

Interpret the metrics offered by the static analyzer

Assignment start date: 26th of February 2015, 17:00

Assignment due date: 12th of March 2015, 00:00

Project format: PDF

Expected length: 2-4 pages

Pass criteria: You have to score minimum 1 point to be able to attend the exam. You can score maximum 25 points.

Problem description:

To improve performance, maintainability and readability of a program, static analysis proves to be an invaluable tool. It can tell us about which areas of the code to rewrite, eliminate, decouple, etc, so that the overall performance of the code improves.

Your assignment

Step 1:

Choose from your personal archive a Java, CSharp, C++, C, HTML or Visual Basic project that you have previously developed. If you don't have any code that belongs to you, you can take an open-source project from the internet as well (i.e: <http://sourceforge.net/>), but you have to state this fact and also state the source.

Upload the project on a public repository. For example: create a free Github account (<https://github.com/>), create a public repository and upload your project there.

Write in your assignment the address of your public repository.

Step 2:

Download SourceMonitor from here: <http://www.campwoodsw.com/sourcemonitor.html>. Run and install it. Please note it is only available for Windows machines.

Create in SourceMonitor a new project:

- Select the source code language (depending on which zipfile you have chosen)
- Give the SourceMonitor project a name
- Select the project that you want to analyze from its storing location
- Tick "Allow parsing of UTF-8 files"
- Finish creating the new SourceMonitor project

Step 3 (max. 5p):

Run the static analyzer on the code.

In SourceMonitor main window, select the checkpoint name, go to "View" menu and choose "Display **checkpoint** metrics Kiviat graph". Take a screenshot of your Kiviat graph. Write the following code analysis observations and interpretation:

- What does the graph tell you? How do you interpret the metrics applied on your project?

Now, do the same with a significant file from your project (select it, go to "View" menu and choose "Display **file** metrics Kiviat graph"). You can do the same for more than one file, if you have more than one file you

would like to analyze. Take a screenshot of your Kiviat graph. Write the following code analysis observations and interpretation:

- What does the graph tell you? How do you interpret the metrics applied on your file? How are they different the metrics you obtained on the whole project, compared with the metrics on this file?

Step 4 (max. 5p):

In SourceMonitor main window, double-click the checkpoint name. Take a screenshot of the result.

- Which is the biggest file you have in your project? How long the file? How many methods in it?

Step 5 (max. 5p):

In SourceMonitor main window, select your project (checkpoint name), go to “View” menu item and choose “Display method metrics”. Take a screenshot of the result.

- Which is the most complex method in your project? How many statements does it have?
- Would you refactor any of the methods you have in your project?
- What can you say about how coupled/decoupled the program is?

Step 6 (max 10p):

How would you improve your code, based on the metrics you have obtained with this analyzer?

Note:

The purpose of this assignment is to get you familiarized with checking different code metrics, interpret them depending on the project being analyzed and asses what exactly needs to be done to improve it.

Considerations:

Before interpreting the results, it may be good to search on the internet information about Kiviat graphs, metrics and what do they tell you.