doesn't have separate Back-End or Front-End guys so usually week may consist of UI, server-side or Database work depending on the tasks on the backlog and sprint goals.

Most of our UI is made With ASP.NET Razor syntax but also a lot of AngularJS, JQuery and A custom Widget framework is utilized in views. Our Back-End duties are handled by Entity Framework which handles most of the transactions to database. Sometimes I have to use PowerShell scripting for the deployments.

I have to know all of the fore mentioned frameworks and technologies to succeed in my work. This means also that I have to have sound understanding on how things work together. It is crucial to understand how the different layers and modules work with each other in order to succeed.

1.2 Skills Needed

I have to possess coding skills in previous chapter as well as finding information fast and utilizing it. Good problem solving skills have proved to be invaluable. Good communication skills are very crucial as I have to communicate with the Scrum team.

My biggest resources in finding information and educating myself are MSDN pages and some blogs, forums etc. Most of what I learned during working I have picked up from there. There are also some online “schools” such as Udemy which I use to learn new things.

1.3 My working environment

We work in an open office environment our team currently consists of 4 members and our communication is done in English. We use SCRUM as our methodology and it mostly defines our days and sprints. Product we are working with right now is intended to be either Cloud based or installed to customer premises.

1.4 My goals

I will evaluate my learning and professional development during these 13 weeks. I will also try to take a look at my daily work. By the end of this thesis project I wish to have a good overall view of my progress as a developer. Hopefully I will be able to evaluate my professional growth.

2 Starting point

We are a Development team consisting of developers, product owner and testers. We are agile team and we use Scrum as the methodology on which we build our daily work. Currently we have three week sprints after which we usually have a demo on new features.

2.1 The daily duties of developer

Days usually start with dev team having a daily SCRUM where every member tells about three things:

- What did I do yesterday?
- What will I do today?
- Is there any impediments?

After daily we either set out to duties for the day or discuss a hot topic such as a design workshop for a new feature. We organize our work with our backlog where we have all of our tasks and work items. Tasks can be roughly divided into two categories: Development tasks and Design tasks. Development tasks mostly consist of either bug fixes or developing new features. Design tasks can be anything from investigations to meetings with people outside of our team.

Developing new features is usually more demanding task taking anything from couple of hours to a week. These tasks usually include implementing a new feature based on requirements. Bug fixes can range from a simple Configuration file changes to refactoring a class or a module from the application.

Sometimes deployments and handling of problems in different environments also are part of our duties. Deployments are usually easy as it is just starting a build. But when something goes wrong it is usually bit more of a challenge. Same goes for other environments. If there are problems they might prove to be a whole week of work just to fix a small issue.

2.2 Interest groups in the workplace

Internal

- Product owners
- Stakeholders
- Team

External

- Customer companies

3 Diary Entries

This part will hold my diary entries from the 13 weeks. Here I will analyze my own doings.

3.1 Week 1

3.1.1 Monday 21.3

Today was the sprint planning day for our team. We planned for the next three weeks. We are a relatively new team so our velocity is not steady yet and our goal is now to pick up just enough tasks to have little bit of pressure towards the end of sprint.

Our planning day starts with meetings with Product Owners and stakeholders to prioritize the items on backlog. In that meeting team decides how many velocity points they take into sprint. After that we have a group session where we take all the items on sprint backlog add tasks and hours to them. After that we check all the items with the team so that we all agree on the amount of hours.

After sprint planning we started to work on the items on the backlog. I picked up task concerning incorrectly saved data on view where angular is used for data binding. For some reason the angular controllers did not save correct data to the database. After some investigation I found out that a lot of the logic behind and the view should be refactored a bit. And some new functionality should be added.

I spent most of the day Debugging and digging the code on how
Everything works. After a while of debugging and digging up on all the layers I had to ask one of our testers for a quite a lot of information.

3.1.2 Tuesday 22.3

Today I continued on the task I picked up yesterday. I still had a quite a lot of stuff to investigate. After a half a day of investigation and asking for more details from my coworkers I finally had a solution for my task.

While fixing things I accidentally recreated a bug from another Item on sprint backlog. So after asking my coworkers I reserved that item for later. At the end of the day I was nearly finished and all there is left for tomorrow is brushing up and doing some testing on Dev environment.

3.1.3 Wednesday 23.3

Today I completed all my work on the UI layer. I started to work on a problem which appeared on removing a property from one of the Database models. This created multiple issues on all layers of the application due to the property being used as a search term for many of queries. Most of the work included just removing any trace of the property from different methods but for some things I needed to rewrite some LINQ queries.

3.1.4 Thursday 24.3

Today after daily scrum we had a Backlog grooming session. We have grooming sessions once a week on Thursdays. Usually we go through some of New backlog items make sure that everybody is on the same page and then give it effort points from 1-13. 1 being a small task and 13 being a large task taking several days and including large amount of work.

After grooming I had small session with one of the testers. There was some issues with email sending. We had setup a scheduled task to run every day. There was no actual problems. Day proceeded as usual I did a small bug fix.

3.1.5 Week Analysis

This week was me working on a one big PBI for the most part of the week. During the whole week I had to do lot of testing for the changes I had been working on. I did some refactoring on whole lot of classes and methods. Workwise this week was pretty standard.

A lot of the work this week included designing a solution for an existing problem. This work included lot of work in database and back-end logic and some minor fixes in front end layer. All in all this was a learning experience on how everything works together as one. In the end everything works as a whole unit.

Most of this weeks solutions have to do with working on angular and using LINQ to fetch data that is needed. Having not used LINQ that much I needed to do some research. MSDN revealed upon searching that there is two ways of doing the LINQ. Method Syntax and Query Syntax (MSDN 2016a.). I chose method syntax over query syntax because it lets me to do single queries in one line. Comparing it to query syntax which may span over few lines even for simple SELECT operation.

3.2 Week 2

3.2.1 Tuesday 29.3

There was still small issues I had not foreseen when I modified the DB model. One of the testers had pointed out that one of the views did not work as it should. Therefore, I spent couple of hours on fixing this small issue. Then I picked up a PBI concerning Angular validations when having a URL from external service. I found out that not all of the Permissions for users were loaded in Controller.

At the end of the day I did a deployment to our testing environment with new features etc. I found out that for some reason I did not have correct permissions on our build server.

3.2.2 Wednesday 30.3

I finished the work on yesterday's bug and picked up a new PBI. Now there was a problem with exporting data to our core system. The problem was with some of the exported fields were incorrectly fetched and saved from DB. Fixing that took most of the day. Most of the fix is just mapping the entities properties between each other

3.2.3 Thursday 31.3

After Daily scrum we had a Backlog grooming session and design session. There was a new feature we had to design the implementation for. We groomed couple of items so that we have enough for next sprint. After that we started brainstorming the design. We came up with a solution and one of the team members started to implement it after the session. My day proceeded with small bug fixes and testing those fixes.

3.2.4 Friday 1.4

April Fools. There was a new issue, which had to be fixed for next release, which is right around the corner. Problem was popups, which were created with Kendo UI framework. Popups could not be opened second time and there was load of problems with filtering the data on the grids. I found out that some of those were created so that they simply cannot be opened twice. I ended up rewriting lot of the code. After a day of trouble shooting and testing I had fixed the issue.

3.2.5 Weekly Analysis

This week was great for me. I feel like I got a lot of things done and I learned some new things and learned some things about our technology stack and learned quite a lot about the architecture of the application. This week felt like a standard week with bit more complex tasks and bit more stuff to go over. In addition, release of the next version pressures the team noticeably so we all have some stress.

3.3 Week 3

3.3.1 Monday 4.4

Release is tomorrow so everybody was working really hard to fix all the little bugs and issues. In the daily team decided to omit some Sprint Backlog Items that were for the next version in favour of fixing bugs before release because of request by product owner. Day was full of bug fixes that the tester kept reporting to the team. There is still some bug fixes left for tomorrow but mostly small issues

3.3.2 Tuesday 5.4

After daily we got email about two urgent bugs found in testing. I picked one of those and started to work. There was a time strict limit for the fixes to be ready after noon. Team is bit stressed of course but eventually after few hardworking hours we got the bugs fixed and package out.
Rest of the day went with investigating a new feature about integration with another product.

3.3.3 Wednesday 6.4

After daily we had a grooming session. In the grooming we didn't groom that much because we already have two sprints worth of groomed stuff. I started reading all the documentation I had gathered about the integration. I installed instances of the service to the test environments. I started writing documentation for our dev team about the Interfaces and communication scheme etc. writing documents took most of the day.

3.3.4 Thursday 7.4

I wrote a document on client service and how it communicates with our product. I also wrote instructions on how to install all the needed components of integration to Dev environment. I also added needed files to source control.

3.3.5 Friday 8.4

Today is demo day so in daily we planned what we wanted to demo today. There is small issues on our environments that need to be fixed. SQL server connection times out for some reason even though the connection is established. We demoed for stakeholders and POs what we did in the last sprint and what we will do on the next sprint.

3.3.6 Weekly Analysis

This week was fuelled by stress. Lots of hard work and dedication was needed for this week. But in the end it payed of we had our release ready and all of the minor and bigger issues reported were fixed. This week also marked the end of the sprint.

I started to gather information on integration with an another product. This proved to be a horrendous task with a lot of reading and writing some developers notes to my colleagues.

This week I had to write some documentation. As I was really a noob to this kinds of work I had to research it a bit and stumbled upon great resource which helped me to write the documentation. According to that there is four essential parts to it: Title, Section, Steps and Appendix. (Grier 2016.) I followed a similar layout for that and did write the document. But more importantly I followed the steps defined there. Started by questioning who am writing for? The developer team. Then I collected the all available documentation. After that wrote the outline and first draft. Gave the document to my team mate to read through then I reviewed and reformatted.

3.4 Week 4

3.4.1 Monday 11.4

The day started with planning meeting with all the stakeholders and team deciding how many points we take into sprint. We looked at last sprints velocity and took items into sprint accordingly. After meeting with all the stakeholders' team takes their own meeting, where we make tasks for each PBI and go through them to make sure everybody is on the same page. After that, I picked up a task for substituting users. This task is actually quite large and needs a lot of work.

3.4.2 Tuesday 12.4

I started to work on the user substitution by looking at what we already had. Current functionality does not support multiple substitutes, so that had to be fixed. I had to Remove client side validations and server side validations and modify the substitute saving so that multiple substitutes can be had simultaneously. Most of the work was easy just to remove code but somethings had to be modified heavily just to make things work

3.4.3 Wednesday 13.4

I had to modify the register to save and to fetch a list of current substitutes. I still had to do some work to get the saving working for substitutes. After that it was just a matter of modifying LINQ syntax to fetch multiple records from database. These changes needed a lot of testing since there may be some things that break when done incorrectly.

3.4.4 Thursday 14.4

Today we skipped grooming because we already have enough items for the next two sprints. After daily, I worked on cleaning the views for substituting. Most of the work was just to make sure that needed functionality is there and that all the records are saved and fetched with no errors thrown.

3.4.5 Friday 15.4

Today I started to work on how the substitute can handle things on substituted users' behalf. This included changes in email sending and many other things as well. Email sending for substitutes was an easy thing to do as it was just making sure that all the substitutes are fetched correctly and get an email.

3.4.6 Weekly Analysis

This week I worked on a big item that still is midway through and needs a lot of testing after I am done with it. But I hope that in doing this I don't make that many mistakes. Otherwise, this week was easy going and not too rushed.

Like couple of weeks ago, I worked on many LINQ statements. However, this time I opted for Query syntax since it was easier to read when there is complicated subqueries and where clauses. This time the readability was my first choice because it is really a complicated query. Readability affects how easy it is to maintain the code later on or change it to do different things. It is also much easier to understand code that is written to be readable. But readable does not mean that it is simpler or that it takes few lines. Readable simply means that the purpose of given code is to be understood easily.

Writing code to be readable means that in long run it will be easier to fix and find bugs. Bu first of all it is also easier to implement new features in future and prevent malfunction(Crouch 2016.).

3.5 Week 5

3.5.1 Monday 18.4

Today I mostly continued on working with the last Friday's changes and I am waiting another team member to fix his issues before I can continue my part. Since his and my work are bit intertwined.

3.5.2 Tuesday 19.4

Today I investigated how to get the substituting to work so that substitute can handle things on behalf of the original user. This requires many changes in JavaScript and in Controllers and Manager Layer.

3.5.3 Wednesday 20.4

I worked on how to get the substitute to see the documents of the original user.
This included a lot of small issues to be fixed and change in data access layer.
I modified a big LINQ statement to fetch data for substitute. It only needs small modifications anymore

3.5.4 Thursday 21.4

I was sick

3.5.5 Friday 22.4

I got the query to work. Next step was to modify the view to show the data in a correct way so that it is filterable and so that everything works, as it should. That was mostly the easy part now what there is to do is to make the substitute able to work on behalf of the original user. I continued my investigation on this and at the end of the day, I was not close to a solution.

3.5.6 Weekly analysis

This week felt for me that I did not progress with my stuff, as I would have wanted.
In the end it is a big item to work on so that is reason enough why everything seems to go slow. However, for most part after this week, I am almost ready with this and there is only small clean up to do.

This week involved working with different layers of application. We Utilize the ASP.NET MVC framework. MVC means Model, View and Controller. In addition, each of these things have their unique purpose. Model defines the data, View is used to Show the data and Controller gives the read, write and update methods. That is of course overly simplified view and it there is more to it than this. This framework also allows for easier testing and separation of tasks. (MSDN 2016b)

This makes it easier to modularize and build the application to support separation of concerns. Separation of Concerns simply means that every part or module of application have their clear and well-defined responsibilities.

3.6 Week 6

3.6.1 Monday 25.4

Today I fixed and completed the clean up for user substituting and added some logic to enforce rules in substituting. Rules are such as substitute has to have same or higher privileges than substituted user. The work itself was trivial but due to Kendo framework error, handling had some issues.

3.6.2 Tuesday 26.4

Added many small fixes to the substituting. Most of them were just cosmetic changes in views. After that, I took a task on investigation of refactoring message sending. As of now there are five different emails sent from the system. After a brief, while a realized that it will take just too much time to refactor message sending.

3.6.3 Wednesday 27.4

After daily I noticed that after merging two branches all of my changes were now broken. Most of the substituting is now useless. This is really frustrating. I spent whole day on investigating what exactly was the issue. On the other branch there was just too much changes in core logic that affected the substituting.

3.6.4 Thursday 28.4

The yesterday's conflicts were caused by auto merges and the sheer amount of difference in the branches. After reverting and fixing some things the guys who did the big changes in the other brand took the task. Since they know more about the changes in the engine they should work on it. Lesson learned never do any big changes that might affect each other in different branches. I fixed a bug in when document comes in if there is errors it gets wrong date. The fix itself was somewhat tricky. The error comes up in a completely different place than where the fix was supposed to be.

3.6.5 Friday 29.4

After daily scrum, I started to get ready for demo. For demo, I deployed the latest sources to our testing environment. However, something was wrong and parts of the application were unavailable. This was due to recent changes in configuration files. We have many different configuration files for each environment we are deploying. There was a configuration for each of the web apps and each service. But we changed it so that only one place has all of the configuration files and these are linked to other places. This created some problems since the some configuration files were not linked. However, thankfully that was easy to fix. In demo, I demoed some changes done in last sprint. After demo, we had the retrospective for this sprint. After retrospective, I still had some time left for day so I started to work on a bug.
The bug was just a small UI fix, which was easy to do.

3.6.6 Weekly Analysis

This week I learned an important lesson on my trade. If working on big changes simultaneously don't do it on different branches. But all in all this week was filled with lot of frustration at the merges. But in the end everything was managed and the sprint ended well.

Branching is a big part of version control. It is good to know when to branch. Branching supports parallel development and release management (MSDN 2016c)

3.7 Week 7

I was on holiday

3.8 Week 8

3.8.1 Monday 9.5

Today I started to work on setting lazy loading of. In Entity Framework it means that when entities are loaded related entities and properties are loaded only when they are needed. Because this may allow multiple queries to database which may affect performance negatively. We decided to eagerly load all of the needed data. This means we have to go through all of the repository classes and debug to see which properties are touched. I started to work on the task debugging and going through.

3.8.2 Tuesday 10.5

I continued my work on this task. Since this is going through a lot of code this takes me at least several days. Only few of the classes needed to be modified. Also did some clean-up on code that was no longer in use.

3.8.3 Wednesday 11.5

Still going through the repositories to find out which need to be modified. I have shared this task with other team member who goes through other projects. There is still thousands of lines of code to go through. I managed to finish one part of the task and moved on to the next repositories.

3.8.4 Thursday 12.5

Same stuff different day. Still going through the repositories. There are quite a lot of the repositories to go through and some controller or other actor of course references each repository. This means that this is quite a strenuous task. Nevertheless, in the end the performance benefits will be worth this

3.8.5 Friday 13.5

Managed to finish all of my work before Daily scrum. For change, I picked up a task related to improving search functionality. This means adding a new search criteria and removing some unnecessary ones from the UI.

Seems like adding search criteria is not that easy of a task I need to add a new property to loads of places and since the search result is a database view joined from multiple tables I need to modify that and create a migration for it. Most of today I spent fixing the UI and adding the new search field to UI.

3.8.6 Weekly Analysis

This week was mostly working on the lazy loading stuff, which is a big change in how the application works under the hood. I am quite positive that we are going to have issues regarding to these changes. All in all pretty standard week nothing that big or new.

We use the Entity Framework Code first in the application. This means that database is created on the fly based on the current configuration and models (Entity Framework Tutorial 2016).
In this case, we do not have to create any kind of database design or configuration (Guthri 2010.).
Database tables are created based on objects, which eliminates the need for designing and modelling the entities, and database. In entity framework dbcontext class (MSDN 2015) handles the database access. Context represent the runtime values of the data in Database. In this context, those objects are called entities. Entity framework maps the objects and relationships to correspond with the database.

3.9 Week 8

3.9.1 Monday 16.5

Today I finished the tasks with the new search functionality. Most of what was left was just standard clean up. Then I picked up tasks with the lazy loading that one of the team members dropped. These tasks are just going through everything that accesses the Database. As a task it is pretty exhausting and boring and takes time to go through. Nevertheless, we have to be sure to go through everything because when we disable lazy loading we have to

3.9.2 Tuesday 17.5

Still doing the task from yesterday. Added some includes and went through a lot of code just by reading and trying to understand.

3.9.3 Wednesday 18.5

Finished the task with lazy loading. After daily I picked up a task that I knew for sure was fixed. After a While of investigating turns out that due to auto merge when we merged the branches, my changes got lost, for some reason. Therefore, I re-added that and checked in. After that, there was a task that required a bit more attention. There was a bug in the changes done in lazy loading.

3.9.4 Thursday 19.5

Continued the yesterday's bug fix. It took a while to hunt down where the bug was coming from. Eventually I found it to be one of the repositories that was handled. A one include statement was missing from a LINQ query that resulted in a null property for an object. After fixing that, I extensively tested that everything was ok.

3.9.5 Friday 20.5

Demo day. After daily scrum, we decided on what to demo and we split the features to demo among the team. I got to demo a bit of the lazy loading things. Since most of the change is under the hood it was more of a telling what we did instead of showing. Of course there was some improvements in speed but not that noticeable. We also had a productive retro where we decided to change our way of working to be more effective as a team.

3.9.6 Weekly Analysis

This week was marked with the disabling of lazy loading. It was a big and exhausting task that resulted in bugs and some issues. But those were easy to identify and easy to fix by adding include statements to LINQ. Next week is sprint planning.

We chose the eager loading because we have quite comprehensive entities. Lazy loading loads reference of the related entities with it.
(MSDN 2015b). But with lazy loading you load only the needed entities and have full control on what you load and what data you access. This of course creates a performance benefits if used in correct setting. Using eager loading you can manage the needed queries. And in our case the use of eager loading brings performance benefits compared to what the lazy loading brings for us.
Since we have already many calls to the database it is better for us to use the eager loading. In our context we use the eager loading to load big chunks of data only when needed. Since some of our entities are linked and might have multiple references to other entities. It is safer to go through all and load just the needed data.

3.10 Week 9

3.10.1 Monday 23.5

Sprint planning day 1. This sprint marks the leaving of one of our senior members, which means that we will lose huge amount of knowledge. But nevertheless we decided to plan a sprint with less points taken in than in previous sprints. First we had a meeting with the PO's on what to take in to the sprint. After that, we went through every item and added tasks to those. After planning I started to work on a task, that I was sure was already done. For some reason the fix I had earlier done was not in this new branch after merging. Probable reason was that the particular JS file was worked on during last sprint so much that when merged the changes were omitted from the older branch. Nevertheless, it was just a few lines of code and testing to be done so nothing too hard.

3.10.2 Tuesday 24.5

Today after daily, I changed an icon in UI. Task that should have been a walk in the park was a little more complicated. It involved writing some CSS and making sure that the new icon set matches the old one. If there would have been errors it would have been a tremendous task to fix. Luckily, all of the icons were okay. After fiddling around with that task, I picked up a task to work on fixing another Minor UI thing which causes a really annoying problem. Problem was that after a certain row has been handled in grid then it breaks in to a new line. It was small fix in CSS file. After fixing these small issues, I picked up a task concerning errors in import. There is error when importing documents. The imported documents with some additional data do not comply with same rules as ones without. Therefore, I started to investigate this.

3.10.3 Wednesday 25.5

After daily, I continued the task from yesterday. After a while, it occurred to me that the task was bigger than initially thought. We initially thought that the task was easier than what it was. It was tough to find the place in code where stuff goes wrong. I spent whole day debugging and going through the code.

3.10.4 Thursday 26.5

After yesterday's debugging I have located the source of the problem. Instead of being in one place, it is in at least two places. After staring at this problem for a while, I came up with a solution to fix it. First fix was trivial to do but the second one needed some LINQ-magic. It took me a while to write the LINQ statement. However, after I got it and now works flawlessly.

3.10.5 Friday 27.5

After daily, I started to do a clean up and testing the fixes I had done. Testing I did quite extensively to make sure that no bugs are there still and that these fixes did not create any new bugs. After a while of testing, I was requested by the tester to check one thing that was somehow loosely related to this. In the end, the thing was related but not related to be from the same Backlog item.

3.10.6 Weekly analysis

This week marked the start of the sprint and I spent most of the week working on the import problem. I managed to do a quite complex LINQ statement that I am quite proud of. Other than that it was pretty standard week.

3.11 Week 10

3.11.1 Monday 30.5

There was a bug with some Default settings not being enforced for Documents. Issues were due to a problem in Lazy Loading. After confirming with testers that this was fixed I Continued to fix issues with earlier fixes on the import. Apparently, if there was errors in the document and a user corrected those it went through a completely different route and was not updated as I fixed earlier.

3.11.2 Tuesday 31.5

I was sick for rest of the week

3.11.3 Weekly analysis

This week was cut short by getting a flu. Again, there were some problems with the Lazy loading. Most of these issues are caused by some references to properties that are not used but are not fetched from the database. In addition, this causes issues with null values.

These problems thankfully are usually easy to fix just by removing the references to unneeded properties.

3.12 Week 11

3.12.1 Monday 6.6

I felt still a bit under the weather. I started to work with a problem in integration of an external Dimension checker. There was an issue with it handling already handled rows in a document. The fix was not as easy as you would have thought and required a lot of Debugging. After fixing these issues, I started to work on implementing same kind of logic in to the back end. It is a tremendous task.

3.12.2 Tuesday 7.6

It is a tremendous task to get the dimension checker to work on back end and requires creation of new class. In addition, requires a lot of fixing here and there. Added some small fixes for front end also.

3.12.3 Wednesday 8.6

Continued on implementing the dimension checking. Now I am able to get the results into our system. After a lengthy discussion with the PO about the requirements I started to work on the implementing it according to requirements. It just required some really complicated coding

3.12.4 Thursday 9.6

Ran in to big problems with my development environment. For some reason a part of the application that monitors Files for import does not work for me at all. For everybody else it works. Not me. I spent whole day trying to fix this issue with no success

3.12.5 Friday 10.6

Still same issue. Today is demo day so that I have to forget for a while to prepare for demo. And after demo we have Retro so I didn’t have lot of time to fix the issue. But I still managed to fix the issue by the end of the day. Now I have more issues.

3.12.6 Weekly analysis

This week was frustrating for me. The problems in my development environment really stopped me from doing work. I have an interesting task that is blocked by issues on top of issues.

The task that I have has to do with integrating to another application, which is quite interesting and can prove to be a great learning opportunity.

3.13 Week 12

3.13.1 Monday 13.6

Today we have the sprint planning so I do not have that much time to do fixes for issues I had. In the sprint planning, we decided to concentrate on getting next release out. Therefore, this means the whole sprint is just fixing bugs. We have had couple of bugs due to the changes in last sprint and most of them are small and annoying.

3.13.2 Tuesday 14.6

Today I continued to work on the issue I had with a fellow team member. We found out that the problem was due to our authentication context dropping for some reason. We do not know why it drops yet only when. Dropping is due to process accessing external resources and that is when the context gets dropped.

3.13.3 Wednesday 15.6

Today we finally found out what the problem was. The problem was due to parallel processing and id of the thread running. After fixing that I finally could continue to implementing the changes I had worked on earlier.

3.13.4 Thursday 16.6

Finally I am in testing phase with the external dimension checker. Everything seems to work fine except for a small bug I was able to catch while debugging. After fixing and patching it up everything seems to work as it should and I am very happy with my work.

3.13.5 Friday 17.6

After daily, I picked up a task from backlog. It was a bug with user rights and authentication having a "small" issue. After a while of investigation, I found out that it was far more complicated than initially thought. There was no authentication for that view. There was scraps of really sloppily implementation that granted access to everyone with right URL which is kind of a big security issue.

3.13.6 Weekly analysis

Finally got the implementation to work with the External Dimension Checker after having some issues initially. It has been a long and frustrating week. I learned a lot of how the authentication works for different parts of the application. Our application is bunch of webpages in front end and windows services and programs running on back end. I have been wondering for while how the authentication works for the application as whole.

This time threads and parallel processing caused some issues to me. Issue was that there were multiple threads running and when another thread was used, the context was lost. This means that some very crucial data resolves to undefined.

3.14 Week 13

3.14.1 Monday 20.6

Fixed some issues with mappings between Dimensions, Fix itself was a small string comparison between names of different dimensions. Fixed also issues with date handling on multiple places to be in uniform with each other.

3.14.2 Tuesday 21.6

Did some builds and merges with the different branches in source control. Sometimes merges cause huge clashes between different versions. Thankfully, this time there were only few issues to correct and most of them were easy to fix.

3.14.3 Wednesday 22.6

Fixed issues with user rights. There was an issue with users not having enough rights seeing stuff they should not see. This was due to poorly implemented JavaScript code

3.14.4 Thursday 23.6

Holiday

3.14.5 Weekly analysis

Write something more here. This should be longer

3.15 Week 14

3.15.1 Monday 27.6

I have been informed that there are issues in the demo environment and that it is very urgent and I need to address it as soon as possible. The issue is with our file monitoring service in one of the demo environments not starting at all. It throws a no access exception. I have to co-ordinate with so many people that it is making me crazy.

3.15.2 Tuesday 28.6

Continuing investigation with the issue. Picked up a task with file stream opened but not properly closed. This causes some memory leak issues and may affect performance fatally. I will continue tomorrow, as it seems that the issue with demo environment is not going forward thanks to holidays.

3.15.3 Wednesday 29.6

Today I fixed an issue with memory leak. There was an issue that File Streamer was left open for one component monitoring files. This issue is not crucial but may prove to have a bigger impact later on. This I fixed with simple Using clause, which automatically disposes the File Stream. This required some working around and intensive googling.

3.15.4 Thursday 30.6

Finally got the needed people in same room to troubleshoot and fix the issue. After a lot of going through databases and checking AD rights etc. we found the real issue with the service. The service account didn't have sufficient rights and there was some issues with database rights too.

3.15.5 Friday 31.6

Demo day and last day of the sprint. Mostly what we demoed was just a load of bug fixes. Not that much new features to show. After retro we ended the day.

3.15.6 Weekly analysis

This week I investigated the Memory leak issue. Memory management is a new thing to me, since that is something that I do not have to worry about usually. But this time I had to go through lot of google results to find what and how. But initially the issue was a file stream was opened and then it was not properly disposed. This proved a big issue when there was lot of processing to do. It causes the stack overflow error which in turn crashes the whole process. This is a big problem. However, thankfully it was easy enough to analyse and fix.

This memory leak was caused by a file stream not being closed correctly. I simply enclosed the whole problem area with using block.
As it invokes the disposal of the stream after all the necessary operations are done. (Velkov 2012)

4 Analysis

4.1 Discussion

During the time of writing this thesis, I think I have progressed as a professional developer. I have learned so much. Many of the things I have learned are through natural progression. For instance, I have learned how entity framework works, I have learned how to access entities in database with the LINQ clauses, and I have created complicated queries with multiple subqueries. The most important thing I have learned is trouble shooting issues and finding the bug. I know how to make complex SQL scripts to get the data what I need to see. Hunting for the cause of the bug is a talent. It is talent that one learns while doing it. I have also learned many other processes besides development. Setting up environments, system administrator skills, setting up an email sending from SMTP server, all are skills I have picked up during my work.

I have also realised the importance of documentation and well-defined and developed processes around application development. If the processes of project management are not mature, enough then it will also affect the Dev team. Many external factors play part in work of regular programmer. I have learned how to go through the stack from front end to back end. This was a big issue initially to me. Since I did not have a picture of the whole application. Now I am able to grasp much larger concepts

By the time, I am finishing this thesis I feel like I have progressed from a junior developer into a developer. I have developed my workflow during these weeks.
I usually start by recreating the issue. After that, I will try to understand what happens and why it happens. Then I will draw a plan on what should happen and how. After this, I try to test as much as I can.

4.2 Conclusion

This has been a very good learning opportunity for me

And writing this thesis has shown me how much I have actually progressed as a professional. Straight out of school to real work life was a shock initially and showed me how little I knew. During these weeks I have seen many situations and solved plethora of problems. I have learned to react to different real life scenarios that might happen I nsoftware development. There are unknown factors that cannot be accounted for. This means that I as a developer have to have thinking skills and problem solving skills. It is not just the proficiency in programming but also the aforementioned skills that I have to possess to be a great developer. This thesis helps me to see that there are so many aspects to my job that constant learning is needed in order to succeed and to become a better professional. I have benefitted a lot from writing down my daily duties and it has helped me to progress. with this thesis project I was able to see where I need to do some improving and what are my problem areas. I know what skills are the most important to develop further.

5 References

MSDN 2016a. Microsoft developer Network. Query Syntax and Method Syntax in LINQ (C#). URL: https://msdn.microsoft.com/en-us/library/bb397947.aspx/. Read 21.10.2016.

MSDN 2016b. Microsoft Developer Network. ASP.NET MVC Overview. URL: https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx. Accessed:21.10.2016

MSDN 2016c Microsoft Developer Network. Chapter 5 – Defining Your Branching and Merging Strategy. URL: https://msdn.microsoft.com/en-us/library/bb668955.aspx. Accessed:21.10.2016

MSDN 2015. Microsoft Developer Network. Working with DbContext. URL: https://msdn.microsoft.com/en-us/data/jj729737(v=vs.113).aspx. Accessed:21.10.2016

MSDN 2015b. Microsoft Developer Network. Loading Related Entities. URL: https://msdn.microsoft.com/en-us/data/jj574232.aspx. Accessed:21.10.2016

Grier, S. 2016. IT managers inbox. URL: http://itmanagersinbox.com/1556/how-to-write-it-technical-documentation/. Accessed:21.10.2016

Crouch, S. 2016. Software Sustainability Institute URL: https://www.software.ac.uk/resources/guides/developing-maintainable-software. Accessed:21.10.2016

Entity Framework Tutorial 2016. Entity Framework Tutorial. What Is Code First? URL: http://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx

Guthri, S. 2010. Code-First Development with Entity Framework 4URL: https://weblogs.asp.net/scottgu/code-first-development-with-entity-framework-4

Velkov, L. 2012. Common Scenarios That Result In Memory Leaks And How To Avoid Them URL: http://mentormate.com/blog/common-scenarios-that-result-in-memory-leaks-and-how-to-avoid-them/.