Retrospective - Iteration 5

https://docs.google.com/spreads **Multimedia Services** **Team: The First Order**

User Story	Task #	Assigned to	Done	Estimated effor	Actual effort	Notes	Pull request			
Sprint planning	Create sprint plan	Ruben	yes	2	2		documentation	Documentation		
As a team, we want to have a good	Create sprint retro	Ruben	yes	2	2		documentation			
overview of tasks that need to be done										
and we want to reflect on work from										
previous sprints to improve.										
Product architecture	Keep architecture doc	Hung	yes	3	3		documentation			
As a user, I want to be able to see the	up-to-date	Martin		2	0	Martin had to leave early when this document was created.				
big picture of the software, so I know										
how to implement new functionality.										
Pull requests	Handle pull requests	Ruben	yes	4	4			Code quality		
As a developer, I want my code to work		Hung		4	4					
correctly and I want others to know about		Chris		4	2					
and comment on my code.		Christian		4	1					
		Martin		4	2					
Testing	Make tests for front-end	Christian	yes	5	0	Testing is up to 72% now, most tests were written for the back-end.				
As a developer, I want my code to be		Martin		5	9		https://github.com/rubenwiersma/thefirstorder/pull/62			
reliable and secure. Testing has to be up		Hung		0	5	Testing the new entity structure and service classes.	https://github.com/rubenwiersma/thefirstorder/pull/55			
at 75%.										
Changing cues	Create changing cues option	Christian	yes/no	4	6	Cues can be added, but changes are not yet maintained in the database.	https://github.co	Features		
As a scheduler, I can easily change the		Ruben		3	2					
position of a cue so I can change my mind.						*This feature is not yet merged, because it needs more code fixing.*				
Improve screen size										
As a scheduler, I can view the application on	Make design full-screen	Ruben	yes	3	2	Implementing this took a little less time than expected, everything	https://github.com/rubenwiersma/thefirstorder/pull/48			
a large screen. This enables me to view		Chris		3	0	scaled nicely.				
large pieces of the score at once and	Make design more like an app	Ruben	yes	3	3		https://github.com/rubenwiersma/thefirstorder/pull/65			
to create large maps.										
Export	Create XML export	Martin	yes	5	5		https://github.com/rubenwiersma/thefirstorder/pull/53			
As a scheduler, I can export the created script		Ruben		4	5	Ruben took on implementing the front-end work for linking the XML button.				
and hand it over to e.g. a camera operator.	Create XML download button	Hung	yes	2	0	Was done by Ruben.				
Project based working		Martin		1	0					
As a scheduler, I can load a score into the	Create option to link maps	Chris	yes/no	4	4	Everything but the map is now linked to a project.	https://github.com/rubenwiersma/thefirstorder/pull/49			
project which I can see in the application.	to project	Christian		3	0					
	Create option to load scores	Chris	no	4	0	Loading scores was lower in our priority list.				
		Christian		2	0					
	Create options to load and	Chris	yes	0			https://github.com/rubenwiersma/thefirstorder/pull/64			
	create projects						https://github.com/rubenwiersma/thefirstorder/pull/60			
User friendliness	Improve map user friendliness	Christian	no	3	3					
As a user, I want to be able to use the map		Hung		0	3	Hung spent time on fixing unstructured code in order to improve the map.				
more easily.										
Backend APIs	Create APIs for accessing data	Hung	A	4	3			Structure		
As a programmer, I want to have models and		Chris		3	3					
apis ready for future development.	Create APIs for editing data	Hung		4	2					
		Chris		3	0					
	Fix relations within database	Martin	B	4	1	The database schema needed to be refactored for each change, this took	https://github.com/rubenwiersma/thefirstorder/pull/54			
		Hung		4	7	more time, since all forms and front-end pages were manually updated.	https://github.com/rubenwiersma/thefirstorder/pull/50			

Problem 1

Description: A team member was unable to work during meetings because his laptop was used by someone else or he had to work somewhere else.

Reaction: We addressed this as a team and gave him some slack to see what the result would be of the work at home. It turned out that this code was not as good as we'd like it to be, so we decided to

discuss this with our TA. The code will not be merged untill it is fixed, even if that means we can't merge the functionality for this week.

Problem 2

Description: In the retrospective, we can see that most tasks that were split between two members (we tried to split each task) were done by one person.

Reaction: The one person that started the task usually just finished the code, but the other person did spend some time helping out or working together. They did spent less time on the task than the other, but usually did work together in talking about and discussing ways to approach the task.

Problem 3

Description: Not all team members are involved in pull requests.

Reaction: Usually, two people in the group check all the pull requests, but some members need to be motivated to look deep into the code. As we were sitting together, we could ask others to do a pull request but we hope this happens on own initiative more next weeks.

Adjustments

1. We will switch around front-end/back-end tasks in order to achieve higher front-end code quality and to get more knowledge of the system across the entire team.

2. Two people assigned to one task will discuss their task before they start and the one responsible will decide on who does what.

3. Less active commenters on pull requests will be motivated by the others to participate in discussions on code.

4. We will not merge code untill we are all satisfied with the quality of the code. Other reasons to merge quickly should not trump code quality.

5. We will be more strict on meeting together so we can help each other achieving a high standard.