Code Inspection Report

Anti-Spam Configuration Software
Development Project

BSc in LIGE-PL
Academic Year 2017/2018 - 1º Semester
Software Engineering I

Group 102
69565, Rodolfo Arnaldo, I-PLC1
65345, Tiago Rodrigues, I-PLC1
68958, Catarina Carriço, I-PLC1
69399, Ricardo Duarte, I-PLC1

ISCTE-IUL, Instituto Universitário de Lisboa
1649-026 Lisbon
Portugal

December 18th 2017

Table of Contents

Introduction

AntiSpamConfigurationForBalancedProfessionalAndLeisureMailbox is a learning software that configures a list of rules of a mailbox in order to minimize the spam mails received. Giving a log of spam emails and a log of ham emails, the system configures the rule's weights and gives the user the best outcome for a professional and leisure mailbox.

1. Code inspection – Main GUI and Manual Configuration GUI

Description of the software component being inspected

Meeting date: *Meeting duration:* *Moderator:* *Producer:* *Inspector:* *Recorder:*	*20/11/2017* *20 minutes* *Rodolfo Arnaldo* *Tiago Rodrigues* *Tiago Rodrigues* *Catarina Carriço*
Component name (Package/Class/Method):	*Package antiSpamFilter.GUI*
Component was compiled:	*Yes*
Component was executed:	*Yes*
Component was tested without errors:	*Yes*
Testing coverage achieved:	*45,1% (not considered to the final coverage 75%)*

a. Code inspection checklist

The checklist for Java code inspection used in this project is available at http://www.cs.toronto.edu/~sme/CSC444F/handouts/java_checklist.pdf and in blackboard ES1 page.

b. Found defects

Identify and describe found defects, opinions and suggestions.

Found defect Id	Package, Class, Method, Line	Defect category	Description
1	antiSpamFilterGUI, constructor	Validation	The system exits in the close button without confirming
2	antiSpamFilterStyles, ATextField	Visual	The ATextField should have a placeholder

c. Corrective measures

1, Tiago Rodrigues will correct the defect until 25th November.
2, Tiago Rodrigues will implement the placeholder until 25th November.

d. Conclusions of the inspection process

The main GUI class, the manual configuration GUI class and the Styles class work very well together. They have most of the validations working and the system cannot exit with an exception or error. After correcting the two defects, the GUI is ready to be show to the client in order for him to validate the classes.

2. Code inspection – Emails, Rules, Stream classes

Description of the software component being inspected

Meeting date: *Meeting duration:* *Moderator:* *Producer:* *Inspector:* *Recorder:*	*29/11/2017* *15 minutes* *Rodolfo Arnaldo* *Rodolfo Arnaldo* *Tiago Rodrigues* *Ricardo Duarte*
Component name (Package/Class/Method):	*Package antiSpamFilter.emails,* *Package antiSpamFilter.rules*
Component was compiled:	*Yes*
Component was executed:	*Yes*
Component was tested without errors:	*Yes*
Testing coverage achieved:	*77,8% emails package, 100% rules package*

a. Code inspection checklist

The checklist for Java code inspection used in this project is available at http://www.cs.toronto.edu/~sme/CSC444F/handouts/java_checklist.pdf and in blackboard ES1 page.

b. Found defects

Identify and describe found defects, opinions and suggestions.

Found defect Id	Package, Class, Method, Line	Defect category	Description
1	antiSpamFilter.EmailStream	FileReader	If some line is corrupted, the exception doesn't close the file. (warning)

c. Corrective measures

1, Catarina Carriço will correct the defect until 5th December.

d. Conclusions of the inspection process

The Email and Rule structure are simple and well implemented. The test coverage is comfortable and all the main attributes and variables are according to the establish. All the names and comments are in English.

3. Code inspection – Validation classes

Description of the software component being inspected

Meeting date: *Meeting duration:* *Moderator:* *Producer:* *Inspector:* *Recorder:*	*05/12/2017* *15 minutes* *Rodolfo Arnaldo* *Ricardo Duarte* *Tiago Rodrigues* *Catarina Carriço*
Component name (Package/Class/Method):	*Package antiSpamFilter.validations*
Component was compiled:	*Yes*
Component was executed:	*Yes*
Component was tested without errors:	*Yes*
Testing coverage achieved:	*81,5%*

a. Code inspection checklist

The checklist for Java code inspection used in this project is available at http://www.cs.toronto.edu/~sme/CSC444F/handouts/java_checklist.pdf and in blackboard ES1 page.

b. Found defects

Identify and describe found defects, opinions and suggestions.

Found defect Id	**Package, Class, Method, Line**	**Defect category**	**Description**
1	antiSpamFilter.ReadLOG	Validation	The confirmation of the name of the file should not exist
2	antiSpamFilter.ReadCF	Language	The names of the variables are not in English
3	antiSpamFilter.ReadCF, line 44	Optimize	The FOR procedure is unnecessary

c. Corrective measures

1, Catarina Carriço will correct the defect until 8th December.
2, Tiago Rodrigues will correct the defect until 8th December.
3, Rodolfo Arnaldo will correct the defect until 8th December.

d. Conclusions of the inspection process

Despite the defects, the validation classes are working well. The validation of the files is correct and blocks the tests of fake files.

4. Code inspection – AntiSpamFilter classes

Description of the software component being inspected

Meeting date: *Meeting duration:* *Moderator:* *Producer:* *Inspector:* *Recorder:*	*18/12/2017* *20 minutes* *Rodolfo Arnaldo* *Tiago Rodrigues* *Tiago Rodrigues* *Catarina Carriço*
Component name (Package/Class/Method):	*Package antiSpamFilter*
Component was compiled:	*Yes*
Component was executed:	*Yes*
Component was tested without errors:	*Yes*
Testing coverage achieved:	*79,4%*

a. Code inspection checklist

The checklist for Java code inspection used in this project is available at http://www.cs.toronto.edu/~sme/CSC444F/handouts/java_checklist.pdf and in blackboard ES1 page.

b. Found defects

Identify and describe found defects, opinions and suggestions.

Found defect Id	**Package, Class, Method, Line**	**Defect category**	**Description**
1	antiSpamFilterAutomaticConfiguration	Language	The names of the variables are not in English
2	antiSpamFilterManualConfiguration	Language	Some comments are not in English

c. Corrective measures

1, Catarina Carriço will correct the defect until 20th December.
2, Ricardo Duarte will correct the defect until 20th December.

d. Conclusions of the inspection process

The overall processes of the main classes are functional and well implemented. The use of the other classes is well constructed and called. The software is fully functional.