Process Report

STUDY : PROTO

Members: Kwangjin Lee, Evaldas Drasutis, Emilian Alexandru, Florin Deleanu
Tutor: Stan S. van Hartingsveldt

Date: 07/10/2020

Table of contents

1. Work Division

Documentation:

Project plan

Main: Kwangjin, Evaldas, Emilian, Florin

URS

Main: Kwangjin, Evaldas, Emilian, Florin

Minutes

Main: Kwangjin, Emilian

Agendas

Main: Florin

Test plan

Main: Kwangjin, Evaldas, Emilian, Florin

Process report

Main: Kwangjin, Evaldas, Emilian, Florin

C# Application

GUI

Main: Kwangjin, Florin

Employee management system

Main: Kwangjin, Florin

Employee scheduling system

Main: Kwangjin, Florin

Employee attendance checking

Main: Kwangjin, Florin

Employee statistics

Main: Kwangjin, Florin

Product management system

Main: Emilian
Secondary: Kwangjin, Florin

Ticket system

Main: Evaldas
Secondary: Kwangjin

2. Personal reflection

//asked from stan in email: What went well? What didn't go well? Why didn't it go well? What would you do differently next time?

Kwangjin Lee

1. What went well?
   - Teammates are all passionate, no one is late for discussion, we give proper feedback to each other.
   - We could understand other teammate's code, which is important for a group working as programmers.

2. What didn't go well? & Why didn't it go well?
   - It was hard to understand what to do at first.
   - Working with databases was the most difficult part, we were not able to find any error in visual studio, so we had to find it one by one
   - Due to the fact that we do not have the same skills in programming, someone should do more work.
   -
3. What would you do differently next time?

For the next time I would like to be more careful for time distribution to have enough tests and clear codes for the program. It would be nice if I learned a more efficient way of programming. So that I can catch every small mistake.

Evaldas Drasutis

4. What went well?
    - The group communicated well on the documentation, as well as managed to recognise the problems we had and find their respective solutions quickly.
    - As a team we programmed everything we needed, with the help of fellow team members, we managed to finish it to function properly.
    - In my experience in this project, I got to recognize different types of solutions that I don't frequently use that could be used as a better way of programming.
    - As a team we kept organizing everything we needed to do and stayed in communication with each other on the basis of this project, so I'd say our communication went well.
5. What didn't go well? & Why didn't it go well?
    - The team had more efficient programming approaches to the solution, whereas my approach was not as structured to be maintainable.
    - Due to the lack of our own period to learn and adapt our own solutions, it took too long so other teammates had to interfere and assist our code in the last few weeks of development.
6. What would you do differently next time?

Next time I'd like to be more flexible with my code and adapt quicker to newer types of implementation. I'd like to communicate and work together more as a team and learn from each other, so that we can work fully on the same level as others. That way we can depend on each other. I'd like to not slow down the process of the group project with my shortcomings.

Emilian Alexandru

1.What went well?

- The team applied the waterfall methodology with little deficiencies.
- The software solution went quite well, with a few minor bugs as the team managed to stay on track and nearly all the musts and shoulds functional requirements were implemented.
- An active attitude, organising group and tutor meetings beforehand, was encountered with nearly all members, managing to communicate with the client and the tutor and apply the received feedback in due time. Due to my team members, I was able to develop a better understanding of how to successfully apply and design the functional requirements for the project but also how a team should function.

2. What didn't go well? & Why didn't it go well?

The team went through a somewhat stressful situation during the last moments of the development, as not all members successfully finished their assigned requirements in time, resulting in an uneven work division, a longer implementation of the functional requirement and sometimes unfinished implementations.

3.What would you do differently next time?

I will try to have a more active communication with the tutor but also with the team members and work more on finishing the assigned tasks and applying feedback in time, with as little deficiencies and as optimally implemented as possible.

Florin Deleanu

1.What went well?

- The communication between members was very good, apart from little misunderstanding which were solved by more detailed explanations
- With the feedback from the meeting with the teacher we were able to get a more clear view of what we did and did not do well during the app building process and thus update it to keep on track
- Seeing the different coding styles from my teammates I was able to understand new approaches to solving the same problem

2.What didn't go well? Why didn't it go well?

- At the beginning of the app we didn't really know how we would include the database so we had to revise the design of our code in order to be able to integrate the database functionality.
- Because of the different coding styles of our members it took us longer than expected to merge the functionality of the code together
- Toward the deadline, not all the functionality was working as intended, meaning some of us had to work extra on the others' code

3.What would you do differently next time?

I would make sure to start the implementation earlier so as to have enough time to deal with unexpected situations without being constrained by the time. I would also make sure to set multiple smaller deadlines for features of the project in order to be sure that everything is working on time.

3.Applying waterfall reflection

Strengths	Weakness
Good for defining the goal and deliverables	Delays the development and testing of the

	application by not being able to travel back to previous stages
Gives a clear idea of the application to the client from the beginning	The results are not visible until the end of the project life cycle
Phases do not overlap	Can ruin the project by having defective initial requirements