Sprint Retrospective, Iteration # 2

1/12/2021 - 8/12/2021

Issue	Task	Task Assigned To	Estimated Effort per Task (in hours)	Actual Effort per Task (in hours)	Done	Notes
#50	*Implement basic backbone of the TA service*	*Maurits & Winstijn*	*5*	*4*	*✓, but not reviewed*	
#29	*Implement retrieving contract of students*	*Winstijn*	*1*	*4*	*✓, but not reviewed*	*Overshot massively due to unexpected difficulties when developing.*
#32	*Expose all still to be approved hours*	*Winstijn*	*2*	*2*	*X*	*Rework needed since there was an oversight in the definition of non accepted hours in our architecture.*
#30	*Approving / Rejecting of worked hours*	*Maurits*	*1*	*3.5*	*✓, but not reviewed*	*Overshot the time massively due to unforeseen JPA errors and initial difficulty mocking*
#33	*Implement signing of contract by student.*	*Maurits*	*2*	*2*	*✓, but not reviewed*	
#25	*Allow students to candidate themselves*	*Mattheo*	*2*	*2.5*	*✓, but not reviewed*	*Took a bit longer than expected because of getting used to spring again.*

#26	*Implement candidating requirements*	*Mattheo*	*0.25*	*0.3*	*✓, but not reviewed*	
#17	*Create the initial structure of the Course microservice*	*Ali*	*2*	*3.5*	*✓*	*We had some changes in the id of the course and how it will be communicated.*
#48	*Create the initial structure of the Hiring microservice*	*Julie, Martin & Mattheo*	*1*	*2.5*	*✓*	*Understanding the code so far (especially the authentication part) took far longer than expected.*
#23, #24	*Create a complete test suite for the authentication service*	*Martin*	*2.5*	*3*	*✓*	*Took a bit longer as I wanted to make it a flexible example that uses practices that can easily be adapted for the other microservices.*
#17	*Test suite (unit and integration tests) for course microservice*	*Ali*	*5*	*3*	*X*	*Unexpected errors were encountered due to Date class*
#37	*Implement withdrawing candidacy*	*Julie*	*3*	*4*	*X*	*Took time to understand how to properly implement stuff on the backend as I didn't do that for OOPP. Getting an idea of how to tackle such a problem.*

Project: TA-contract service
Group: 13B
Scrum Master: Ali

Main Problems Encountered

Problem 1: MRs aren't opened on time

Description:
George is only able to see our work if it has been merged into dev branches. He can see our current work by quickly seeing the merge requests that are open. However currently we only open an MR if the code is actually ready to be merged.

Reaction:
From now on when someone starts to work on an issue they will open a merge request in draft. Double check that the merge request is a request to dev and that it has the required 2 approvals needed.

Problem 2: Code review

Description:
There was misalignment on the amount of time it would take for us to review code. We hadn’t included the time it would take for code review to our time estimates. There is also a great deal of passive viewing when looking at the open merge requests with no assigned reviewer. No assigned reviewer means anybody can review, and should as soon as possible.

Reaction:
We see the positives of including the total amount of time to be spent on an issue (code reviews are also a part of that) so that we can better plan ourselves, from now on we will put the estimated time to review as part of the merge requests tracked time. The team will be less passive when looking at open merge requests. A channel will be created on discord where pings will be called to all available people that a new merge request is available for review. Merge requests should also clearly specify blocking merge requests whenever possible to quickly illustrate what requests should be reviewed in which order.

Problem 3: Timeline

Description:
There was no information available on the deadline of sprints until the 3rd sprint. We had decided on the start and end of a sprint as Thursday after the meeting before learning about the default sprint deadlines. This has caused the work on our feature branches to be hidden from our TA.

Reaction:
In sprint 3 and later, the sprint deadline is set as Wednesday morning. Therefore, all feature branches for the current sprint will be merged to dev before Wednesday morning (including code reviews).

Problem 4: Overshooting time budget

Description:
Several of our issues took way more time than we expected. Mainly because we all needed to get used to spring and the architecture of our project. Some budgets were also just plain not realistic since they did not include testing or some other crucial part of the work. We also forgot to make time estimations for reviewing merge requests.

Reaction:
We will be trying to estimate the time issues will take more accurately by taking more time into account for testing and setting up a new environment. Also we will add time estimates to merge requests to indicate how long reviewing the code will take.

Adjustments for the next Sprint Plan

Motivate any adjustments that will be made for the next Sprint Plan.

Action Points