RomanNumerals.cpp Inspection Plan

Palmer Cluff, Joshua Jolley, Austin Moore

# 1 OVERVIEW

We are implementing a modified version of Fagan’s inspection method (FIM) to inspect RomanNumerals.cpp. FIM consists of a team of inspectors with assigned roles following predefined steps to find the flaws in a document [1]. We chose this method for its well defined roles and process steps. Our modified method includes the following three roles: moderator, reader, and tester. Our modified process consists of the following four steps: planning, overview, preparation, examination. We have eliminated the author role, the rework step, and the follow-up step from FIM because the duties and work related to them are beyond the scope of this assignment. Our final modification to FIM was to enlarge the scope of our included roles to include the applicable parts of Boehm’s reliability and availability checklist.

# 2 ROLES

## 2.1 MODERATOR

### 2.1.1 DESCRIPTION

This is the key person in the inspection who manages the team and coordinate the inspection process. The moderator will coordinate using email, Google Docs, and Slack.   
The moderator is responsible for:

* assembling the documented bugs and the submission of the assembled document.
* the manual technique of simple scenarios [2].
* finding defects relating to *error messages and diagnostics*, as defined by Boehm’s reliability and availability checklist [2].

### 2.1.2 ASSIGNMENT

Palmer Cluff

## 2.2 READER

### 2.2.1 DESCRIPTION

The programmer who paraphrases code during the meeting in human readable terms [1].  
The reader is responsible for:

* the manual technique of reading[[1]](#footnote-0) [2].
* finding defects relating to *reliable input handling* as defined by Boehm’s readability and availability checklist [2].

### 2.2.2 ASSIGNMENT

Austin Moore

## 2.3 TESTER

### 2.3.1 DESCRIPTION

The programmer who reviews the product from the testing point of view. The tester is responsible for:

* creating and running unit and integration tests as necessary to exercise the code and find defects.
* the manual technique of simple automated models[[2]](#footnote-1) [2].
* finding defects relating to *reliable execution* as defined by Boehm’s readability and availability checklist [2].

### 2.3.2 ASSIGNMENT

Joshua Jolley

# 3 PROCESS

## 3.1 PLANNING

The planning step consists of forming an inspection team and assigning roles to each member [1]. The creation of this document, which includes the assignment of roles and an explanation of process definitions, is our planning step.

## 3.2 OVERVIEW

The overview step informs the inspection team about the product the document to be reviewed is for [1]. Our overview step consists of reading the assignment descriptions and rubric for Teach 04 and Ponder 04.

## 3.3 PREPARATION

Each member of the inspection team will independently inspect RomanNumerals.cpp. While inspecting, members will fulfill the duties of their role as described in section 2. In addition to their roles, each member will briefly document the bugs they find. This document must include a technically descriptive title, the steps required to reproduce the defect, and the line or lines of code that cause the defect.

## 3.4 EXAMINATION

The inspection team will meet in Slack at a time coordinated by the moderator. The scenarios, tests, and documented bugs found during preparation will each be presented to the team. The team will verify the validity of the bugs presented. The moderator will then assemble the verified bugs into a document for submission. Upon completion, the team will review the bug document, making corrections as necessary. After any necessary corrections have been made, each member of the team will append a brief narrative describing the teamwork that occurred during this assignment. Once all members have completed this, the moderator will submit the completed document to group 5’s discussion board

# SOURCES

[1] A. Aurum et al., "State-of-the-art: Software inspections after 25 years," Software Testing, Verification and Reliability, vol. 12, no. 3, pp. 133-154, Feb. 2002,   
[Online] Available: [http://onlinelibrary.wiley.com/doi/10.1002/stvr.243/pdf](https://content.byui.edu/items/8bcbc45e-012a-48e0-800d-7082dd962f15/1/?.vi=file&attachment.uuid=285b9a01-bb33-40e1-9244-b041b0da45e7)

[2] B. Boehm, "Verifying and validating software requirements and design specifications," IEEE Software, vol. 1, no. 1, pp. 75-88, Jan. 1984,   
[Online] Available: [http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1695100&isnumber=35712](https://content.byui.edu/items/8bcbc45e-012a-48e0-800d-7082dd962f15/1/?.vi=file&attachment.uuid=bb5fb287-2175-4a68-845f-080b954bb124)

1. The scope of this assignment is small enough that many parts of this technique won’t apply. Be judicious in deciding what to apply. [↑](#footnote-ref-0)
2. See footnote 1. [↑](#footnote-ref-1)