# Introduction

Hello and welcome to the presentation of our Second Semester Project. We are group 2 of class Y, also called the No Brain No Pain group, composed of Dragos, Chiril, Dan and Matas. The project our client came up with is a Music app, or how the development team called it CHADMusic. The client complained about his inability to listen to music on a add-free app, also concluded that a lot of those apps do not meet the expectations when it comes to functionalities like display of lyrics and highlight them, sound design, shuffle playlist, putting the app to sleep when not used, download music/listen offline means of using playlist functionalities in general and if some of those features are around, they come at a cost of a monthly membership. Beforehand our customer tried Spotify and Tidal, only Spotify containing adds and some features like listening in the background, and more being locked. Tidal won’t even let you listen to music without a subscription. The conclusion is that everyone should be able to listen to music add-free and benefit from all the features that the app has.

## Demonstration

In order to satisfy the needs of our customer the team ended up with a nice looking music app, that is possible to play music, show lyrics and provide basic features for the current release. We can show you some important points of the program:

* The app can connect to a server
* The app can receive mp3 files from the server and play them
* The app can fetch the lyrics of the currently playing song
* The app can repeat a song
* The app can register and login users

## Implementation

To make his wish come true, the team was required to make sure that they make exactly what the customer wants, and so the team decided that they will use Agile in the implementation of the app. Unified-Process as well as SCRUM are considered to be agile, and both containing a cyclic incremental way of developing, SCRUM containing sprints as the cyclic iteration and UP containing phases in which certain development is done.

Agile is made on a set of beliefs that deduct how the team should manage the workflow:

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

The SCRUM methodology contains 3 different roles:

* Product Owner who interacts directly with the customer, and makes sure that the product ends up the way that the customer wants
* Scrum Master who must make sure that everyone of the team works efficient, follows SCRUM and agile, and is motivated to finish the product
* Developers which are the ones that implement the product, the ones that make the product according to the Product Owner’s requirements.

The UP (Unified Process) contains 4 phases in which the team goes through each step:

1. Inception - the purpose is to establish a shared understanding of the opportunity and the business case for producing a solution before committing to the solution's development
2. Elaboration - during this phase the project team is expected to capture a majority of system's requirements
3. Construction - the design of the system is finalized and refined, and the system is built using the basis created during elaboration phase
4. Transition - the project team focuses on correcting defects and modifying the system to correct previously unidentified problems, the milestone of this phase is called Product Release

With the Manifesto for Agile Software Development in mind, the team started on the inception phase with the project description, from which the team understood the idea of the project. Afterwards the product owner extracted the requirements and made sure that he understood correctly from the customer their importance.

At the end of this phase the project had a starting point, everyone knew where to start from, what to do and how the project should be like, after which the elaboration phase started.

This phase is the phase when the most time was spent in this project. It is the phase where the kernel, the base of the project must be done, and since none of us knew how to play sounds in java, get lyrics from the internet on java, transfer music files between two computers and design a good database for the server to operate from, it required a lot of hours of research… and when we say a lot, we mean it. The class diagram was drawn at the end of this phase very complicated and big, but we managed to get it smaller in the next phases.

The elaboration started with creating the GUI, we created the base on which to start on, and afterwards we added features step by step. Of course, the main point of a music app is to make the app play music, so that was our first requirement. Afterwards we managed to attach the lyrics API to our project after which the hard part came. The task sounded simple, but it was extremely complicated. The task was to make the server send a mp3 file to the client based on what the client chose in the list. Nothing to complicated about letting the server know the selection of the client side, but very complicated to send the file. Our approach was to convert the file into a byte array, and then send the whole byte array to the client where again it will be converted back to a mp3 file (Sounds simple right? :P well it wasn’t :DD).

After the elaboration phase the class diagram was still big and complicated, but on the next phase which is Construction we removed unnecessary classes which the client should not know about and which the server didn’t need.

We started the construction of features after two hard sprints of elaboration. Sprint 4 was focused mainly on features and looks, at this point everything was coming together and the team’s motivation was sky high.

As the deadline was approaching, we were a bit disappointed that the construction phase lasted only one sprint, but we had to make sure that we go to Transition phase as well.

The 5th and the last sprint we had was to make the login function of the program and to finish documenting. We left some time to make sure we optimize unnecessary classes, test what had to be tested and make sure one last time before zipping that everything looks and works as intended.

## Testing phase

After construction phase, in the transition phase we tested everything. The testing was done using ZOMB+E but it was not so complicated for us since a lot of the things we used didn’t require thoroughly testing. Every class from the model was tested using black boxing. We cared more about the output, if we get the correct output rather than testing every part of the method. During the viewModel testing it was a bit more complicated, since for it to work we needed to simulate the GUI to test it without using the GUI.

## Results & Discussion

At the end but not least, we believe as a team that we did a good job making sure we deliver a good quality product. Despite its latest release, we are satisfied with the outcome of the project. Even though the requirements list has still a lot more user stories undone, we think that we accomplished a big part of the important requirements which in theory makes it prone to easy updates and improvements.

Pointing out the good parts of it, it was an amazing feeling to see that for example a team member had the server running, and another one(s) had the client open and listening to music. The lyrics are working flawlessly, the server<->client communication works without any errors… and that is with the client not being connected continuously to the server, the way the server and the client communicate is very cool in our opinion. The client connects to the server only when it needs to and disconnects as fast as it receives what it requested.

## Learning outcome

We are aware that this semester project was focused on teaching us how to use code in Agile, how UP works and familiarize ourselves with SCRUM and sprints.

It made us understand now how people work in a real-world scenario which is way better than the way we did it the first semester.

Of course, at the beginning we didn’t do it right (who would’ve known :P) so the first sprint wasn’t following Agile, but after more supervisor meetings we understood the way, everything should work (fortunately) and naturally starting a bit to stress out because we were not on the right path. We believe that if we didn’t mess up now, we could’ve done it in the future, maybe in an internship or so on.

We did also learn the hard way that the thing you learn in the classes is just the tip of the iceberg and there is always a lot more to do and a lot more research that we were comfortable to do at the beginning.

We are grateful to each other for the constant support we gave to every one of us. As a group we work very well, output a lot of work in a short period of time, we make progress every time, we meet every day and talk every issue out. Without constant supervising we wouldn’t be able to make this project in time. Thank you for all the support you provided us with and thank you for your patience and we hope you enjoyed.