horizontal line

Current State Process Analysis Report

The purpose of this document is to summarize the findings of the first two stages of the Process Improvement Methodology – Project Scope and Gather Current State.

Contents

[Week 7 2](#_Toc43852117)

[Week 8 3](#_Toc43852118)

[Week 9 4](#_Toc43852119)

[Week 10 5](#_Toc43852120)

[Week 11 6](#_Toc43852121)

[Week 12 7](#_Toc43852122)

[Week 13 8](#_Toc43852123)

[Week 14 9](#_Toc43852124)

[Week 15 10](#_Toc43852125)

[Week 16 11](#_Toc43852126)

[Week 17 12](#_Toc43852127)

Personal Reflections………………………………………………………………………14/13

Applying Iterative……………………………………………………………………………15

Waterfall vs Agile…………………………………………………………………………….16

# Week 7

Meeting Date | time *30.03.2020* | *11:30* | Meeting location *Fontys R1*

Today’s decisions and subjects were regarding the new model we will be working on (agile) instead of the old one (waterfall). After receiving feedback from the mentor and from the group that tested our product, we began working on bug fixing and discussing how the website will function and added old versions of the database/graphs.

Completed Tasks

1. Everyone started working on fixing bug fixes on their part of the software
2. Radoslav updated the process report

# Week 8

Meeting Date | time 0*9.04.2020* | *11:30* | Meeting location *Fontys R1*

This week we concluded that we should update the master feature list. We also started working on peer reviews, updating all documents related to the database and design, and made sure that we need to test our app for the next iteration. We also discussed how to

refactoring the code.

Completed Tasks

1. Everyone started working on refactoring the code.
2. Radoslav updated the process report
3. Lukas added a DB helper class
4. Everyone discussed internally what we should work on for the remainder of the week

# Week 9

Meeting Date | time *16.04.2020* | *11:30* | Meeting location *Fontys R1*

This week we concluded that we should create a list of restock request, adding a date for when the stock request was made, fire date for employees and not instantly deleting employees from the database, instead we should “soft delete” them. We also discussed what the statistics page should contain and what the website might look like.

Completed Tasks

1. Radoslav made the statistics page
2. Radoslav worked on the login system of the website
3. Radoslav updated the process report
4. Kristian worked on restock request
5. Kristian worked on adding dates to stock requests
6. Ivan worked on adding fire dates for employees
7. Ivan worked on soft delete
8. Lukas worked on wireframes
9. Lukas worked on designing website structure

# Week 10

Meeting Date | time 23*.04.2020* | *11:30* | Meeting location *Fontys R1*

This week we concluded that we should add a password encryption for employees in the application and website. We also discussed adding BSN to employees, add multi line for address column, add gender to database, add shifts to db, an updated design of the database diagrams and add attendance. We also concluded that we would need a “forgot password” feature on the website.

Completed Tasks

1. Radoslav updated a bug with departments list box
2. Radoslav updated the process report
3. Everyone worked on updating the database
4. Kristian worked on forgot password feature
5. Ivan worked on adding features to employees
6. Lukas worked on designing the login page
7. Lukas worked on employee information page

# Week 11

Meeting Date | time 7*.04.2020* | *11:30* | Meeting location *Online*

In this week we discussed the upcoming client meeting and the goals that need to be reached in order to meet the criteria. We spoke about continuing with debugging our code and creating a presentation for the meeting. We spoke about updating the UML Class Diagram.

Completed Tasks

1. Radoslav and Kristian created the Presentation
2. Lukas Created the Generation of shifts in the website
3. Ivan debugged the User System

# Week 12

Meeting Date | time 14*.05.2020* | 10*:00* | Meeting location *Online*

We held the meeting with the client. We presented our application and we spoke with the client about some new features we might have in the future. She concluded that a Cashier App can be completed, also an algorithm which automates the assignment of the shifts by checking the preferred employee shift. We were given the task to add more information regarding the employee and shift system.

Completed Tasks

1. Everyone attended the meeting and gave their reflection at the end, while being active during the discussion.

# Week 13

Meeting Date | time 21*.05.2020* | 11*:30* | Meeting location *Online*

During the meeting we spoke about the new possible addition to our software solution – the shift algorithm. We placed it as high priority and as something that needed to be placed above the other possible extensions. We also spoke about creating an Activity Diagram and the Cashier application, which was placed on low priority for now.

Completed Tasks

1. Kristian updated the Iteration History Document
2. Ivan refactored the Employee System

# Week 14

Meeting Date | time 28*.05.2020* | 10*:30* | Meeting location *Online*

In this meeting our mentor gave us a task to split into two pairs and review each other’s code. Lukas and Radoslav were the first team and Kristian and Ivan the second. During the meeting we concluded that we will not have time for all possible additions mentioned in the project backlog, so we decided to not implement the Cashier application and the check-in system, that we had planned. We put code improvement and refactoring as number one priority.

We also agreed that we needed to improve the UML Class Diagram.

Completed Tasks

1. Radoslav refactored the Department system
2. Kristian refactored the Product system
3. Kristian added Restock Request filter for all Stocks

# Week 15

Meeting Date | time 04*.06.2020* | 10*:30* | Meeting location *Online*

During the meeting we discussed the importance of the Shift algorithm and discussed how it was going to be implemented. Our mentor reviewed the document we sent as pairs and concluded that the current code needs to be refactored as some of its systems were not following the SOLID methodology. Our mentor also informed us about the final document that must be created before the last meeting.

Completed Tasks

1. Radoslav created the Activity Diagram.
2. Radoslav created the front end of one of the two pages for the shift algorithm.
3. Ivan created the front end of one of the two pages for the shift algorithm.
4. Kristian added the Updated By feature for the Shift system.
5. Kristian refactored the code by sorting the classes into folders and splitting the large data access class intro smaller parts.

# Week 16

Meeting Date | time 11*.06.2020* | 10*:30* | Meeting location *Online*

In this meeting we discussed the part about improving our existing code and refactoring it in order to apply the design principles that we learnt during this semester.

We went through the iteration additions that we must have for the end of the semester those being: The shift algorithm and the improved UML Class Diagram.

Lastly some general questions were asked regarding the grading system during the exam week.

Completed Tasks

1. Kristian added new possible additions to the Iteration History document
2. Kristian updated the Process report and the URS document

# Week 17

Meeting Date | time 18*.06.2020* | 10*:30* | Meeting location *Online*

During this meeting we made a demo of our current application and after that we discussed some additional features that could be added before the last meeting with the client. Those features being: The Cashiers Application and some more Statistics for the managers.

While we were presenting the demo our mentor gave us some tips about how we might improve it and what else we might want to add as additional features. We went over the needed documentation for the final presentation again and we asked some general questions about it as well.

Completed Tasks

1. Lukas created the Shift Algorithm.
2. Lukas Added a new feature: Fired employees cannot log in to the website.
3. Ivan refactored some of his code in the application.
4. Kristian Created the Cashiers Application
5. Kristian Created the Manager Income View Form

Member Remarks | *Radoslav Karaganchev | Ivan Marinchev| Lukas Rimavičius | Kristian Lachev*

Kristian Lachev

This project really managed to make me improve as a software developer. I gained a lot of knowledge in almost all aspects of programming, from planning to developing and I learnt how to better manage my time. From my experience with both the Waterfall and the AGILE methodologies I could definitely say that I enjoyed the latter of the two. The reason for that lies in the fact that you communicate more with the client, so we as software developers don’t have to guess as to what exactly the client has envisioned.

During the second part of the semester while working with the AGILE methodology, I found that there was far less ambiguity with than with Waterfall, because of the frequent meetings with the mentor and the client. We had set a clear goal from the start and I found it easier to stay on the course towards it and not guess what features we are going to have, because of the lack of communication with the client.

Regarding the project as a whole, I liked the teamwork aspect of it as it prepares us for the real world, where we will most likely be in a team and we will have to work with other people towards one goal. There will be some things that we might not like about that but at the same time the lessons that we learn here will be valuable in the future.

All in all I liked this course, and I would definitely say it has helped me improve in a lot of aspects. Out of the two mentioned methodologies I very much prefer AGILE as, like the name implies, it gives us more flexibility as a team and more communication with the client in order to create a software that the client would be pleased with. The thing that I liked the most about the course was the team aspect of it and I would definitely say that it was a valuable experience.

Radoslav Karaganchev

In this semester I have learned how to work in an agile setting which has benefits I really enjoyed like work-life balance, meaning that I chose when and how much to work on. Also the communication between group members, client and tutor were significantly better compared to waterfall. Overall this project was a good learning experience and has demonstrated what challenges I will face in a real working environment. The project also helped me with time management, soft skills and overall c# knowledge.

The Iterative Process

At the beginning of the second part of the semester we were getting used to the iterative process in the AGILE methodology, but we soon realized that it is by far the better choice when it come to picking between this and Waterfall. When it come to the distribution of work the iterative process that we had helped us immensely.

The strengths of this process come from the fact that we have far more communication with the client and thus when we deliver the full application in the end, everyone is happy. The communication that we had with the client during the semester really made the difference as to which new features we needed to

add.

Another major strength of this process is that it makes us more adaptable when coding, meaning that even If we have many features we can always choose which one we want to actually include in the application and which one we discard, or save for later. This adaptability gives us the freedom to plan out each feature we are going to have in the future.

One of the cons of this iterative process might be the strain we put on the client, as we do have a meeting every two weeks and we keep close contact. All this communication means that the software engineers are much busier and too much workload might hinder the quality of the code that is being written. The constant meetings with the client do ensure an application that will satisfy both client and programmer, but it does become cumbersome when enough time passes.

In the end the iterative process has more strengths than weaknesses and even if there is a lot more strain on the client and the programmers, it does ensure that the end application satisfies both parties. Mental strain can always be avoided with more social interactions or more fun activities.

Differences between Agile and Waterfall

There are differences when you compare both methodologies but probably the biggest one is the communication you have with the client.

While we were working with the Waterfall method we noticed that most of the features that we had implemented did differentiate the client’s original view. As much as we wanted for our software solution to satisfy the client in the end we received mixed responses, as some of the features did differentiate. When it came to coding, we had more of a messy approach. A lot of the code that was made was rushed and there was a slight hint of disorganization as we did not follow any strict plan. We just had an application that needed to be made. In the end we did deliver but it was far from perfect and neither the client nor we were

satisfied.

While working with Agile however, we did notice a change when it came to our approach. We came in more prepared and with the better client/programmer communication, we really did achieve more in the long run. We had a clear view of the project and what we needed to complete as a team, and the work distribution was more even. When we completed the first two iterations we already made more progress than when we finished the project with the Waterfall methodology.

We had more team chemistry and the fact that we increased out meeting frequency really helped with communication.

Another large difference between Agile and Waterfall was the flexibility. Agile, in itself, is very flexible, and when it comes to comparing the two, there is no contest. As for Waterfall, when the client defines the requirements at the beginning and decides to change them later it is very difficult to implement those changes. Often when we talk about clients, the requirements that they first set will most likely change and the methodology that the programmers follow should be flexible so when that change in requirements comes the team can adapt quickly and smoothly.

Overall, with Agile the client and the software developers have better communication and in the long run that makes a big difference. Agile is far more flexible than Waterfall and that flexibility and adaptability gives it an edge when it comes to the changing requirements of the client.