Group Organization

Introduction

AREA is an ambitious project. Its objective is to create **links of Action-Reaction** between **services** (e.g.: Youtube or Spotify). Our service will detect an **Action** chosen by the user (e.g.: liking a YouTube video), it will then take **data** from this action (e.g.: the title, the genre or the author of the video) and use it while triggering a **Reaction** (e.g.: adding the song linked to the video title to a Spotify Playlist).

To accomplish all of that, we need three extensive parts. Each one will have its own importance in the project. The first one will be an **application server**. It's the part that will handle all the business logic in the background (e.g.: checking for action, triggering reaction). But accessing this without an interface is quite difficult for an average user, that's why we'll have two clients to help customers with the utilization of our service. First, a **web client** that will be used to access our platform from any device connected to the Internet. But also, a **mobile application** to give our mobile users the best experience possible.

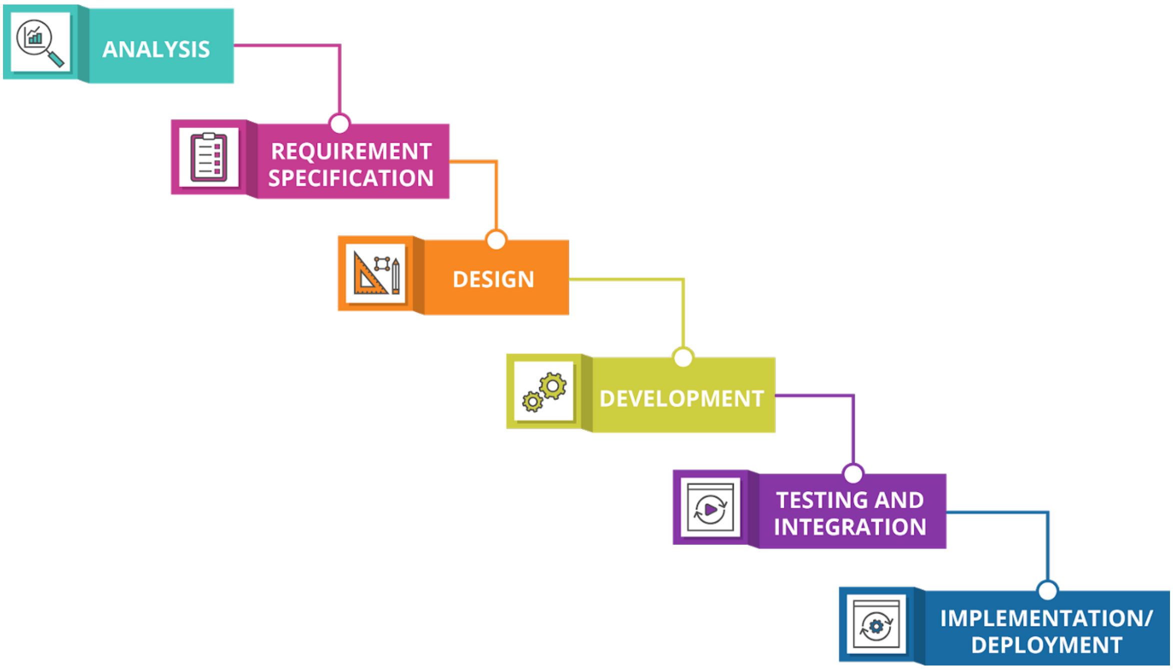
To successfully achieve this project, the group needs crucial organization and communication in order to lose the least time and make the least mistakes. That's where this document comes into play. From our project management methodology, to the urgency plan in case a member leaves the work group, you'll find everything you need to know about our group organization.

Our project management methodology

The key to a successful project is to have a great project management strategy, and it's easy to be overwhelmed by the number of methods present nowadays. Some of them don't take into consideration changes and prefer to follow the plan, others do the strict opposite and finally some of them take the best of both worlds and follow a plan while accepting that change could happen at any given moment.

We're going to talk about three famous methods that came into our mind while trying to find our happiness. The Waterfall model, the V model and the Agile model.

Let's start with the Waterfall, it takes its name from the way the project flows in a one way through all phases:



The main problem with this method is that it's not really made and ready for changes in main concepts. If the requirement specification changes, nothing is ready and everything is going to crumble. Something we don't want for our organization system. Moreover, this mode relies heavily on planification and since we're not the most experienced developers we might make mistakes and need to modify things.

The second concept is the V model, it's an upgrade of the waterfall system. It improves some inefficiency that waterfall had, but doesn't really fix the planification and changes' problem. So we decided to look at the third method.

Agile seemed like an ideal solution for us, it's something that led to success a preceding long term project for many members of the group. We decided to modify some of the things we had in the past and it led us to our actual organization. The actual organization works as the following:

- The project is divided into multiple sprints.
- The first sprint is dedicated to the preparation of the whole project and the second sprint is dedicated to the construction of the MVP. The 4 other sprints are worth 8 days of time and will have different themes that will suffice to complete the project.
- Every beginning of sprint will have a reunion where the already divided task are going be chosen for this sprint and attributed to the respective person
- Then a daily meeting will take place where we attribute daily task to each person, discuss what has been done and if anyone needs help
- At the end of the sprint we all discuss what could have been better this sprint and what we could change to upgrade our quality of work.

Roles in the group

Alexandre
Frantz

Maxime
Hoffbeck

Mathieu
Brujan

Ludovic
Sutter

Timothé
Coniel

I'm happy to present our team! Each one of us is going to fulfill a precise role. We needed to have at least one group leader, three lead developers (one for each pole) and finally someone that can support the three branches of the project when they need to. Let's look at the talent of each member and the things they want to learn related to the project.

	Alexandre	Maxime	Mathieu	Ludovic	Timothé
Quality	Fast Learner	Front-end	Fast Learner	Back-end	Versatile
Ambition	Mobile	Front-end	Leader	Back-end	Everything

We can see that some people already have experience in fields important for the AREA:. Ludovic with great back-end knowledge and Maxime with great front-end knowledge. Making them obvious lead developers for each of their experiences. That leaves us with three people, Alexandre who has a desire to learn mobile, Timothé who would like to learn the three domains important for the area and Mathieu who would like to learn mobile and also be leader of the group. With this the rest of the repartition does itself easily. Alexandre as lead mobile developer, Timothé will support every pole when they need them and Mathieu is going to lead the group.

	Alexandre	Maxime	Mathieu	Ludovic	Timothé
Role	Lead Mobile	Lead Front	Leader	Lead Back	Technical Support

Plans B

Now that we have our roles we need to list our solution if any of us leaves the group or is unable to work.

If Maxime or Ludovic can't work anymore, Timothé, who learnt both the front-end and back-end framework will replace the missing member. In the case where Alexandre is the one who cannot work, Mathieu, who is the assistant mobile developer will replace him. If he can't work, a new group leader will have to be elected between the 4 members that are still here. And finally if Timothé is unable to work every member would have to work way more to finish the project as no one could replace his role.

We also discussed all together and decided on the limits when someone is not considered in our group anymore.

Tools that will help the organization and communication

In order to communicate between each other at all times we decided to set-up a Discord server since it's a medium we're used to. Both written messages and vocal discussion are possible, making meetings online and offline as easy as it gets. To keep a good organization and a good overview of the whole project, we decided to set-up Trello, a tool used for project management to keep tasks on a board under the form of cards. One reason we chose Trello over any other tool that can do the same thing is because Ludovic in our group created a Discord bot to inform us in real time of the movements of a card so we can know when the state of the project changed. Another reason is that we all are beginners in project management tools, so starting with one of the most basic is better than getting overwhelmed by a tool that has too many functionalities. Sometimes less is more.

Github management

Finally, organization in a developer group involves the safety and a good maintenance of code. That's why we chose Github as a service to keep our code. And we gave it a few rules for it to stay in the best state possible.

We'll have a "Main" branch that will contain the last stable version of our program. Any person getting our project in the "Main" branch at any time will get the best stable version of the AREA. that came yet.

Then a branch "Develop", it will be the only branch that will be allowed to add code to our main when a sufficient number of features are ready to be pushed into production.

Each feature in development will be in a branch named after the feature. To merge with "Develop", the developer must open a new Pull Request that at least two people will validate.

Every commit will have the following norm: Fix/Feat/Refactor: commit comment explicit enough to know what it is about.

Conclusion

In conclusion, our project management methodology is a derivative of agile adapted for our project and our group. This choice is justified to prevent change and mistakes due to our inexperience. Each person has a role representing his actual capacity and ambition. We planned every back-up in case a problem arises. We chose Discord and Trello as tools for communication and organization. Finally we have a really strict Github policy making commits in mains impossible leaving us with a clean repository.