

Sprint reflection 2

Context Project: Health Informatics

Group: HI4

User Story	Task	Task Assigned To	Estimated Effort per Task	Actual effort per task	Done (yes/no)	Notes
The user should be able to link different data files together for a single patient into a sequential data structure	- Filter the files based on patient ID (and handle different data formats/ types)	Hans & Sven	5 Hours	5 Hours	yes	
	- Link the data files together (per patient ID)	Hans & Sven	3 Hours	3 Hours	yes	
	- Create sequential data structure for the linked files	Sven	5 Hours	5 Hours	yes	
The user wants the product planning, product vision and emergent architecture to be documented	- Chapter 1 - Introduction	Matthijs	1 Hour	1 Hour	yes	Processed the feedback on initial version EA
	- Chapter 2 - Product	Matthijs	3 Hours	2 Hours	yes	
	- Chapter 3 - Product Backlog	Matthijs	2 Hours	3 Hours	yes	
	- Chapter 4 - Definition of done	Sven	2 Hours	2 Hours	yes	
	- Chapter 5 – Glossary	Everyone	½ hour	½ Hour	yes	Product Vision was improved on Customer Needs + Comparison Existing Products
	- Update Emergent Architecture	Elvan	1 Hour	1 Hour	yes	
	- Improve Product Vision on feedback	Remi	2 Hours	3 Hours	yes	

The user wants to perform transformations on the raw data (the eight c's)	- Think about the operations that should be done to perform the eight c's	Everyone	2 Hours	2 Hours	yes	Shift transformations screen to next week
	- Create the basics of a processing language for the C constructs	Elvan + Remi	5 Hours	7 Hours	yes	
	- Add transformations screen to the GUI (optional, otherwise next sprint)	Remi	2 Hours	-	no (optional)	
The user wants a simple and easy to grasp graphical user interface	- Create different designs of the graphical user interface	Remi	2 Hours	2 Hours	yes	GUI doesn't have to be changed according to Wenxin, only small improvements
	- Suggest them to Wenxin Wang (on Wednesday 6 may at her office)	Everyone	1 Hour	1 Hour	yes	
	- Improve the current GUI based on feedback	Remi	5 hours	-	no	
The user wants to chunk data together, to analyse it for a certain timeframe	- Create a language construct for chunking	Hans	4 Hours	4 Hours	yes	Basic chunking implemented
	- Implement (script) code for the chunking the data	Hans	5 Hours	3 Hours	yes	
The user wants to constrain his data, based on a constraint that is given	- Create a language construct for data constraining	Elvan	4 Hours	4 Hours	yes	Shift (nested) binary operators to next sprint
	- Implement (script) code for constraining	Elvan	5 Hours	3 Hours	yes (unary operators)	

The user wants to add comments (automated or manually) to a certain data block	- Create a language construct for commenting	Matthijs	4 Hours	3 hours	yes	Shift manual commenting to next shift
	- Add automated comments to hospital records for visit details	Matthijs	5 Hours	6 hours	yes	
	- Let the user add comments to generated data block (optional, otherwise next sprint)	Matthijs	3 Hours	-	no	
The user wants to save the output file with a given name and file path	- Add the option to save the output file with a given file name	Elvan	1 Hour	1 Hour	yes	Writer still has to be linked with the GUI
	- Provide a dialog window to save a file	Remi	1 Hour	1 Hour	yes	
The user wants to call our application by a name	- Think of a name for our application	Everyone	1 Hour	1 Hour	yes	AnalyCs
	- Create a logo for the application	Everyone	2 Hours	-	no	

Main Problems Encountered

Problem 1

Description: Underestimated time for the design/ construction of a scripting language

Reaction: The time that was needed to construct a general idea for the scripting language took more time than we had planned. This is mostly because we all started separately on the tasks without first discussing clearly (in details) how we wanted to approach the scripting language. Eventually we did discuss this together with the whole group but some code had to be adjusted in order to meet our agreements. This caused us to have less time left for the actual implementation of the commenting, constraining and chunking operations. We had planned to implement those transformations fully this week but only achieved to implement the basics. Commenting is almost done but chunking and constraining still need to be implemented for a large part. These tasks will be shifted to the next sprint. From now on, when implementing a new feature, we will first make a detailed design and agree with the whole group on it before starting to implement it.

Problem 2

Description: Use pull requests more often

Reaction: During this sprint we had some problems merging pull requests into the master branch. A few branches had to be manually merged since they differed too much from the master branch. This problem can mainly be solved by using pull requests more often and thereby having less changes in the code (smaller branches) before doing a pull request. This is why we will try to use more pull requests when implementing a new feature in the next sprint, to make the merging easier.

Adjustments for the next Sprint Plan

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

- Again make the planning more adaptive: keep some room for extra tasks

During this sprint we ran into problems with our planning for the second time. Our schedule got really tight because of extra time needed to design the basics of the scripting language which we didn't anticipate on. We discussed this problem together with the Software Engineering TA and decided to leave at least two hours room for each group member. Last minute changes can always happen and tasks can take more time than estimated. By leaving more room in our schedule, a lot of last-minute stress could be avoided.

- Assign tasks to only one group member

In the previous sprints we assigned some tasks to multiple group members since we use pair programming for large tasks. This caused some uncertainty about the division of the responsibility for those tasks. If the task wasn't finished shortly before the deadline, which of the two (or three) members needed to finish it? That is why we decided to divide larger tasks into smaller tasks (if possible) and assign those to only one group member from now on. The one who is assigned to a task has full responsibility - even if you got help or collaborated with another group member - to make sure that all work is done, evaluated and committed before the deadline.