# Programming Life - Test and implementation plan

Group 5/E:
Felix Akkermans
Niels Doekemeijer
Thomas van Helden
Albert ten Napel
Jan Pieter Waagmeester

March 18, 2012

## Contents

# 1 Introduction

In this report the different testing techniques we will use for this project will be explained. Because our solution has a clear division between server and client and because these will be developed in different programming environments, we will also need different testing strategies for the client and server. In chapter 2 a prioritization of the requirements can be found using the MoSCoW system. Chapter 3 will explain how we will test the server and client, and what strategies we will use. Lastly chapter 4 will cover the risk analysis, describing the risks for the successful implementation of the system.

# 2 MoSCoW prioritization

| Requirement | MoSCoW |
|---|---|
| Circuit Abstraction | Must |
| Protein specification | Must |
| Available proteins | Must |
| Export XML | Must |
| Interfering signals | Must |
| Invalid signals | Should |
| Re-use BioBricks | Should |
| Multi-client | Could |
| Local back-up | Could |
| Import XML | Wont |
| Biological plausibility | Wont |

# 3 Implementation and tests

## 3.1 Order of implementation of features

### 3.1.1 Iterations

### 3.1.2 Milestones

## 3.2 Server Test plan

### 3.2.1 Unit testing

### 3.2.2 Integration testing

### 3.2.3 Acceptance testing

## 3.3 Client Test plan

### 3.3.1 Unit testing

### 3.3.2 Integration testing

### 3.3.3 Acceptance testing

### 3.4 Testing Client-server integration

Integration testing is done when the individual software modules are combined and need to be tested as a whole. One approach is to use unit tests which test the whole system. The server can be run on the same machine as the client, in this way the unit tests can test these two components.

## 4 Risk analysis

(what are the risks for the successful implementation of the system?)