CECS 543

CECS 543 Metrics Suite Vision

Version <1.0>

ReVision document History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

Iteration 1
Due Feb 20
Start of Lab
Demonstrate in lab working Function Points GUI program as described in this document.

Submit to Beachboard a Use Case for Calculating Function Points with code sizing
Due Feb 8
End of Lab

Table of Contents

Vision

1. Introduction

The purpose of this document is to collect, analyze, and define high-level needs and features of the CECS Metrics Suite. It focuses on the capabilities needed by the stakeholders, and the target users, and **why** these needs exist. The details of how the CECS Metrics Suite fulfils these needs are detailed in the use-case and supplementary specifications.

1.1 Purpose

This document describes the high level features of the new product CECS 543 Metrics Suite. This product allows the user to compute several common software lifecycle metrics including Function Points, Object Points, the Halstead Metrics, McCabe's Cyclomatic Complexity, Class-Oriented Metrics by Chidamber and Kemerer, Lines of Code, the Zero-failure testing metric and the Software Maturity Index.

1.2 Scope

This document describes only the Spring 2018 semester project CECS 543 Metrics Suite.

1.3 Definitions, Acronyms and Abbreviations

There will be many items that need to be defined but these will be examined at a future point this semester.

1.4 References

The project will reference several chapters from the course this semester including Chapters 23 and 26.

2. Positioning

2.1 Business Opportunity

NA.

2.2 Problem Statement

The problem of	computing various difficult software development metrics
affects	the team's ability to collect sufficient information about the quality of the software development lifecycle and its product
the impact of which is	reduced effectiveness of the lifecycle team to produce quality code and learn from our mistakes.
A successful solution would be	to automate to the largest extent possible the collection and computation of these metrics and maintain them for future reference.

2.3 Product Position Statement

2.4 Market Demographics

NA.

2.5 Stakeholder Summary

Name	Represents	Role
Software Engineers	Themselves	They are both the developers and the users of the product.

2.6 User Summary

Name	Description	Stakeholder
Software Engineers	Create requirements, specification, design and code of the product and use the product when completed to aid in future software development.	One in the same.

2.7 User Environment

The user environment consists of a typical software development organization. It hopefully employs a software development process model over which various measures can be collected and used to assist the organization in current and future development. It probably also smells of coffee and stale potato chips.

2.8 Stakeholder Profiles

2.8.1 Software Engineer

Representative	YOU!
Description	Budding young software engineers anxious to learn, improve and grow in software engineering.
Type	A range of expertise from novice developers to seasoned software engineers.
Responsibilities	Both creator and user of the product.
Success Criteria	Success is judged by meeting all the professor's requirements.
Involvement	Both creator and user of the product.
Deliverables	Correct software increments throughout the semester.
Comments / Issues	Ummmm.

2.9 User Profiles

Same as stakeholders.

3. Product Overview

The CECS 543 Metrics Suite is a software product that provides a Graphical User Interface (GUI) to software engineering and other developers for the collection, calculation, review m organization and retrieval of several commonly used software lifecycle metrics or measurements.

The product may be written in any object-oriented language and may run under IOS, Linux, or Windows operating systems.

3.1 Product Perspective

This product is a stand alone software package. However, it does require access to the organization's code base for the computation of several metrics.

3.2 Summary of Capabilities

This product provides software engineers and other development team members with the ability to collect various software development lifecycle metrics.

For the requirements phase the product allows the computation of Function Points and Object Points.

For the design phase several class and object-oriented metrics compute design complexity and issues with class relationships and issues of classes to methods.

For overall complexity of individual modules it computes McCabe's Cyclomatic Complexity and the Halstead metrics. The product provides code parsing capabilities for counting these metrics.

For overall size, Lines of Code can be computed if we can agree on what a line of code is.

For the testing phase the product computes he Zero-Failure method.

For the deployment phase it computes the Software Maturity Index.

The product allows for the input, computation and storage of results for all the foregoing metrics. Metrics are stored by individual projects. This allows for cross-checking of predicted results with actual results. More exact descriptions will be provided in later documents.

3.3 This product Assumptions and Dependencies

None.

3.4 Cost and Pricing

NA.

3.5 Licensing and Installation

Na.

4. Product Features for Iteration 1

This section describes at a high-level the beginning set of features of the CECS 543 Metrics Suite.

As an overall and persistent feature, the product will be in a windowed application running in a graphical environment. Menu systems and sub-systems, frames, panels, radio buttons, check boxes, text boxes, combo boxes and so on are used to communicate from the user to the product and from the product to the user.

THIS PRODUCT IS NOT TO BE DONE AS A WEBPAGE OR WEBSITE.

What follows is a list of basic features and features for the first iteration.

4.1 A windowed application in an object-oriented language.
4.2 Menus for File, Edit, Preferences, Metrics, Help (see Figure 1)

4.3 Under File, New, Open, Save and Exit. Under Edit, nothing yet. Under Preferences, Language. Under Metrics, Function Points. Under Help, nothing yet. Under Functions Points, Enter FP Data
4.4 Minimize, Maximize and Close buttons (close exits the application)
4.5 Language selection dialog in a separate frame (see Figure 2) (from Preferences)
4.6 Save selected language
4.7 "Enter FP Data" creates a new pane in the main (and repeats)
4.8 Separate panes for each metrics type computed (see Figure 4)
4.9 Input function point data (see Figures 4 and 6)
- **4.9.1 Input "External Inputs"**
- **4.9.2 Input "External Outputs"**
- **4.9.3 Input "External Inquiries"**
- **4.9.4 Input "Internal Logical Files"**
- **4.9.5 Input "External Interface Files"**

4.10 Radio buttons for Weighting Factors Simple, Average and Complex for each from 4.9
4.11 Default Weighting Factors to Average for all from 4.9
4.12 Keyboard input for all from 4.9.
4.13 Totals for each input from 4.9 multiplied by its corresponding Weighting Factor (see Figure 5x)
4.14 A separate dialog for entering Value Adjustment Factors (VAF) (see Figure 5)
4.15 Keep VAF fields. If the VAF dialog is opened again, restore the values of each factor from the previous call.
4.16 Change VAF fields when changed by the user.
4.17 Display VAF sum as a single field in Function Points pane (see Figure 4)
4.18 Display Total Count of values from 4.11 (see Figure 6)
4.19 Compute Function Point value
4.20 Display Function Point value in a single field formatted with commas and decimal
4.21 On the FP pane, provide a button to open the VAF dialog
4.22 Default all VAF values to 0
4.23 Display each VAF value as a combo box (see Figure 5)
4.24 Validate values entered for 4.9 are non-negative integers
4.25 Display currently selected language on FP pane or "None"
4.26 Provide a label next to the currently selected language stating "Current Language"
4.27 Provide a button labeled "Change Language" that opens the dialog from 4.5 on the FP pane to allows user to pick or change the language
4.28 Provide a button on the FP pane computes the code size for the currently selected language and current FP value (see Figures 4 and 6)
4.29 The File, Save operation saves the following fields: current language, all VAFs, all five inputs from 4.9, all weighting factors, input total (from 4.13), the computed FP (if available)
4.30 The File, Open operation opens a File Open dialog allowing the user to select a file to open.
4.31 The project files created by CECS 543 Metrics Suite will have the extension .ms.
4.32 The product defaults to looking for .ms files to open.
4.33 Once selected, a project file is opened and a FP pane created and displayed and the

corresponding values in the file copied to their respective controls in the FP pane.
4.34 The File, New operation creates a new project with no data or panes.
4.35 The project files contain the name of the project, the creator's name and a list of all panes in the project and all data related to each pane.
4.36 Provide a dialog to create a new project. (see Figure 3)
4.37 Change Title Bar of Application from "CECS 543 Metrics Suite" to "CECS 543 Metrics Suite - Project Name" (see Figures 4-8)

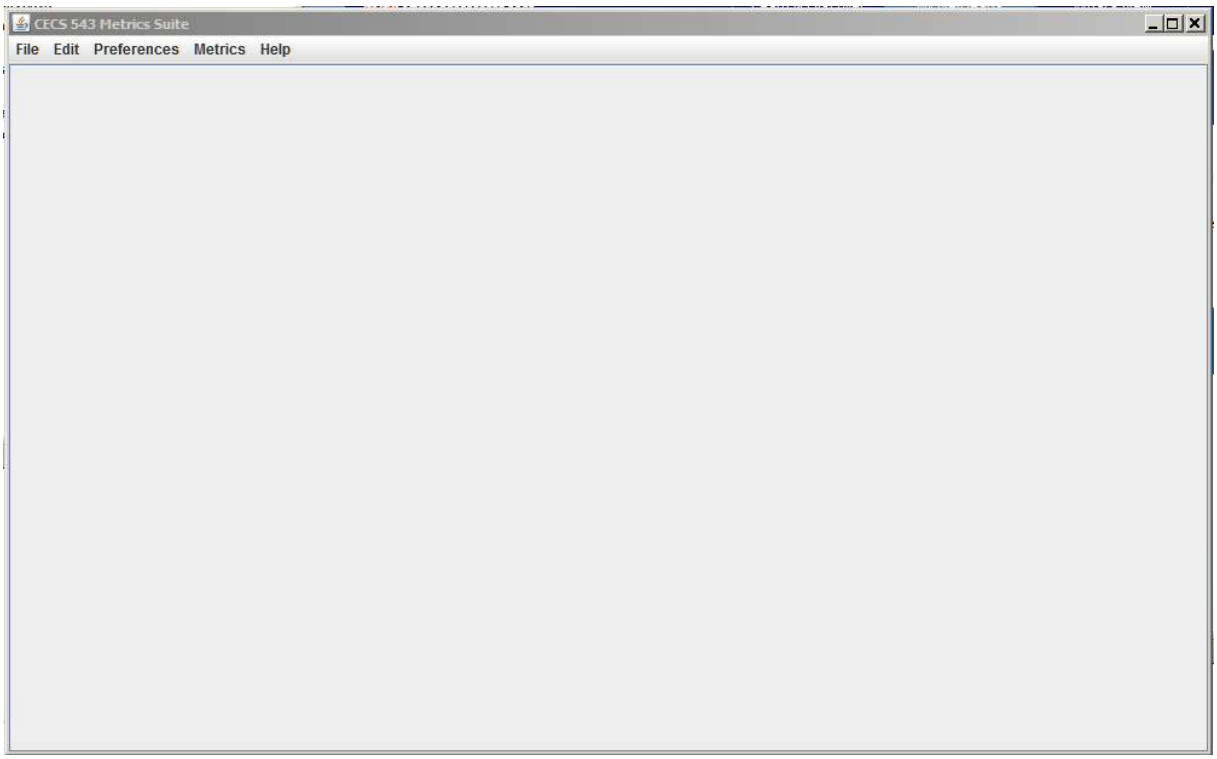

Figure 1Application at start up

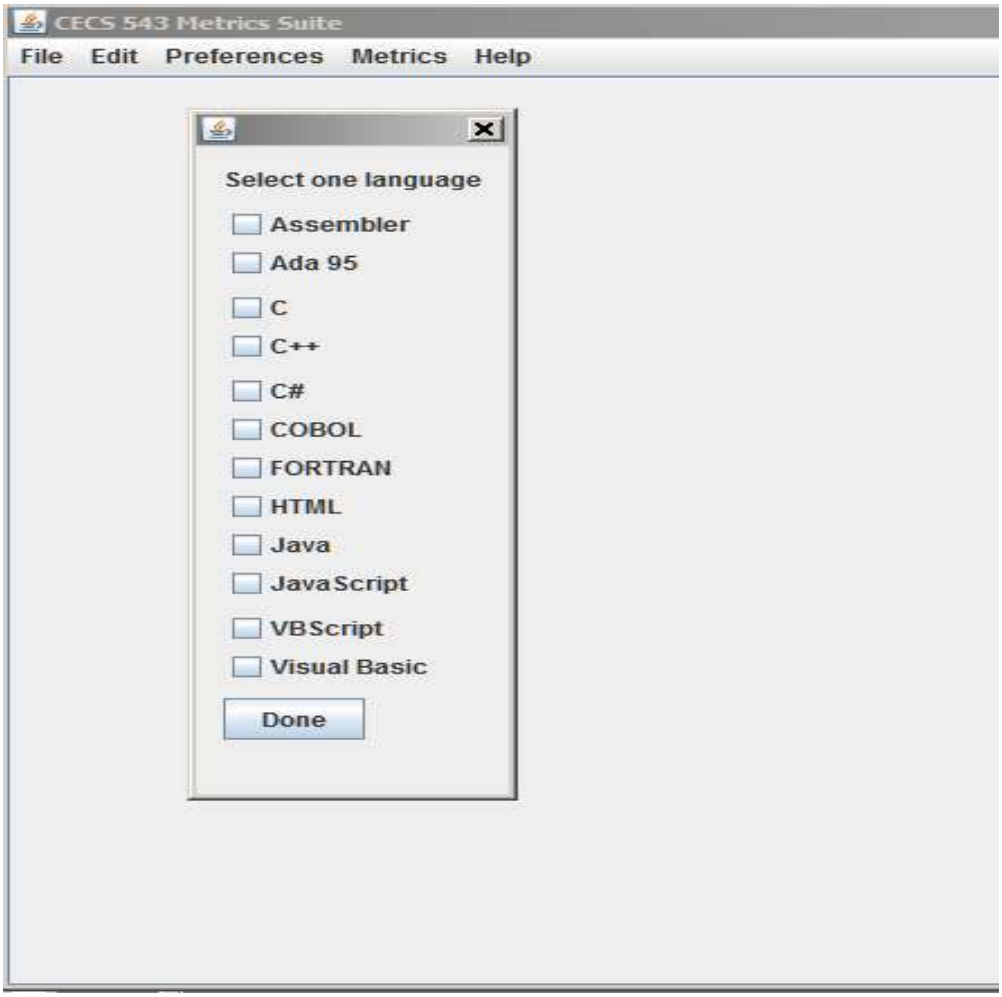

Figure 2 Language selection dialog

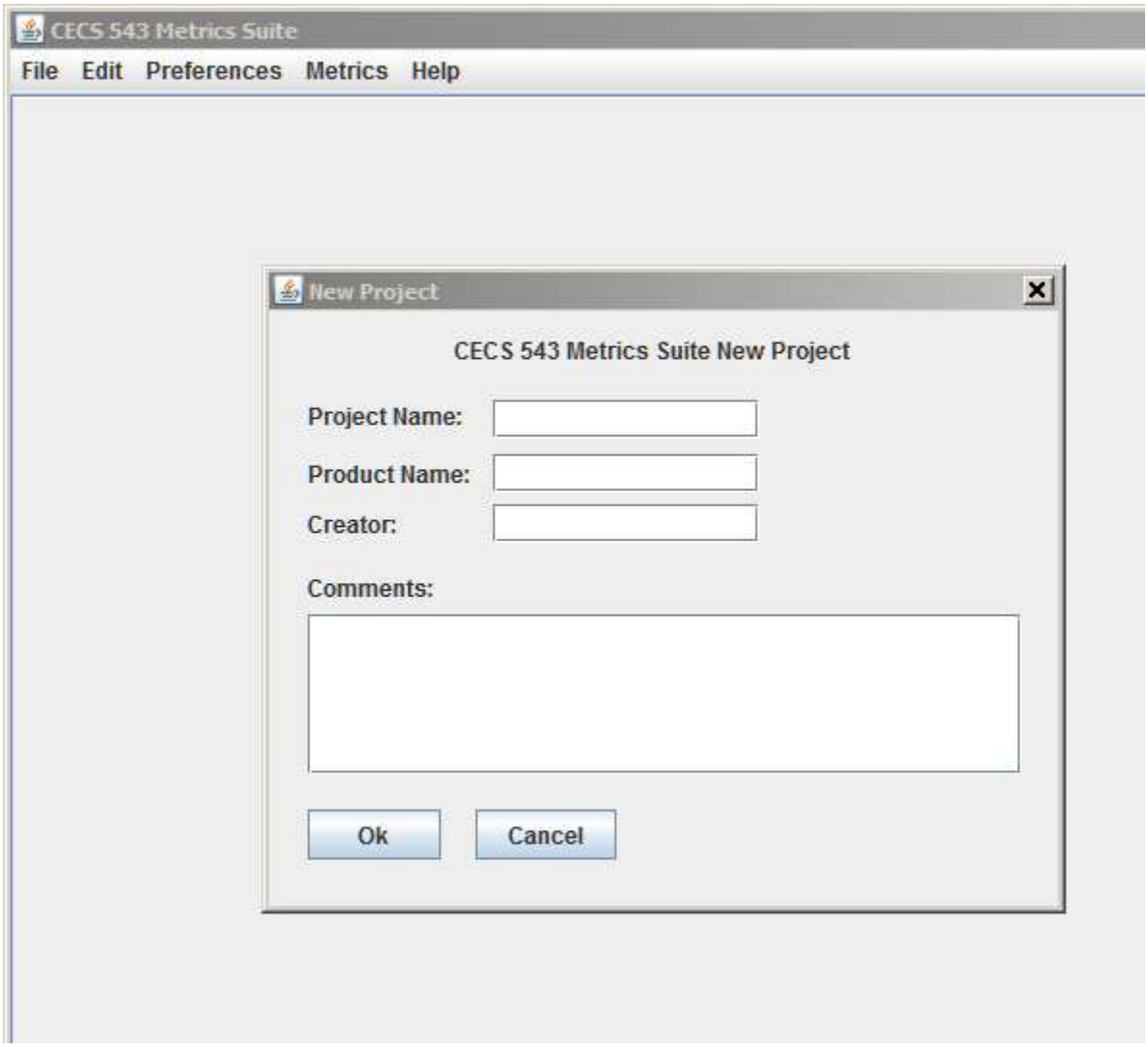

Figure 3 New Project Dialog

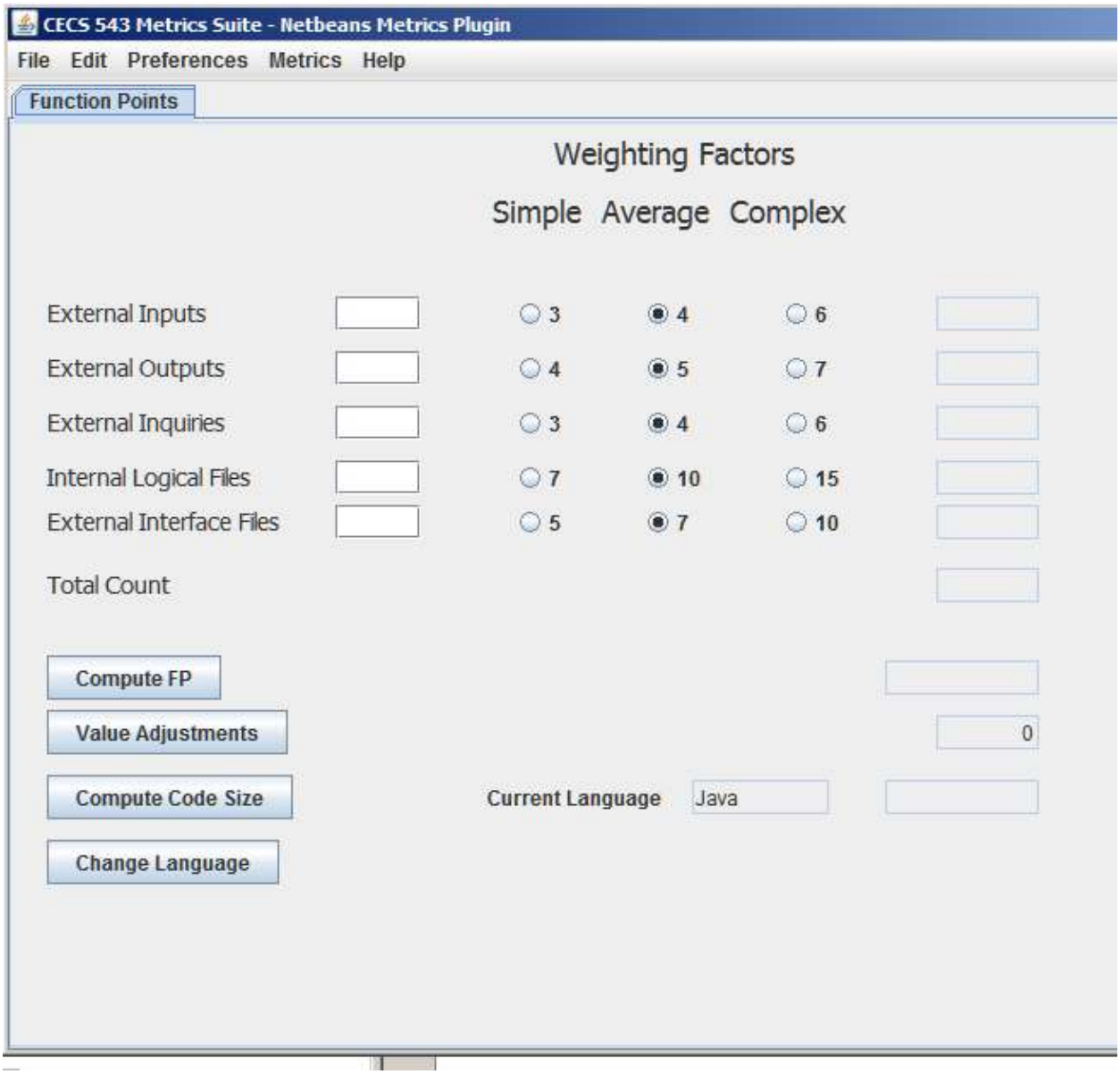

Figure 4 Function Point pane with default data

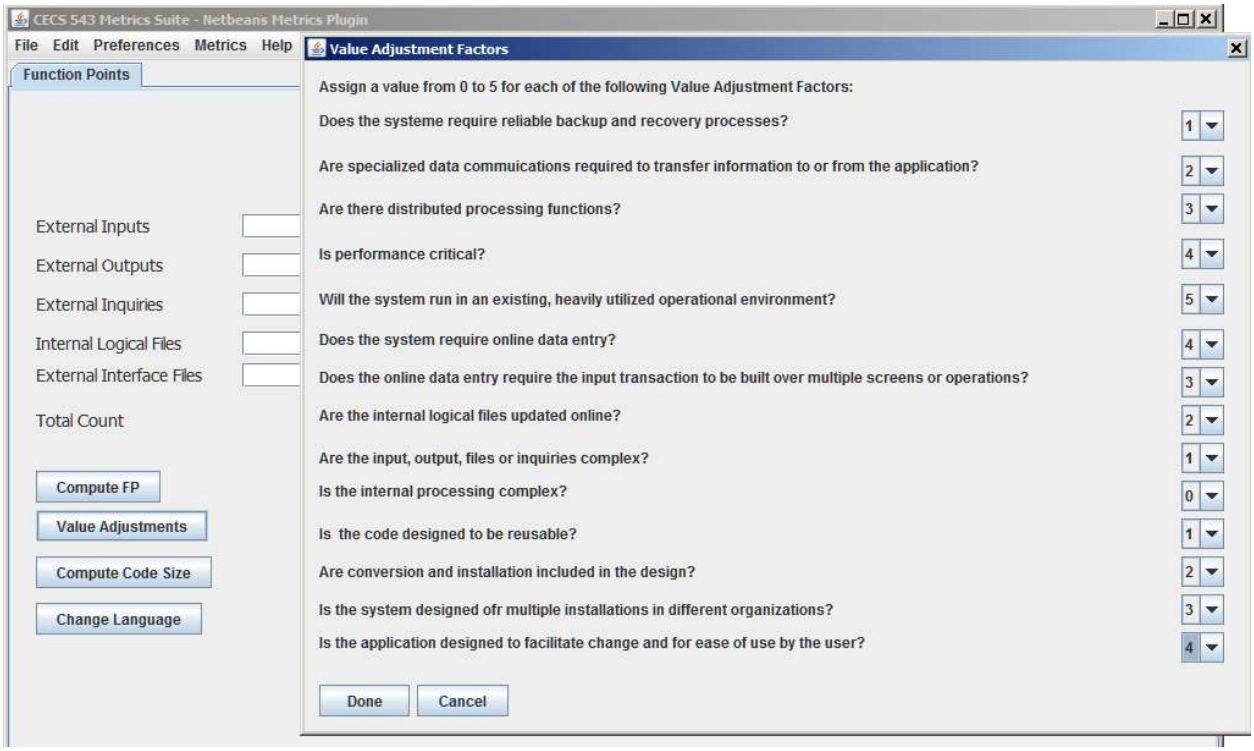

Figure 5 Dialog for Value Adjustment Factors

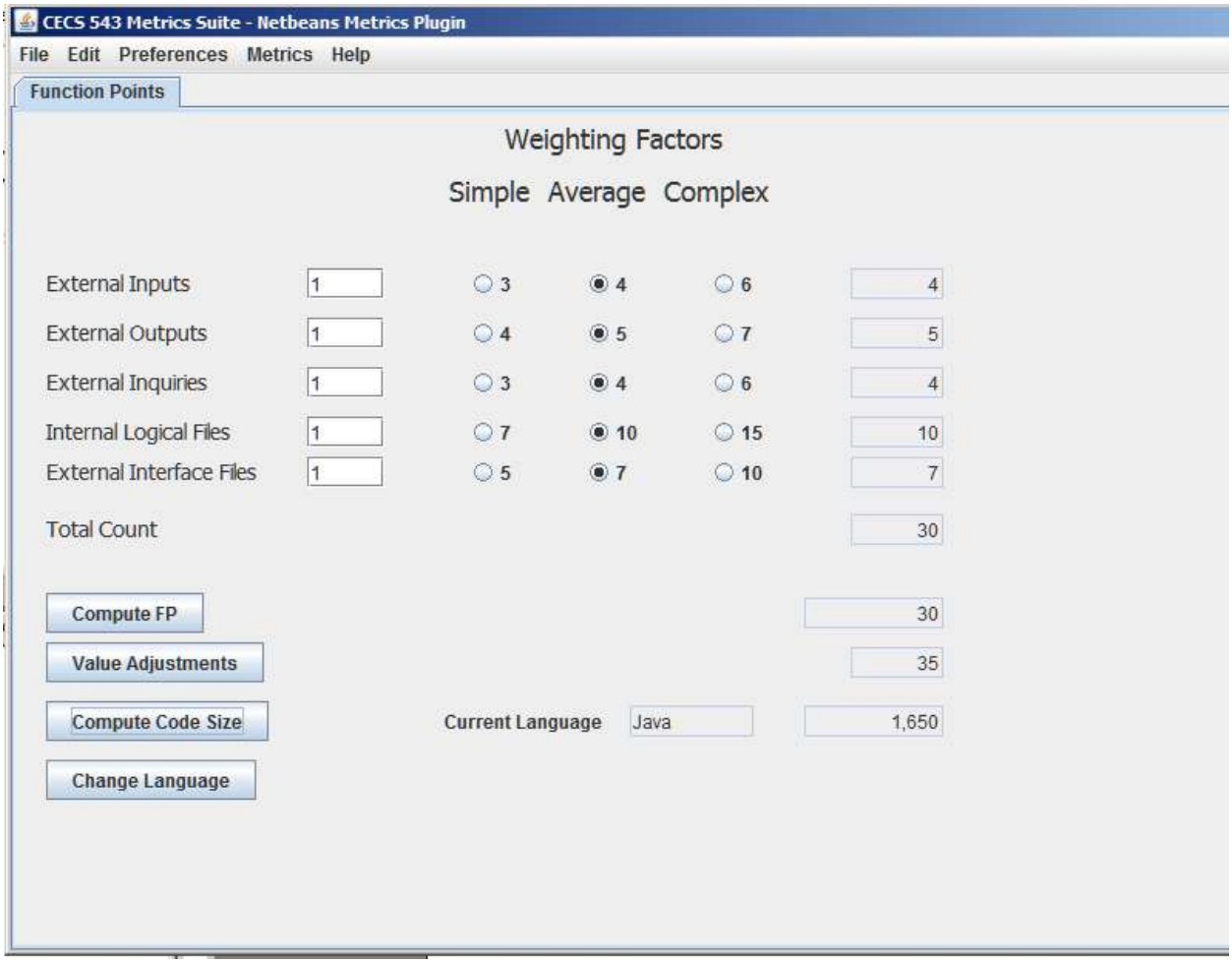

Figure 6 Computed inputs, Total Count and FP with VAFs

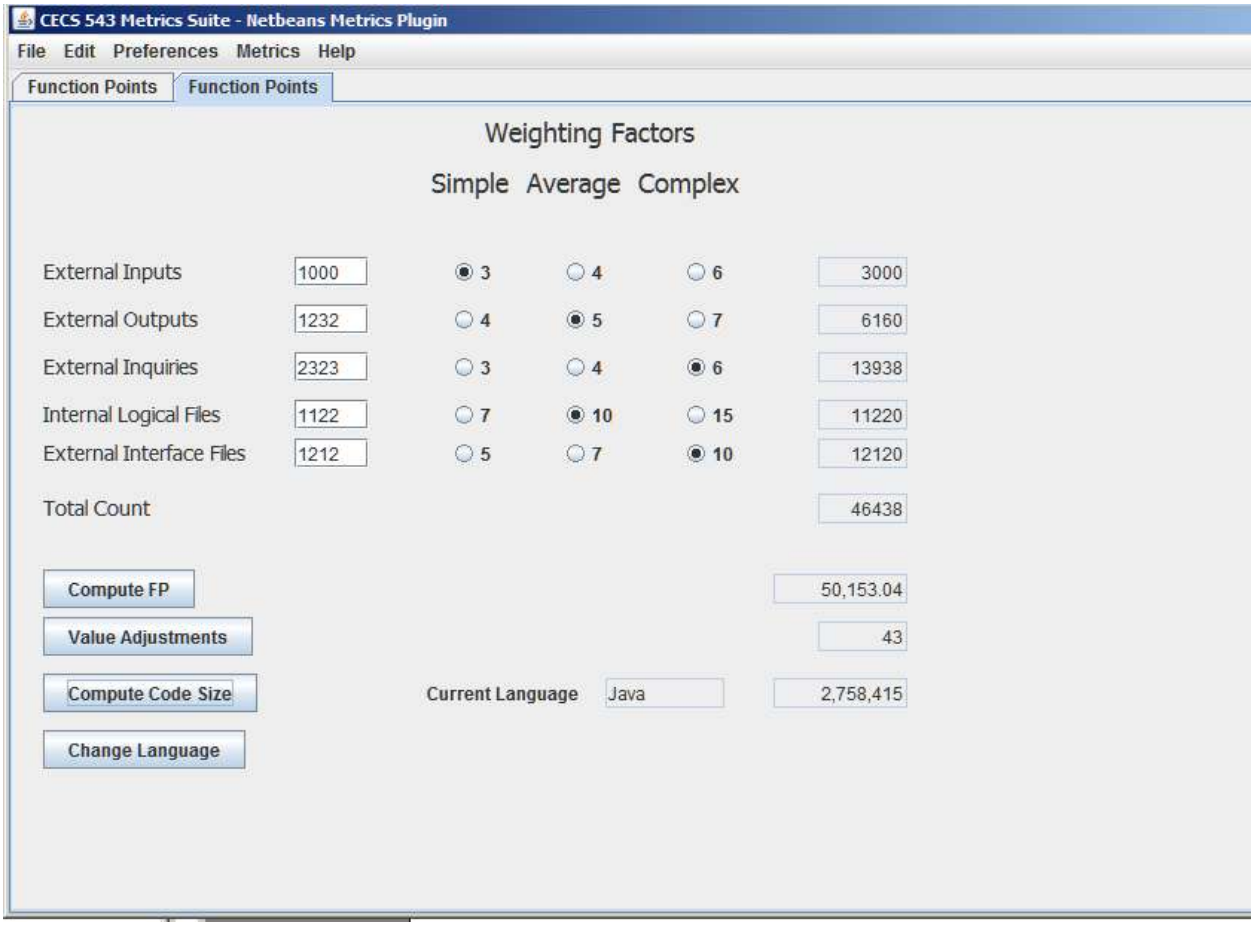

Figure 7 Large project in COBOL with code size

5. Constraints

Must be done in an OO language. Acceptable languages are C++, C#, Java, Ruby, Objective C.