Sprint Retrospective, Iteration 3

Context Project: Programming Life

Group: PL-2 (Pantzerfaust)

User story	Task	Assignee	Estimated effort (hours)	Actual effort (hours)	Done (Yes/No/ Partly)	Notes
As a user, I must be able to see the whole graph in the upper-most view.	Find interesting parts of the two data-strains we want to visualize and pass them to the visualizer	Cas	10.0	8.0	Partly	Work in progress, not yet something working on an arbitrary graph, should be finished in next sprint
	Find a clever way to make a bubble of non-interesting mutations	Casper	7.0	0	No	First, bubbles based on the phylogenetic tree were made. The next step will be to create 'random' bubbles to be able to show a top level view
As a user, I must be able to zoom-in on the graph, such that a single mutation can be distinguished.	Create a bubble-making algorithm	Casper	10.0	24.0	Partly	The bubble creation is done, however the zooming into these bubbles still needs a little tweaking.
	Create an interface/link for the graph visualizer module to use the semantic zooming	Cas	7.0	2.0	Partly	Not yet implemented
	Create an interface/link for the data-server	Cas	5.0	3.0	Yes	Should be extended in the future
	Add at least 1 layer to allow for semantic zooming	Justin	5.0	0.0	No	Without a backend based on database that is actually feasibly, this task cannot be completed.
As a user, I want to click to get information about nodes and scroll to zoom in	Finish the clickable node visuals (the part which wasn't completed last week)	Faris	1.5	4.0	Yes	The graphical effect took a lot longer than expected. It has to be adjusted a little bit to work with animations, but besides that it works correctly.
	Use the edge length of the .nwk file of the phylogenetic tree in the visualization	Faris	4.0	0.0	No	It took a long time to get feedback and the other tasks took longer than expected and this is a lower priority task, so this was not completed.
As a user, I want to be able to compare two DNA samples	Create a framework which accepts DNA samples and visualy compares them	Faris	8.0	9.0	Yes	
	Filter a subset of DNA samples out of the graph, so that a new graph for the subset is created	Wouter	6.0	6.0	Yes	Testing was mostly finished but requires a few additions.
As a user, I want a reliable user interface with as few bugs as possible	Test the most important parts of the GUI (which will most likely not change much)	Faris	4.0	8.0	Yes	Tested more than expected (76%), because I had to wait for feedback on the tree and Wouters part to split the graphs wasn't ready yet.
As a developer, I want to have a good workflow and integration infrastructure	Find bottlenecks in build process and fix these	Wouter	2.5	0.0	No	Low priority

	Add Maven Site to Github Pages	Wouter	2.0	0.0	No	Low priority
	Add unit test coverage on existing code	Cas	4.0	6.0	Yes	100% coverage for most model classes
	PR and Issue management	Wouter	-	7.5	Yes	Newly added as it required significant time.
As a user, I want to load my data in an instaneous manner	Design persistent database structure to store the 'default' representation of the graph	Justin & Wouter	8.0	Justin - 1.0 Wouter - 1.0	Partly	First steps were made but no final design has been made.
	Design API and queries to provide to the GUI	Wouter	8.0	2.0	Partly	Too little time remained because of problem 2, but high-level calls are ready
	Setup Streaming Framework to support streamed parsing	Justin	7.0	4.0	Partly	Backend we used performed more poorly than expected. Need to look for new solution
	Research best option for server and client to communicate.	Justin	2.0	2.0	Partly	As above task is not finished, we can't determine whether our choice is the best choice.

Main Problems Encountered

Problem 1

Description: Underevaluation of Project Management time

Reaction: Other low priority tasks had to be put aside.

Problem 2

Description: Database design proved more difficult and had unenvisioned problems

Reaction: Taking a step back to revise the architecture ideas and incorporate the found problems in the design

Problem 3

Description: Because of problem 2, it wasn't possible to test the client-server communication we've chosen.

Reaction: Not implementing this just yet, when we can test the performance we can proceed.

Adjustments for the next Sprint

1. Calculate time for management, as it will take time and cause wrong estimations of the maximum workload.
2. Let people do the tasks which their tasks depend on themselves.

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

1. This sprint (and earlier sprints), we did not yet calculate time for tasks that do nit include producing things, such as issue management and code review. These tasks do take considerable time and must be accounted for in the maximum workload.
2. If the dependencies are completed at the last moment, the dependent task has to be done at the last moment, so if someone has to do the tasks which their tasks depend on it will be easier to plan the workload.

Problem 4						
Description:	Because of dependencies between tasks (the subgraph creation had to be completed before the subgraphs could be visualized) and different people working on those tasks, the task with dependencies had to be done at the last moment.					
Reaction:	Let people do tasks which their tasks depend on themselves.					
Problem ...						
Description:						
Reaction:						