Automatic Code Inspection

Software Engineering

Chair of Programming Methodology

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

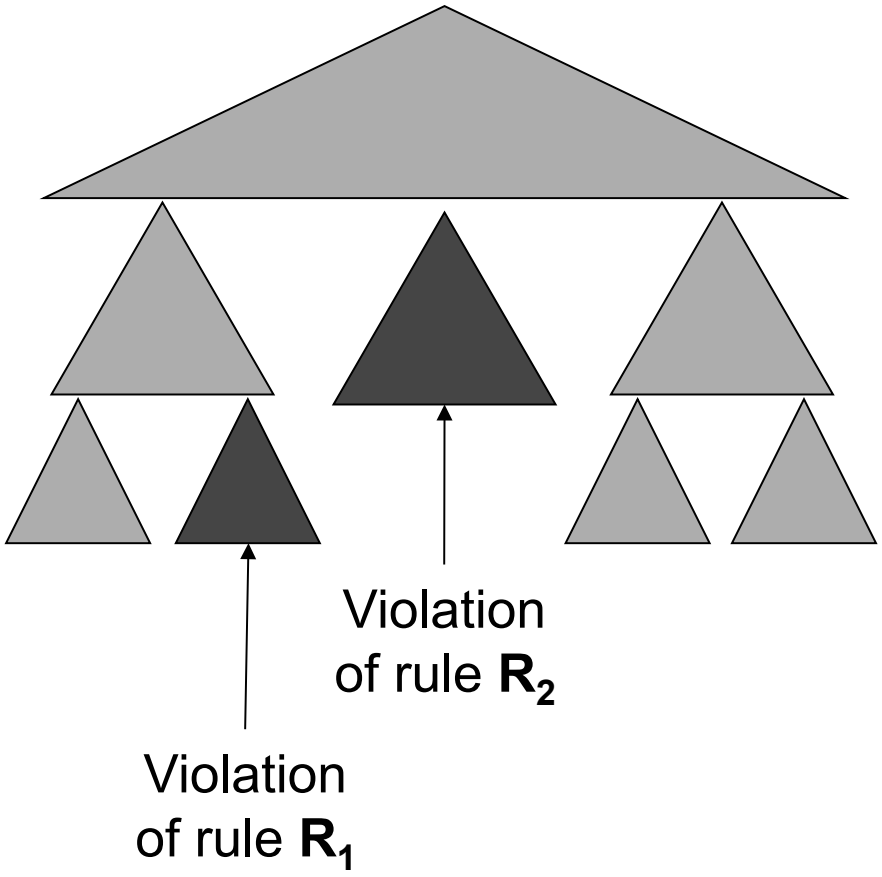
Automation of code reviewing

- Possible bugs
- Code convention violations
- Dead code
- Duplicated code
- Suboptimal code

According to a report released by The Standish Group automated code inspection reduced the number of people needed for manual code reviews by 50%.

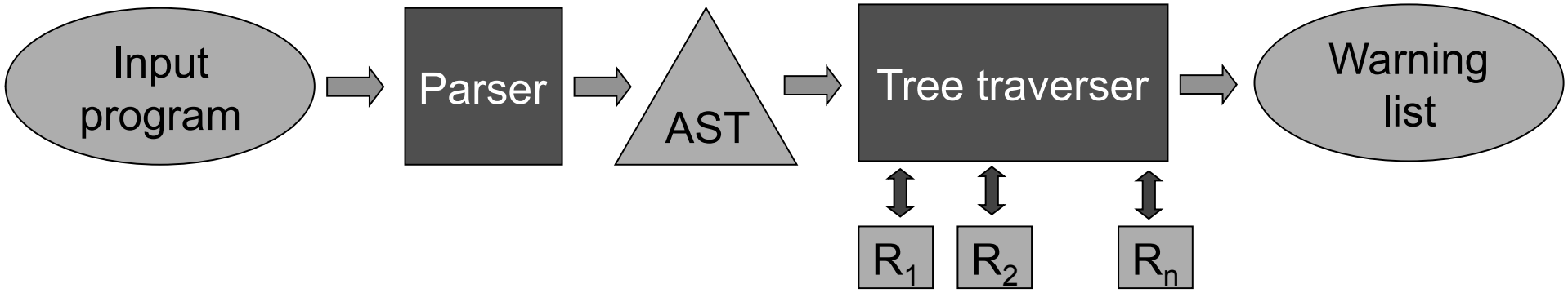
Main concept

- Syntactical matching of rules violation



Rule – is description of set of ASTs that can be reason for the one of the mentioned above problems.

How it works



- For expressing some rules representation of AST should contain whitespaces, tabulations, EOLs, comments
- Rules represented via Java code or XML XQuery

Automatic code inspection versus IDE

- Eclipse
  - Code style
  - Potential programming problems
  - Name shadowing and conflicts
  - Unnecessary code

- Advantages of the automatic code inspection
  - Bigger set of rules
  - User rules

Existing tools

Checkstyle 4.1 - Checkstyle

http://checkstyle.sourceforge.net/

- PMD

http://pmd.sourceforge.net/

- FindBugs

http://findbugs.sourceforge.net/

- JCSC (Java Coding Standard Checker)

http://jcsc.sourceforge.net/

Common features

- Freeware
- 100 – 200 of standard rules
- Configuration via XML
- Sometimes support creation of users rules via Java or XQuery
- Usually support integration in different IDE
- Usually implemented in Java

Why PMD?

- Support user rules creation via both Java code and XML XQuery
- Standard rules cover wide range of rule types
- Stable version
- Support plug-ins for various IDE

What is PMD?

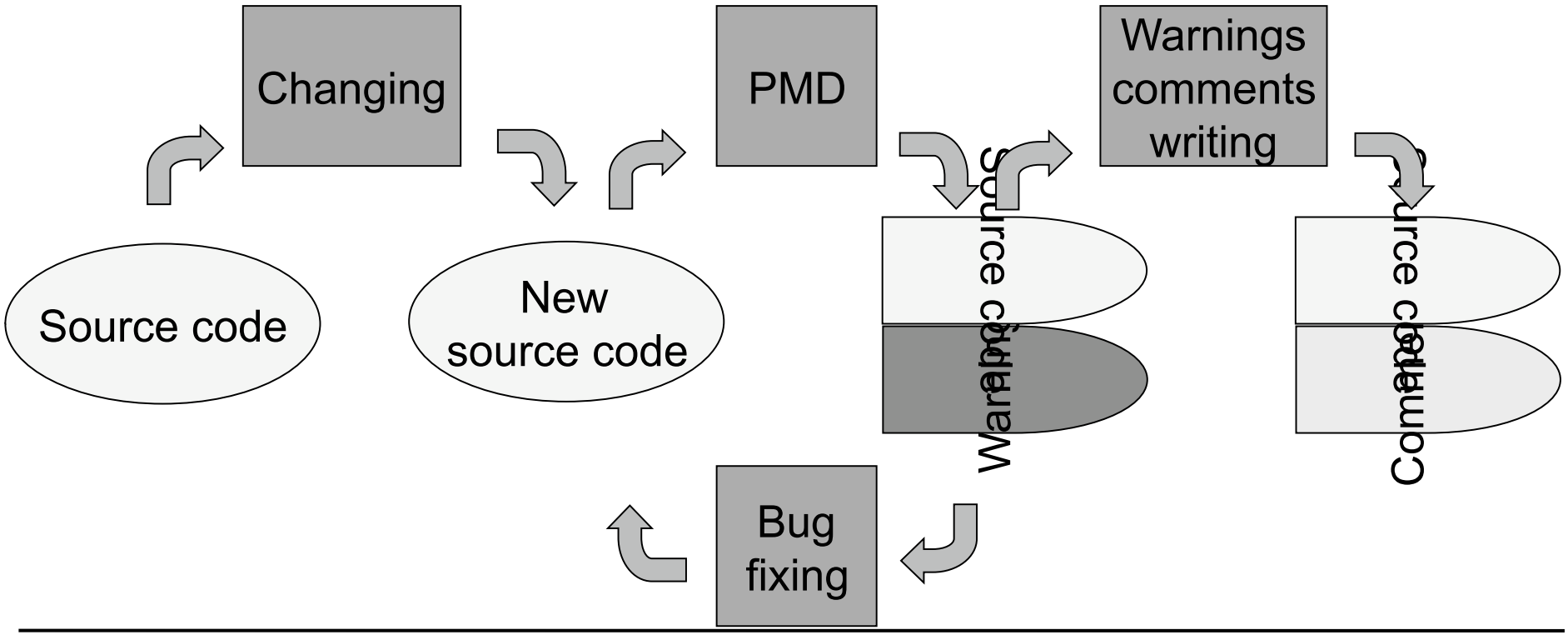
- Last version: 4.2.5
- License: BSD License
- Language: Java
- Rules: around 150
- Rule sets: around 30
- User rules creation via Java and XQuery
- Originally developed to improve the Cougaar (**Cognitive Agent Architecture**) project (DARPA initiative) http://cougaar.org/projects/cougaar-pmd/

What does ‘PMD’ mean?

- Pretty Much Done
- Project Mess Detector
- Project Monitoring Directives
- Project Meets Deadline
- Programming Mistake Detector
- Pounds Mistakes Dead
- PMD Meaning Discovery (recursion, hooray!)
- Programs of Mass Destruction
- A 'Chaotic Metal' rock band name
  “Pretty Marry Dies”

Application of the PMD

- Select standard rules
- Write project specific rules
- Select rules parameters



Pros and Cons

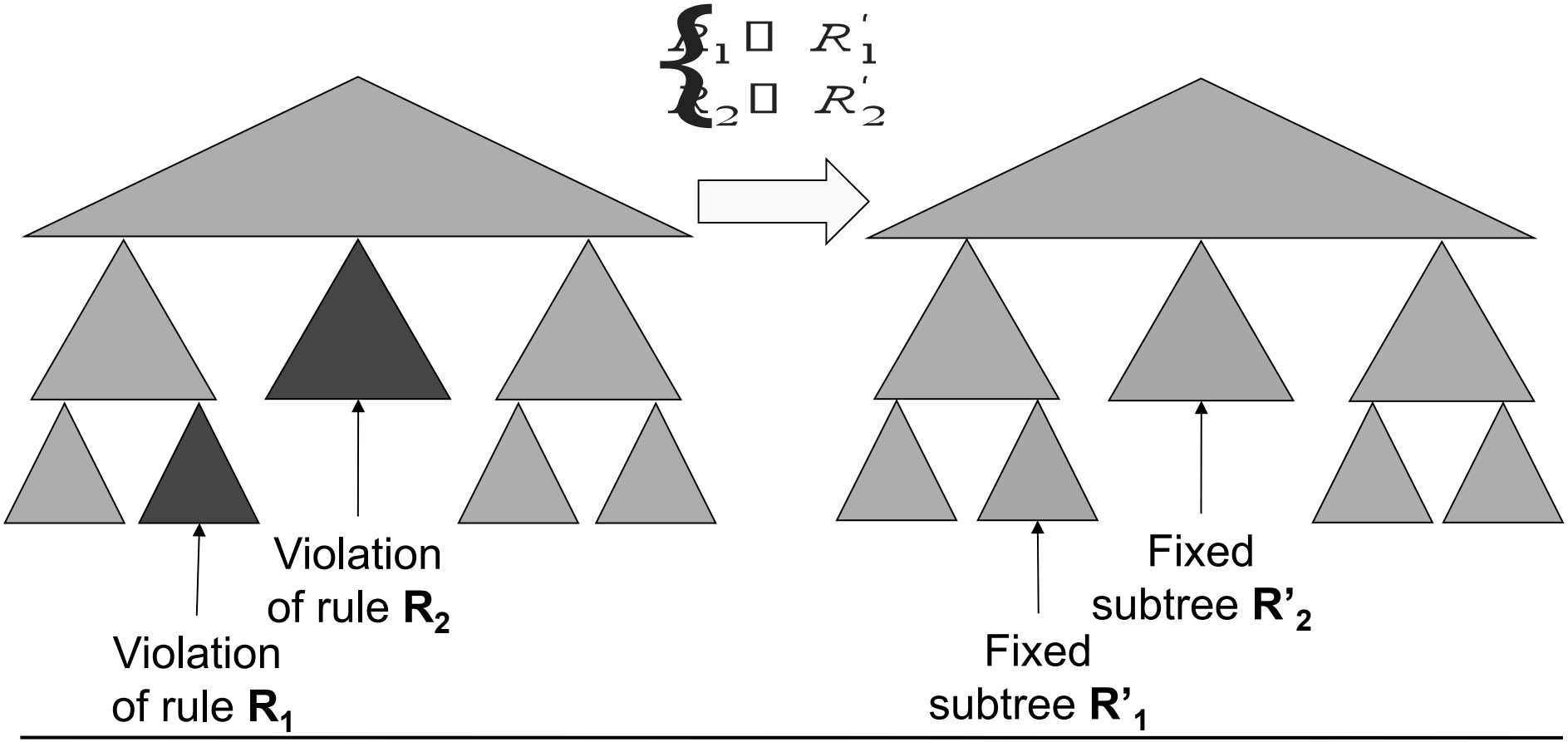
- Pros
  - Don’t require any additional efforts.
  - Users don’t need have any specific knowledge. It’s enough that user understand notion of the AST.

- Cons
  - Not sound. It’s possible that many from the found warnings are not real errors.
  - Not complete. It can miss many real errors.

Future development: automatic bug fixing

- Program transformation via rewriting rules. For example StrategoXT.

  http://www.program-transformation.org/Stratego/JavaFront



Future development: integration with a prover

```
requires i > 5;
int f(int i){
    if (i < 2) //UnconditionalIfStatement
      return i +1;
    else
      return i -1;
}
```

Future development: empirics for filtering warnings

- Goal:
  - Decrease warnings number
  - Remove false warnings

- Problems:
  - Fuzzy warning criteria
  - Context depending warnings