

# Programming Life - SCRUM plan #1

Group 5/E:

Felix Akkermans

Niels Doekemeijer

Thomas van Helden

Albert ten Napel

Jan Pieter Waagmeester

April 21, 2012

## 1 Introduction

In this SCRUM plan we define the tasks we want to complete in this sprint. Every task is assigned to one or two developers and an estimate is provided for the effort required.

For each SCRUM run a milestone is created at GitHub, with issues for the tasks selected. The issues for this milestone can be found on: <https://github.com/FelixAkk/synthbio/issues?milestone=8>.

## 2 Selection and assignment of tasks

Because this is the first iteration, it is hard to predict how things work out. Lots of things are to be sorted out and we have to become familiar with the tools to be used.

### Available time

The time available is four afternoons in two weeks by five people. That's about sixteen hours per person, however, some time should be reserved for the scheduled meeting on wednesday. So the total effort available is about 75 man hours. When writing this document, we made an error in this calculation resulting in a total effort available of 40 man hours, so during the estimation of the tasks we tried to fit all the work in those 40 hours. This might be the reason why there was some underestimation for the tasks in this iteration.

A list of tasks, assignments and effort estimations is included in a table.

### Presentation

Results for this iteration will be presented thursday april 5, 16:30 in the library (project room Australia).

Task	Developer	Effort	
		estimated	actual
#2 Server: Tomcat server that handles HTTP requests HELLO WORLD and class scaffolding	Niels/Albert	2*10	17
#3 Server/Client: Design and document JSON format for circuit (.syn) and protein list.	Jan Pieter	4	3
#4 Client: JavaScript scaffolding	Thomas	8	
#5 Client: Create basic GUI	Felix	8	
#6 Server/client: create and display connection state by ping/pong heartbeat	Niels/Albert	2	3
#7 Client: Show available basic gates	Jan Pieter	3	1 + ??Felix
<b>Total</b>		45	??
<b>Optional tasks<sup>1</sup></b>			
#8 Server: Parse and serve list of Proteins <sup>2</sup>	Jan Pieter <sup>3</sup>	4	16
#11 Client: Display list of Proteins	Jan Pieter <sup>4</sup>	2	2
<b>Total</b>		6	18
<b>Added during iteration<sup>5</sup></b>			
# Ant build environment	Jan Pieter	-	2
<b>Total</b>		-	2
<b>Grand Total</b>		41	??

### 3 Reflection on this iteration

In this section we will give a quick review on this iteration. First we will comment on our initial planning and second we will review the implementation phase and the workflow.

#### 3.1 Planning

The estimated efforts were not that far off the actual time needed for implementation and we are confident that our estimations will improve as the project continues. We did, however, forget to plan a few obvious tasks. For example, code reviewing, tool selection (testing HTML responses and JSON-serialisation) and creating a building environment are tasks that could have been foreseen. Also the time needed for everyone to setup their working environment was underestimated. A lot of these tasks were one-time efforts, but they can consume precious time. We think the number of these small tasks will reduce, as this was the very first iteration. We now have a starting point that can be built on. For next iterations, we really have to think things through, to make sure there will be as little unforeseen subtasks as possible.

<sup>1</sup>Things from next iterations that could be done if sufficient time is available

<sup>2</sup>Only after list is supplied by Alexey

<sup>3</sup>Original plan was *Niels/Albert*, but Jan Pieter finished his tasks...

<sup>4</sup>Initially unassigned

<sup>5</sup>Significant tasks not planned before.

## 3.2 Implementation

The implementation phase for this iteration didn't go as smoothly as we hoped it would go, but it did give us a few points to work on. One of these points is communication; there was little communication between members. This, combined with a low number of commits, made it hard for members to see what everyone was working on and what the overall progress was. Only when the deadline was approaching, the commits started to flow and the project advanced quickly. Late commits make code reviewing and integration testing harder. That is why we want to commit early and commit often in the next iteration.

We are happy with the delivered code and product. Test-driven development was applied correctly and we think we have built a solid base. For this base we needed a few (previously undocumented) tools:

### Build environment

We decided to use Apache ant to build the sources and provide easy access to the JUnit test suites. Currently, four targets are present.

### JSON serialisation

Selected the library from <http://json.org/java> since it is lightweight.

### HTML/JSON Unit testing

We have selected HtmlUnit (<http://htmlunit.sourceforge.net/>) to help test the server applets. This unit acts as a browser (without the GUI) and fetches pages from the server. This helps testing the JSON responses. Unfortunately, we have found no easy way to compare JSON objects in Java. That is why we are comparing the objects as strings. JSON objects are unordered by definition, but we make sure the objects have a predictable layout (alphabetical order).