CMPSC 290
Principles of Software Development
Fall 2013

Laboratory Assignment Six: Automatically Characterizing Programs and Test Suites

Introduction

In the previous laboratory assignments, you have learned about the tools that we will use during the remainder of this semester to specify, design, implement, test, and document Java programs. You have also had several experiences with working in progressively larger teams to complete the phases of the software life cycle, with a recent focus on the elicitation of software requirements and the planning of software projects. In this assignment, you and your team will install, configure, and use software tools that automatically calculate design and implementation metrics that characterize programs and test suites. One week from now, you will present the findings from your analyses.

Calculating Design Quality Metrics with JDepend

After you have written a requirements document for your system and completed the description of the architecture, you must create a design for your program. Of course, it is important to evaluate the quality of your design. If you already have a (partial) implementation of a program and you want to evaluate the quality of its design, you can use a tool called JDepend. You can learn more about JDepend by visiting `http://clarkware.com/software/JDepend.html`. What are the design quality metrics that JDepend calculates? What are the meanings of these metrics? How can you use these metrics to better understand and modify the design of your Java programs? As one of the deliverables for this assignment, your team should use LaTeX to prepare a document that contains formal definitions and equations that describe all of the design quality metrics.

Once you and your team have investigated and discussed the features provided by JDepend, you should pick two separate systems that have been previously implemented by members of your team. Now, copy the source code from these two projects — and any other files that your team deem to be relevant — to the version control repository that you will use for this project. After downloading and installing JDepend, you should add rules to each project's Ant build system that can run the JDepend analysis on the byte code. What does the output of JDepend tell you about the quality of each project's design? Do the chosen programs have good designs? Why or why not?

Source Code Measurement with JavaNCSS

As you are programming your system, it is important to regularly calculate metrics that characterize the quality of your implementation. JavaNCSS is a software tool that can automatically scan your Java source code and report information about the number of non-commented source code statements and the cyclomatic complexity. You can learn more about JavaNCSS by visiting `http://www.kclee.de/clemens/java/javancss/`. After reading the Web site for JavaNCSS, you should search for papers in the ACM Digital Library, available at `http://dl.acm.org/`, that formally describe the meaning of the metrics calculated by JavaNCSS, such as cyclomatic complexity. As one of the deliverables for this assignment, your team should use LaTeX to write a document that contains formal definitions and equations that explain all of the metrics calculated by JavaNCSS.

Once all of the members of your team understand the metrics calculated by JavaNCSS, you should download and install this tool. As you did with JDepend, now you must add rules to each project's build system that can run JavaNCSS on the source code. What does the output of JavaNCSS tell you about the quality of each project's implementation? Do your chosen programs exhibit good implementation characteristics? How can you use the values of these metrics to better understand and modify the implementation of your Java programs? How much of your system's source code is for the program itself? How much of the source code is devoted to the test suite?

Analyzing Real-World Programs

After your team finishes the analysis of the two projects that you completed in previous laboratory assignments, you should identify one or more open-source Java projects that you can download. To ensure that it is easier to analyze a project with both JDepend and JavaNCSS, you should specifically look for ones that contain Ant build systems. After picking the projects that you want to further analyze, you can create separate directories for them in your Bitbucket repository for this laboratory assignment. Now, modify the build system of your project(s) so that you can use JDepend and JavaNCSS to perform an automated analysis of the design and implementation, respectively. What interesting trends can you find in the design and implementation quality metrics for your chosen open-source program(s)? How do the values of these metrics for the open-source programs compare to the values for the projects that you implemented in this class? What can you learn about ways to improve the design and implementation of your Java programs?

Presentation of Important Insights

Once you have completed all of the previous phases of this assignment, you should prepare a short five to ten minute presentation that your team will give at the start of the next laboratory session. Using the HTML-based presentation format that we used in previous laboratory assignments, you should prepare slides that highlight some of the most important insights that you learned from the automated analysis of the design and implementation of your chosen Java programs.

Summary of the Required Deliverables

This assignment invites your team to submit one printed version of the following files:

1. A description of and justification for your team's chosen organization, roles, and tool support
2. A document that clearly explains the meaning of JDepend's design quality metrics
3. An analysis of the values of design quality metrics for three previously implemented systems
4. A document that clearly explains the meaning of the metrics calculated by JavaNCSS
5. An analysis of the values of the implementation metrics for the three systems that you chose
6. Modified implementation artifacts of your chosen systems (e.g., build system and source code)
7. The slides of the presentation that you will give at the start of the next laboratory session

You must also ensure that the instructor has read access to your Bitbucket repository that is named according to the convention `cs290F2013-lab6-team`*k*, with *k* representing the number of your assigned team. Your repository should contain all of the deliverables that you produced during the completion of this assignment. Please see the instructor if you have any questions.