**Tool analysis**

Paper title: Using Static Analysis to Find Bugs

DOI: [10.1109/MS.2008.130](https://ieeexplore.ieee.org/document/4602670)

Team members: Buiga Andreea-Diana, Ceontea Vlad-Călin, Chirea Liviu

**Foreword**

The tool mentioned in the paper, FindBugs, is abandonware, and for the uses of this document, we will use a fork called SpotBugs, equal in features.

**Installation**

The primary distributions of the tool are Eclipse and IntelliJ plugins. We used the IntelliJ version of the tool. The installation was very easy, as it only involves downloading and enabling the plugin in the IDE. Then we can start to use the program from the bottom toolbar.

**Tool Usage**

The tool offers a variety of usage options, including analyzing a selection of files, analyzing classes, or performing a total analysis of the codebase in the current project. We are then presented with a segmented interface, split into:

A list of all “bug items”, as they are called by the program, which can be sorted and grouped based on several criteria, such as category, class, package or severity

An interactive preview of the code where the selected bug item has been found

Analysis of the selected bug item, including description, priority, classification and recommended solutions

**Tool Findings**

After using FindBugs on multiple projects, we found several categories of bugs:

- Bad Practice, such as string equality using ‘==’

- Dodgy code, such as storing data in a variable and then not using it

- Internationalization, such as reliance on default encoding, which causes behaviour

- Correctness, such as a null check on a previously dereferenced entity

- Malicious Code Vulnerability, such as when a mutable object has a public property that references an internal class

- Performance, such as initializing fields that go on to be unused