Comparison of Static Code Analysis Tools for JAVA

Frank Lin, Prateek Tandon

Static Code Analysis

- The analysis of code that is performed without the execution of the code.
  - Advantages:
    - Find errors in code at exact location
    - Find errors earlier in development
    - Allows for quicker turn around for fixes
  - Disadvantages:
    - Tools are only as good as established rules
    - Does not find errors related to the runtime environment (Dynamic Code Analysis)

- Compared 4 popular open-source tools
  - FindBugs, Checkstyle, PMD, SonarGraph

Tools

- **FindBugs**
  - Detects instances of code that are likely to be errors (bug patterns)
  - Checks: Code vulnerability, performance, thread synchronization, etc.
- **Checkstyle**
  - Detects code that deviate from a defined set of coding rules
  - Checks: Code layout, design problems, reusability, etc.
- **PMD**
  - Detects code styles that are suspicious and can potentially cause errors
  - Checks: Dead code, over complicated expressions, suboptimal code, etc.
- **SonarGraph**
  - Detects errors in code structure
  - Checks: Quality of structure, cyclic dependencies, efficiency, etc.

Method

- Analyzed JAVA code using the 4 tools
  - Compared different errors detected, structures, running time, ease of use, etc.
  - Used most general version of tools and extended versions

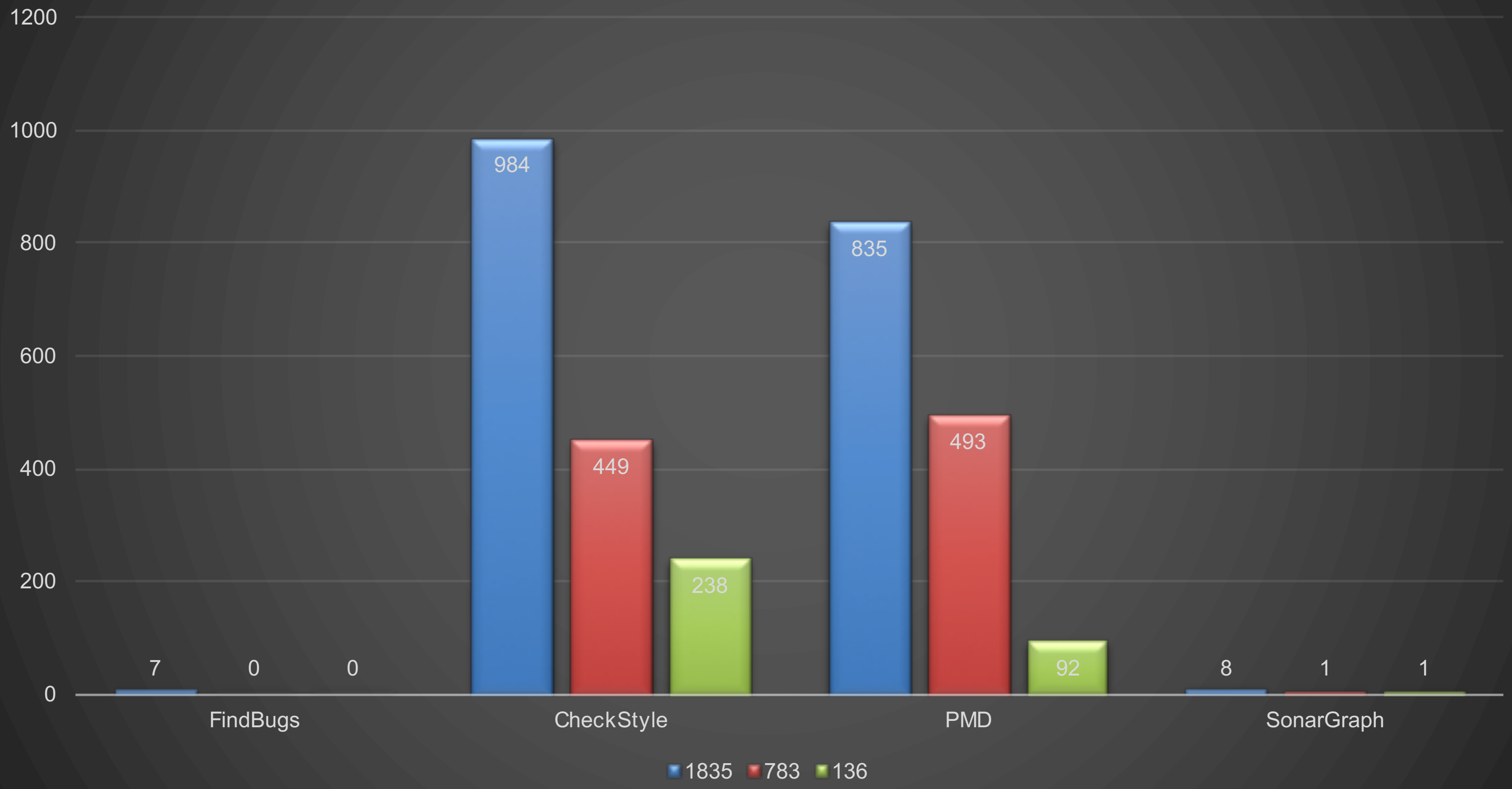
- Eclipse
  - All tools have plugins for eclipse

- Code Tested
  - Very Small (~150 lines)
    - Subtle bugs (null values, mathematical inconsistencies, strings)
  - Small (~800 lines)
    - Algorithms, object equality
  - Medium (~2k lines)
    - GUI, Streams, duplication

Results

	FindBugs	CheckStyle	PMD	SonarGraph
Null pointer dereferences	Yes	No	Yes	Yes
Class/Method/Variable nature	No	Yes	Yes	No
Duplicated code	No	Kind of	Yes	Yes
Blank lines and whitespace	No	Yes	No	No
Data Flow	Only recently	No	No	Yes
Optimization possibility	Yes	No	Yes	Yes
Number of Rules	414	132	234	>500
Requirement	Compiled Code	Uncompiled code	Uncompiled code	Compiled Code
Ease of Use (1-10)	9	7	8	4
Loops, indices, reachability	Yes	No	Yes	Yes
Extra Return statement	No	No	Yes	No
Naming Conventions	No	Yes	Yes	No

Lines of Code vs Number of warnings/errors



Conclusion

- No single best static code analyzer
- Best in terms of usage and results: **Findbugs** (Totally open source)
    - Also incorporating data flow analysis
- Best in terms of in-depth analysis: **SonarGraph** (Open source for medium projects)
    - Huge amount of configurable metrics
- Choice for your project:
    - Simplicity and Speed vs Depth
- Hard to analyze areas:

```
if(condition)
        if(condition)
else
        body
```