

Week 3 Deliverable

Written Report on ‘FindBugs’

1. ***Who (company or individual) developed the tool? What is the cost to the tool user? How do you acquire it?***

FindBugs is a free software developed by David H. Hovemeyer and William W. Pugh and distributed under the terms of the Lesser GNU Public License.

The name FindBugs and the related logo are trademarked by The University of Maryland.

2. ***What testing purpose does the tool serve? (i.e., what problem does it attempt to solve? How does it improve productivity?)***

FindBugs is a program that uses static analysis to look for bugs and different levels of potential problems in Java code.

By using static analysis, FindBugs gives the developer a way to inspect the code without executing the program. The tool inspects Java byte code, saved as class files, to detect occurrences of bug patterns, which are checklist items for possible problems in the Java source.

The specific patterns used by FindBugs are defined in the tool, and users can select which of those patterns they will want to use to analyze the source. The patterns are grouped in different categories. Some of these categories are malicious code vulnerability, security, dodgy, performance, correctness, and bad practice. Every category is composed by patterns, and the developer can select the ones to use during the analysis.

3. ***What programming language(s) does the tool support, if any?***

Unfortunately, FindBugs only supports Java.

4. ***In what phase of software testing is the tool useful?***

FindBugs can be used to find errors and inconsistencies in the early coding stages and at any time before running the application for the first time to better eliminate the possibility of major issues and/or oversights in the lexical structure of the code. Obviously, it can also be used after running the software to help identify any inconsistent or unclear problem with the program, but the major return of using FindBugs is the elimination of bugs and inconsistencies, and good practice tips given by it before the software runs for the first time and eliminating problems that could result in a crash.

5. *What do you need to do in order to use the tool?*

If the user is seeking a stand-alone installation of the tool, then he/she should ensure to have the following technical requirements:

- At least 512 MB of memory available, and even more memory available in case the project to be analyzed is very large;
- A runtime environment compatible with Java 2 Standard Edition, version 1.5 or later.

FindBugs is platform independent, and it will run on GNU/Linux, Windows, and MacOS X.

a. How do you install it?

For a standalone installation, Ant will have to be installed and running properly.

To do so, the user can follow the instructions at <http://ant.apache.org/>.

Once Ant is working properly, the user can download a binary distribution version of FindBugs from <http://findbugs.sourceforge.net/downloads.html>.

Once you have downloaded a binary distribution, extract it into a directory of your choice and follow the step-by-step instructions in the index.html file in the 'findbug-home-directory'/doc/manual directory within the just extracted binary distribution file.

The easiest way, and possibly the better way, to install FindBugs when using Eclipse as the IDE would be to install the related plug-in from the Eclipse Marketplace (under Help), and then using this tool directly on the project to be tested.

b. How do you configure it?

For the standalone installation, the user needs to launch FindBugs from the 'findbug-home-directory'/lib and double-clicking on the findbugs.jar file.

Some basic configurations can be done using the GUI toolbar.

If the user plans on using the command line/terminal to launch and control FindBugs, the path to the bin folder within the FindBugs's directory has to be added to the PATH environment variable.

If, instead, the user plans on using the Eclipse plug-in version, the detector configuration is recommended to include all the patterns that FindBugs has provided except all the hidden ones; the reporter configuration can be set to report all the different priorities (High, Medium, and Low) FindBugs provides along with all the bugs categories.

In either case, if a more advanced configuration is needed, the best way to accomplish so would be by consulting the manual mentioned in a previous question.

c. *How do you use it?*

For the standalone installation, the user needs to launch FindBugs from the 'findbug-home-directory'/lib and double-clicking on the findbugs.jar file. The project that needs to be analyzed will then have to be exported in a .jar format to be loaded as a class archive in the FindBugs GUI when creating a new project (File → New Project).

To use the FindBugs Eclipse plug-in the .jar export file will not be required, and the tool can be used directly on the wanted project by right clicking on it and running the 'Find Bugs' option. The results are shown in the 'Problems' panel.

Either way, FindBugs will provide a list of bugs found in the project divided per category, and the result of the analysis can be saved for later review.

6. *What are the strengths and limitations of the tool?*

A. *Strengths:*

1. It can be installed as a very easy and useful plug-in to a series of different Java IDEs that are widely used by all major Java programmers;
2. It's relatively light, and it can be set up to update by itself;
3. It has the option to run as either a standalone software with a GUI or as a plug-in adapting to user's preferences;
4. It has the capability of adding libraries and/or custom code for future implementations and specific code testing (instructions on how to do so can be found in the later chapters of the manual);
5. Easy to use;
6. Provides good suggestions when examining a bug in Eclipse;
7. It provides a description of why a bug-class can be harmful (from high to low relevance/importance).

B. *Limitations:*

1. It only works with one language limiting the use to only a section of the software development activities going on today;
2. It does not provide any useful indications on how to fix common bugs that are commonly detected.