Code improvement: refactoring and static code analysis

UA/DETI/TQS
Ilídio Oliveira (ico@ua.pt)
v2019-03-19.

universidade de aveiro
departamento de eletrónica,
telecomunicações e informática
deti

Leaning objectives

Give example of bad coding practices (“bed smells”)

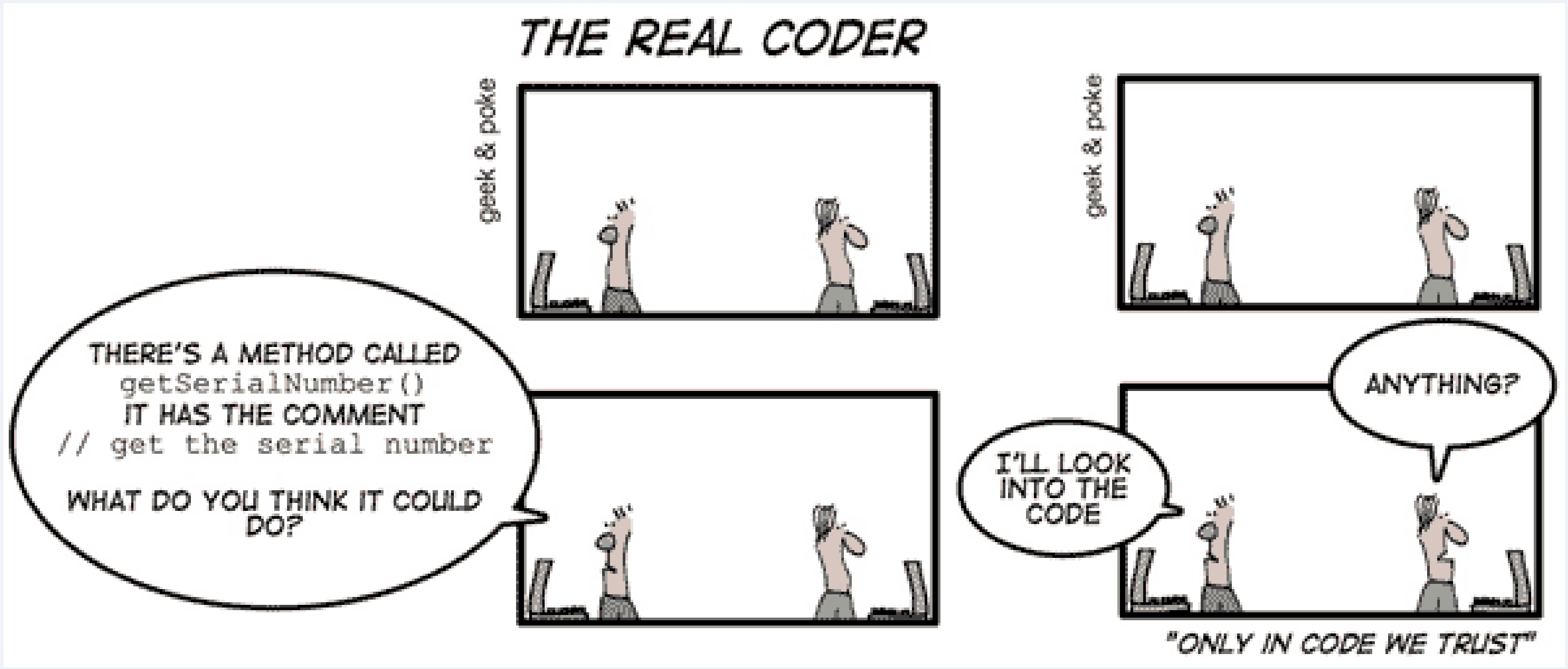
Identify the occurrence of “bed smells” in code

Propose refactoring options for given “bed smells”

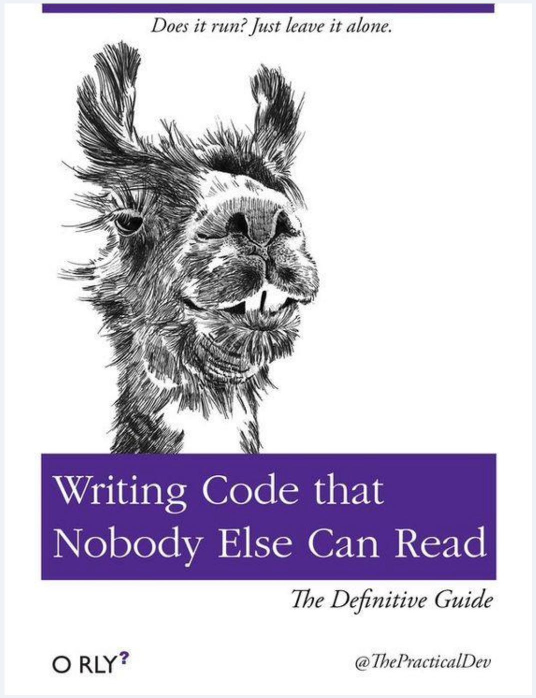
Explain the role of Inspectors (static code analysis)

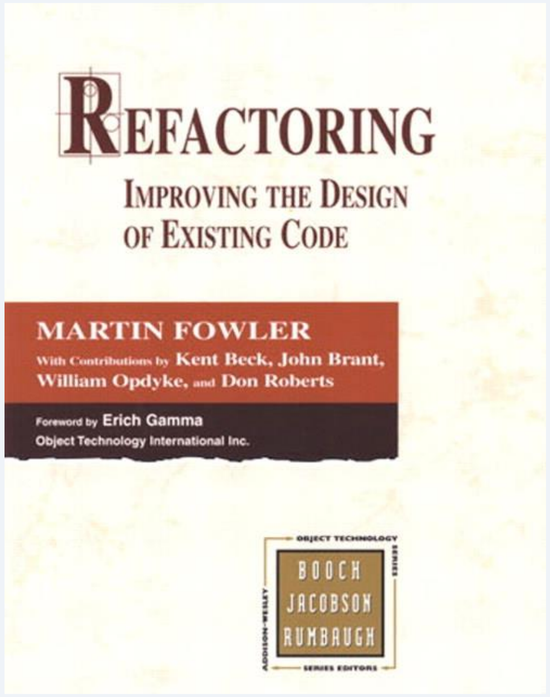
Describe the metrics used in SonarQube

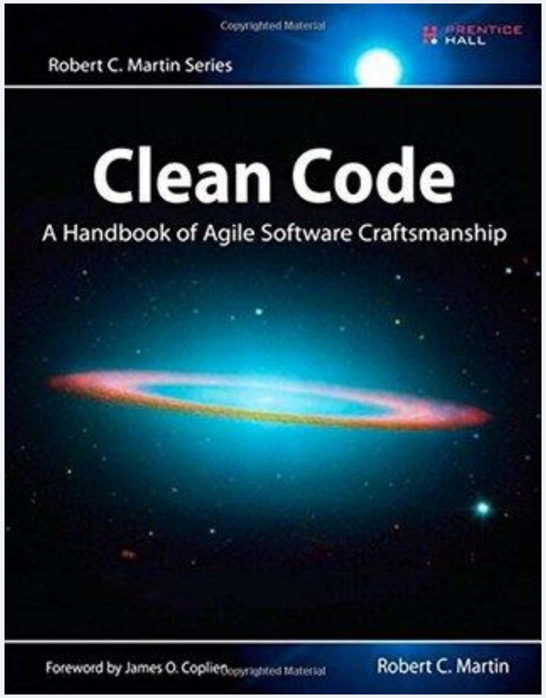
Define the concept of technical debt and explain how it should be managed in a SQEnvironment


THE REAL CODER
geek & poke
THERE'S A METHOD CALLED getSerialNumber() IT HAS THE COMMENT // get the serial number WHAT DO YOU THINK IT COULD DO?
geek & poke
I'LL LOOK INTO THE CODE
ANYTHING?
"ONLY IN CODE WE TRUST"

Not all code is equally easy to maintain


Does it run? Just leave it alone.
Writing Code that Nobody Else Can Read
The Definitive Guide
O RLY?
@ThePracticalDev


REFACTORING
IMPROVING THE DESIGN OF EXISTING CODE
MARTIN FOWLER
With Contributions by Kent Beck, John Brant, William Opdyke, and Don Roberts
Foreword by Erich Gamma
Object Technology International Inc.
OBJECT TECHNOLOGY SERIES
BOOCH JACOBSON RUMBAUGH
SERIES EDITORS
ADDISON-WESLEY



Find the intruder…

Code refactoring

- **Refactoring** is a controlled technique for improving the design of an existing code base
- ...altering its internal structure without changing its external behavior.
- Key aspects:
  - series of "small" transformations
  - preserving functionality & correctness.

Examples

- Extract (duplicate code into a) method
- Extract interface

See also:

- Catalog of refactoring situations (M. Fowler)

- Still useful, given that many IDEs will automate (selected) refactoring situations?...

Why refactoring?

- **Cleaner code → easier to understand and maintain**
- Better design for the current understanding of the architecture
- Reduce complexity → easier to understand and evolve
- Make the code more reusable (for other or more general needs)
- Improve performance
- Improve security (by removing vulnerabilities)

When to refactor

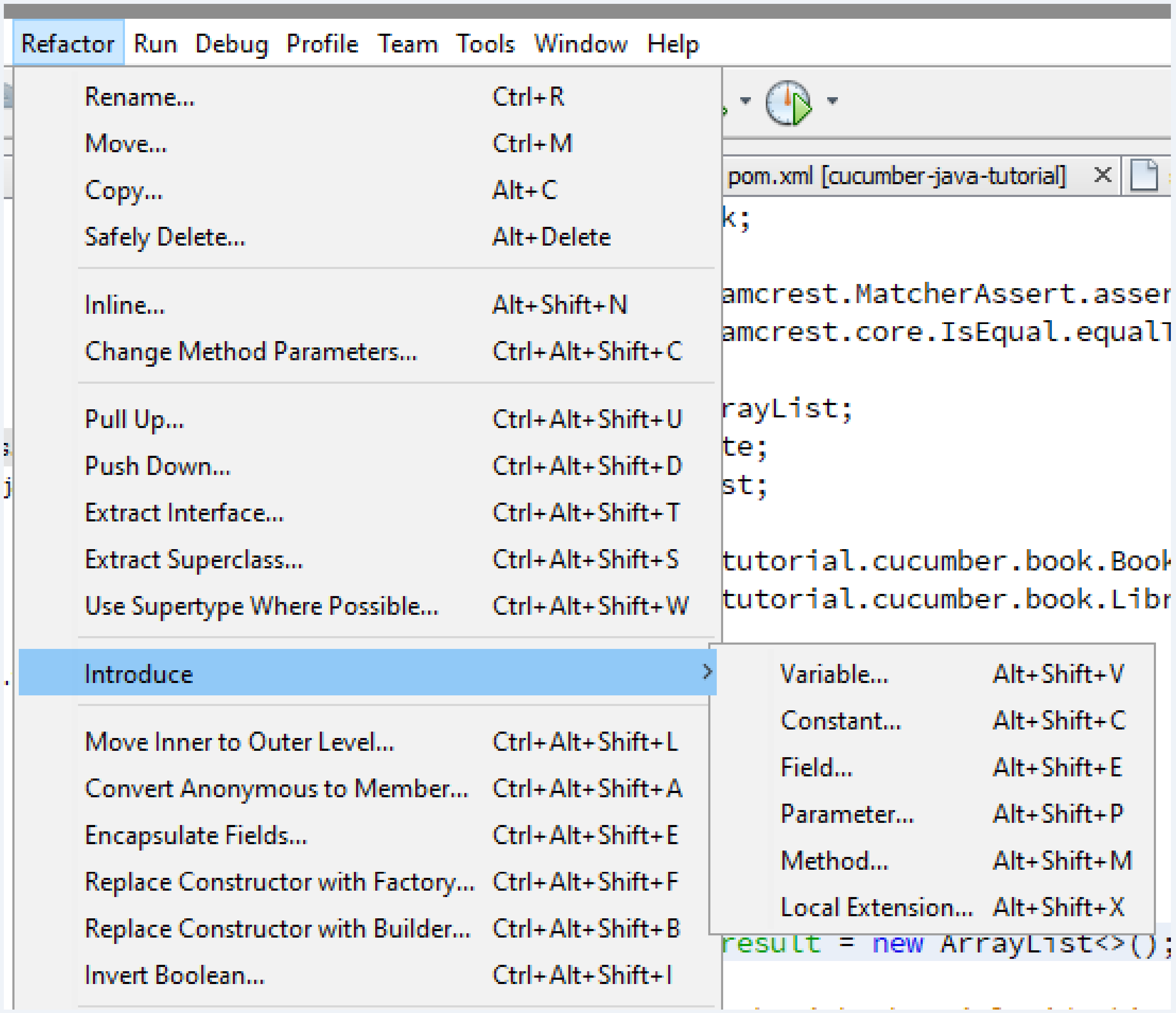
Resolve “code smells” (anti-patterns)

- See: catalog of [bad code smells]

Examples:

- Duplicate code → Extract method
- Long method → Extract method
- Data class → Encapsulate field
- Feature Envy → Move method

NetBeans support



Refactor Run Debug Profile Team Tools Window Help

Rename...	Ctrl+R
Move...	Ctrl+M
Copy...	Alt+C
Safely Delete...	Alt+Delete
Inline...	Alt+Shift+N
Change Method Parameters...	Ctrl+Alt+Shift+C
Pull Up...	Ctrl+Alt+Shift+U
Push Down...	Ctrl+Alt+Shift+D
Extract Interface...	Ctrl+Alt+Shift+T
Extract Superclass...	Ctrl+Alt+Shift+S
Use Supertype Where Possible...	Ctrl+Alt+Shift+W
Introduce	›
Move Inner to Outer Level...	Ctrl+Alt+Shift+L
Convert Anonymous to Member...	Ctrl+Alt+Shift+A
Encapsulate Fields...	Ctrl+Alt+Shift+E
Replace Constructor with Factory...	Ctrl+Alt+Shift+F
Replace Constructor with Builder...	Ctrl+Alt+Shift+B
Invert Boolean...	Ctrl+Alt+Shift+I

Variable...	Alt+Shift+V
Constant...	Alt+Shift+C
Field...	Alt+Shift+E
Parameter...	Alt+Shift+P
Method...	Alt+Shift+M
Local Extension...	Alt+Shift+X

pom.xml [cucumber-java-tutorial]

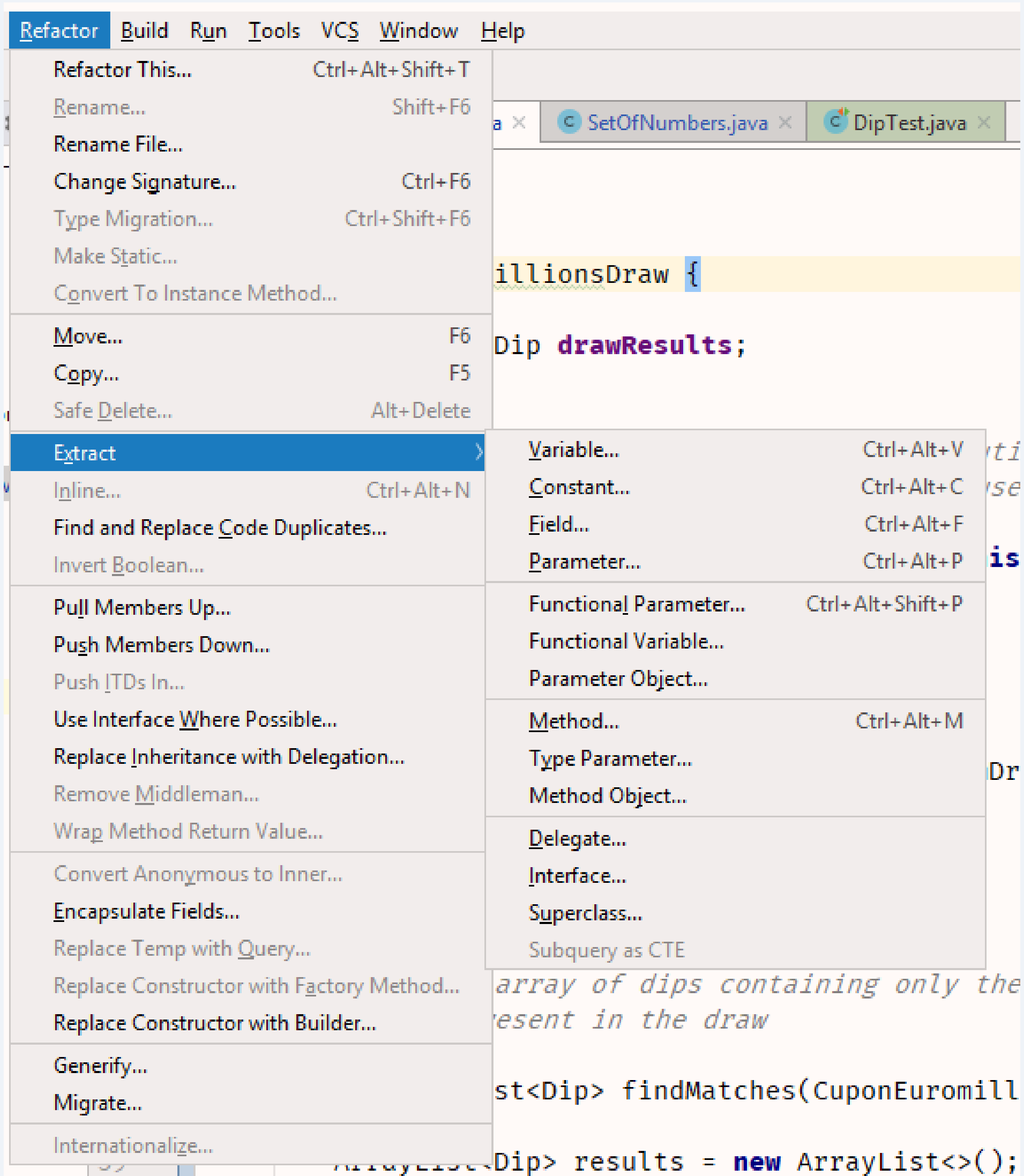
```
k;
amcrest.MatcherAssert.asser
amcrest.core.IsEqual.equalT

rayList;
te;
st;

tutorial.cucumber.book.Book
tutorial.cucumber.book.Libr

result = new ArrayList<>();
```

IntelliJ support



```
illionsDraw {

Dip drawResults;

array of dips containing only the
esent in the draw

st<Dip> findMatches(CuponEuromill
Dip> results = new ArrayList<>();
```

Static Code Analysis

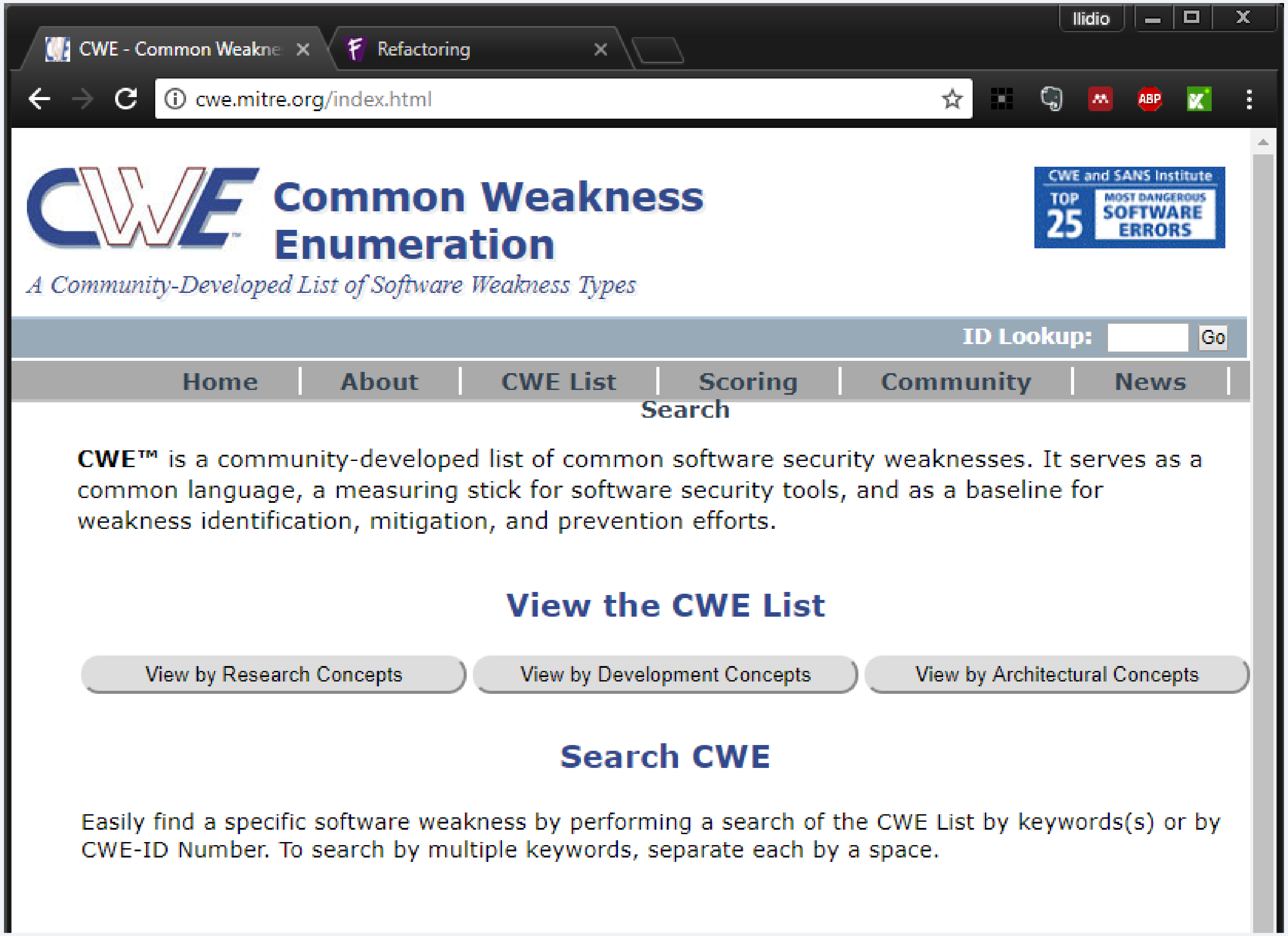
Code inspection

Analysis of code patterns, without running the code

Examples of issues found in SA:

- Referencing a variable with an undefined value
- Variables that are never used
- Unreachable (dead) code
- Programming standards violations
- Security vulnerabilities
- Internationalization (i18n) issues

Catalogs of code weaknesses (setting the vocabulary)


CWE - Common Weakne
Refactoring
Ilidio
cwe.mitre.org/index.html
CWE
Common Weakness Enumeration
A Community-Developed List of Software Weakness Types
CWE and SANS Institute TOP 25 MOST DANGEROUS SOFTWARE ERRORS
ID Lookup:
Go
Home
About
CWE List
Scoring
Community
News
Search
View by Research Concepts
View by Development Concepts
View by Architectural Concepts

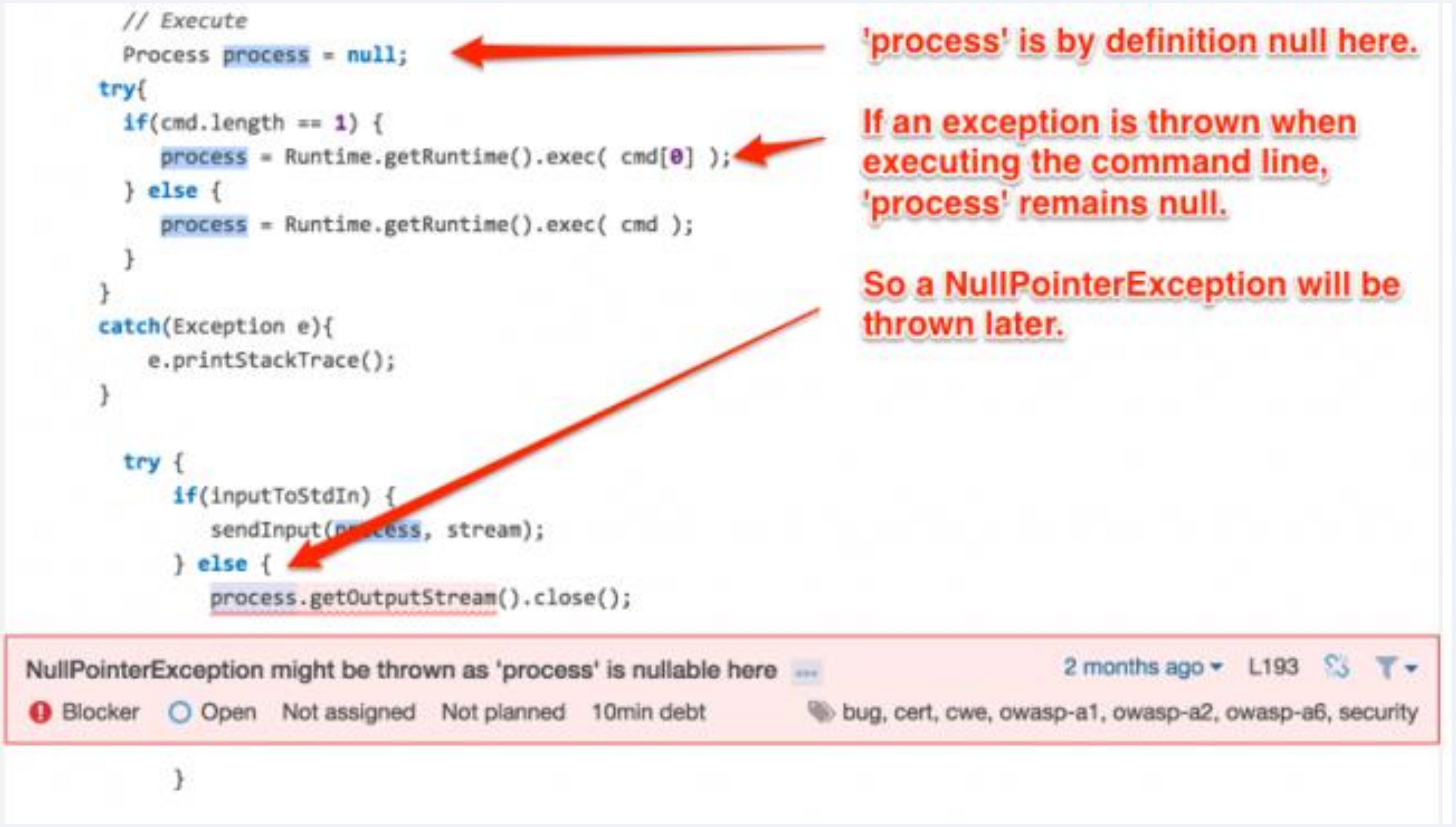
CWE™ is a community-developed list of common software security weaknesses. It serves as a common language, a measuring stick for software security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.

View the CWE List

Search CWE

Easily find a specific software weakness by performing a search of the CWE List by keywords(s) or by CWE-ID Number. To search by multiple keywords, separate each by a space.

NPE due to a badly handled exception



```
  // Execute
  Process process = null;
try{
  if(cmd.length == 1) {
     process = Runtime.getRuntime().exec( cmd[0] );
  } else {
     process = Runtime.getRuntime().exec( cmd );
  }
}
catch(Exception e){
   e.printStackTrace();
}

  try {
      if(inputToStdIn) {
         sendInput(process, stream);
      } else {
         process.getOutputStream().close();
```

NullPointerException might be thrown as 'process' is nullable here ... 2 months ago ▾ L193

Blocker ○ Open Not assigned Not planned 10min debt bug, cert, cwe, owasp-a1, owasp-a2, owasp-a6, security

```
      }
```

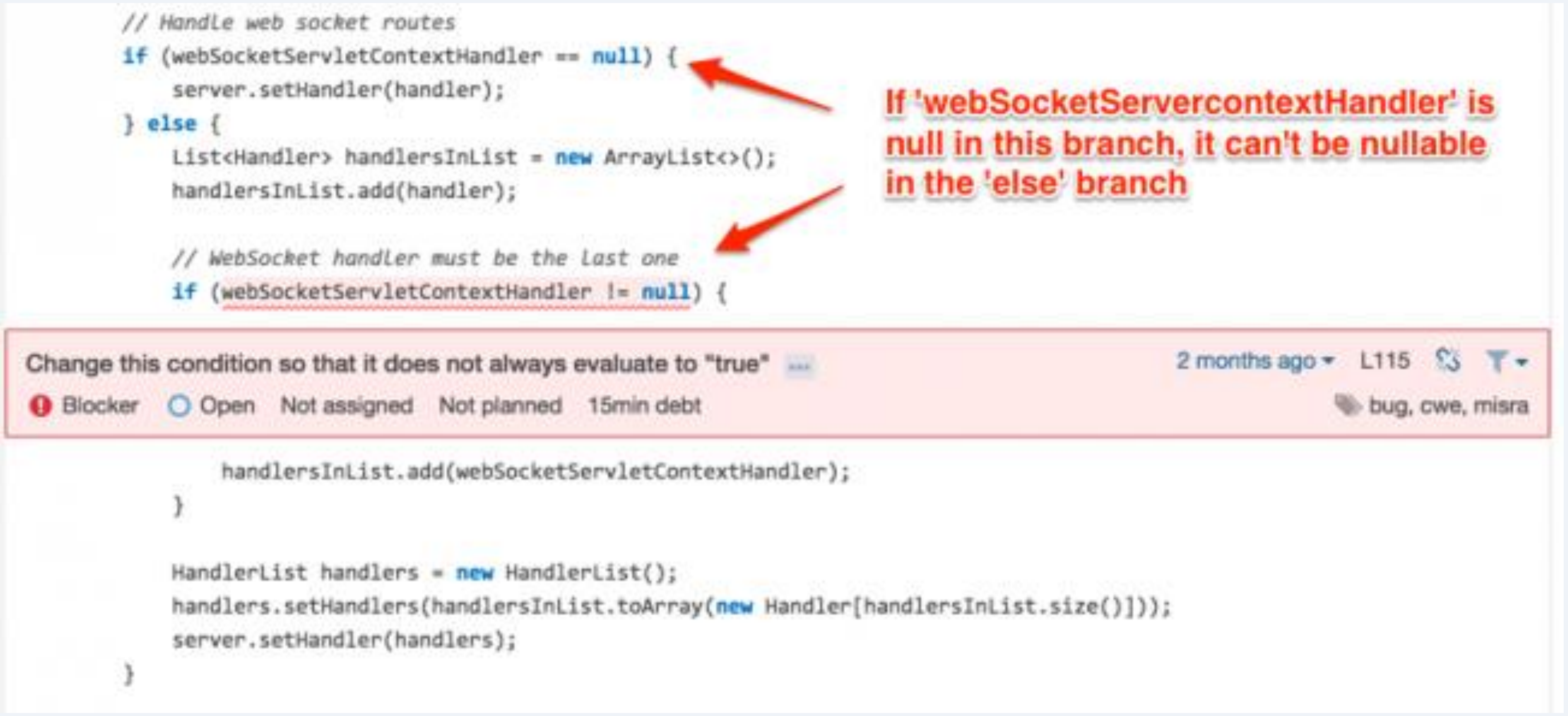
'process' is by definition null here.

If an exception is thrown when executing the command line, 'process' remains null.

So a NullPointerException will be thrown later.

https://blog.sonarsource.com/sonaranalyzer-for-java-tricky-bugs-are-running-scared/

Useless condition



```
// Handle web socket routes
if (webSocketServletContextHandler == null) {
    server.setHandler(handler);
} else {
    List<Handler> handlersInList = new ArrayList<>();
    handlersInList.add(handler);

    // WebSocket handler must be the last one
    if (webSocketServletContextHandler != null) {
```

If 'webSocketServercontextHandler' is null in this branch, it can't be nullable in the 'else' branch

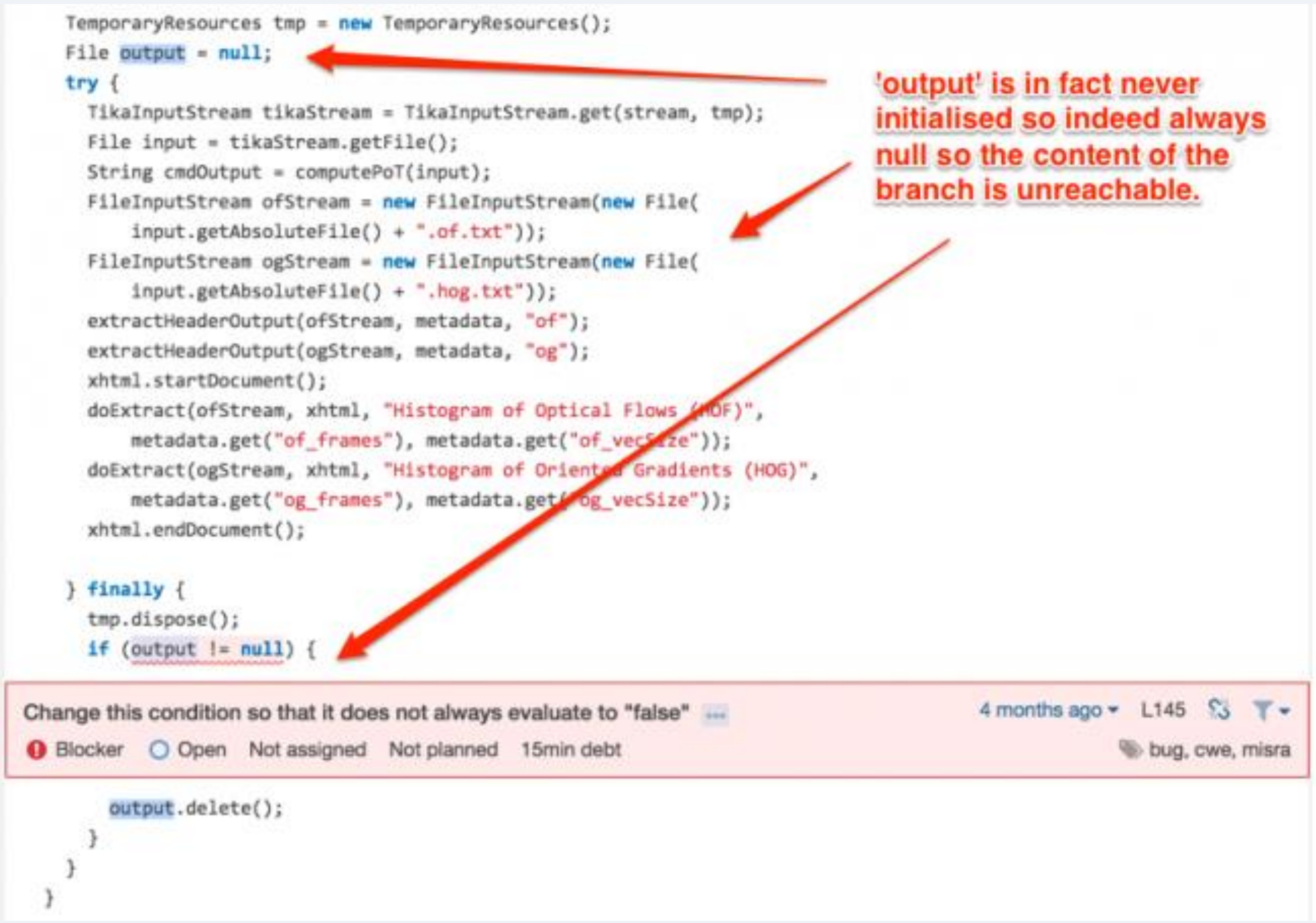
Change this condition so that it does not always evaluate to "true" ...
Blocker Open Not assigned Not planned 15min debt
2 months ago L115
bug, cwe, misra

```
        handlersInList.add(webSocketServletContextHandler);
    }

    HandlerList handlers = new HandlerList();
    handlers.setHandlers(handlersInList.toArray(new Handler[handlersInList.size()]));
    server.setHandler(handlers);
}
```

https://blog.sonarsource.com/sonaranalyzer-for-java-tricky-bugs-are-running-scared/

Suspect unreachable branch



```
    TemporaryResources tmp = new TemporaryResources();
    File output = null;
    try {
      TikaInputStream tikaStream = TikaInputStream.get(stream, tmp);
      File input = tikaStream.getFile();
      String cmdOutput = computePoT(input);
      FileInputStream ofStream = new FileInputStream(new File(
          input.getAbsoluteFile() + ".of.txt"));
      FileInputStream ogStream = new FileInputStream(new File(
          input.getAbsoluteFile() + ".hog.txt"));
      extractHeaderOutput(ofStream, metadata, "of");
      extractHeaderOutput(ogStream, metadata, "og");
      xhtml.startDocument();
      doExtract(ofStream, xhtml, "Histogram of Optical Flows (HOF)",
          metadata.get("of_frames"), metadata.get("of_vecSize"));
      doExtract(ogStream, xhtml, "Histogram of Oriented Gradients (HOG)",
          metadata.get("og_frames"), metadata.get("og_vecSize"));
      xhtml.endDocument();

    } finally {
      tmp.dispose();
      if (output != null) {
```

Change this condition so that it does not always evaluate to "false" ...

Blocker   Open   Not assigned   Not planned   15min debt

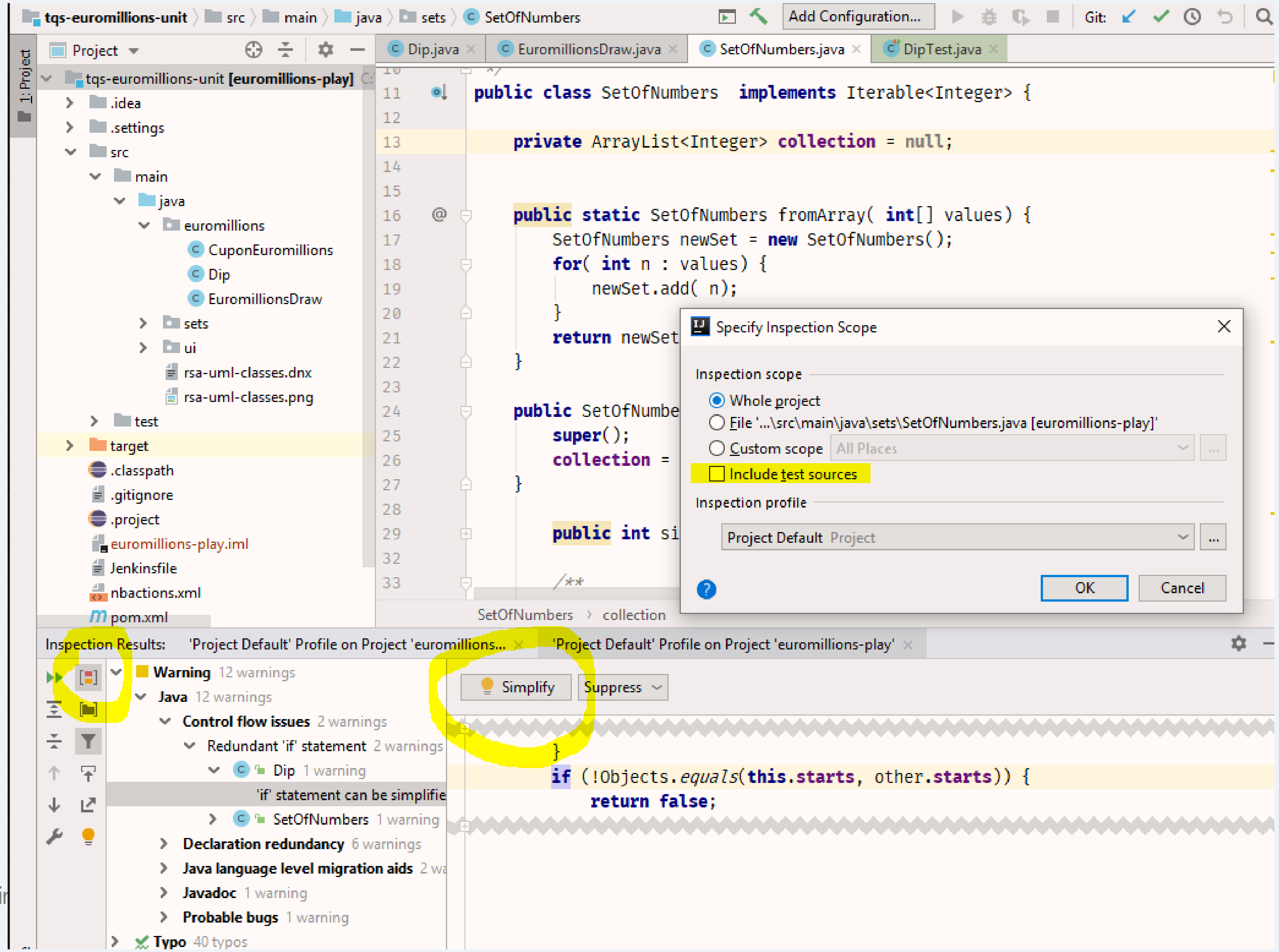
4 months ago   L145

bug, cwe, misra

```
        output.delete();
      }
    }
  }
```

'output' is in fact never initialised so indeed always null so the content of the branch is unreachable.

Code inspection in IntelliJ



tqs-euromillions-unit › src › main › java › sets › SetOfNumbers

Dip.java | EuromillionsDraw.java | SetOfNumbers.java | DipTest.java

```
public class SetOfNumbers  implements Iterable<Integer> {

    private ArrayList<Integer> collection = null;


    public static SetOfNumbers fromArray( int[] values) {
        SetOfNumbers newSet = new SetOfNumbers();
        for( int n : values) {
            newSet.add( n);
        }
        return newSet
    }

    public SetOfNumbe
        super();
        collection =
    }

        public int si

        /**
```

Specify Inspection Scope

Inspection scope

- [✓ Whole project]
- File '...\src\main\java\sets\SetOfNumbers.java [euromillions-play]'
- Custom scope: All Places
- ☐ Include test sources

Inspection profile: Project Default Project

OK | Cancel

Inspection Results: 'Project Default' Profile on Project 'euromillions... | 'Project Default' Profile on Project 'euromillions-play'

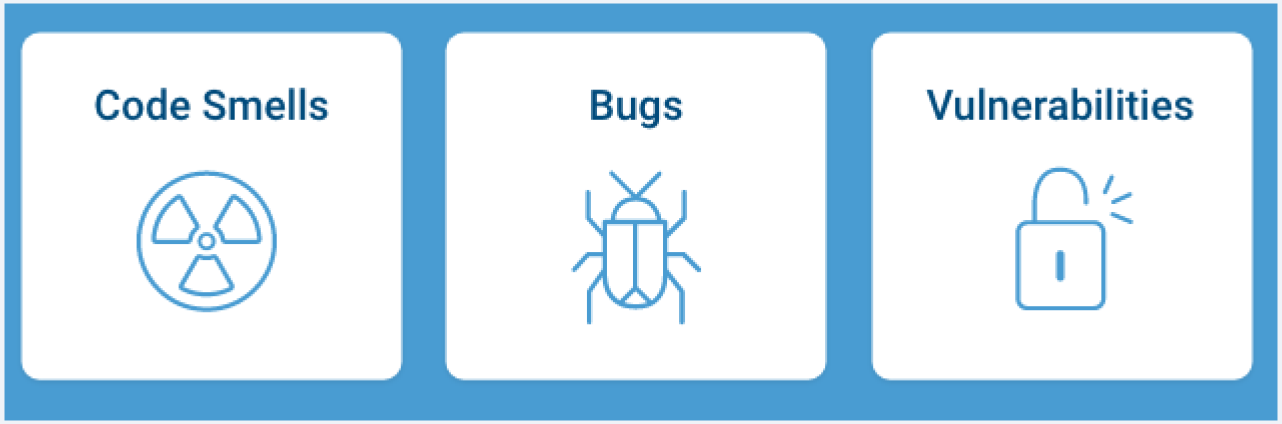
- Warning 12 warnings
  - Java 12 warnings
    - Control flow issues 2 warnings
      - Redundant 'if' statement 2 warnings
        - Dip 1 warning
          - 'if' statement can be simplifie
        - SetOfNumbers 1 warning
    - Declaration redundancy 6 warnings
    - Java language level migration aids 2 wa
    - Javadoc 1 warning
    - Probable bugs 1 warning
- Typo 40 typos

Simplify | Suppress

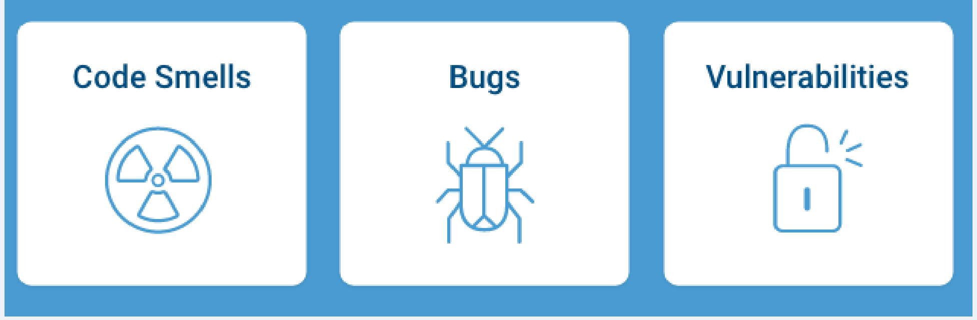
```
}
if (!Objects.equals(this.starts, other.starts)) {
    return false;
```

Advanced inspection frameworks

sonarqube



SonarQube concepts



https://docs.sonarqube.org/display/SONAR/Concepts

Code Smell

- A **maintainability-related issue** in the code. Leaving it as-is means that at best maintainers will have a harder time than they should making changes to the code. At worst, they'll be so confused by the state of the code that they'll introduce additional errors as they make changes.

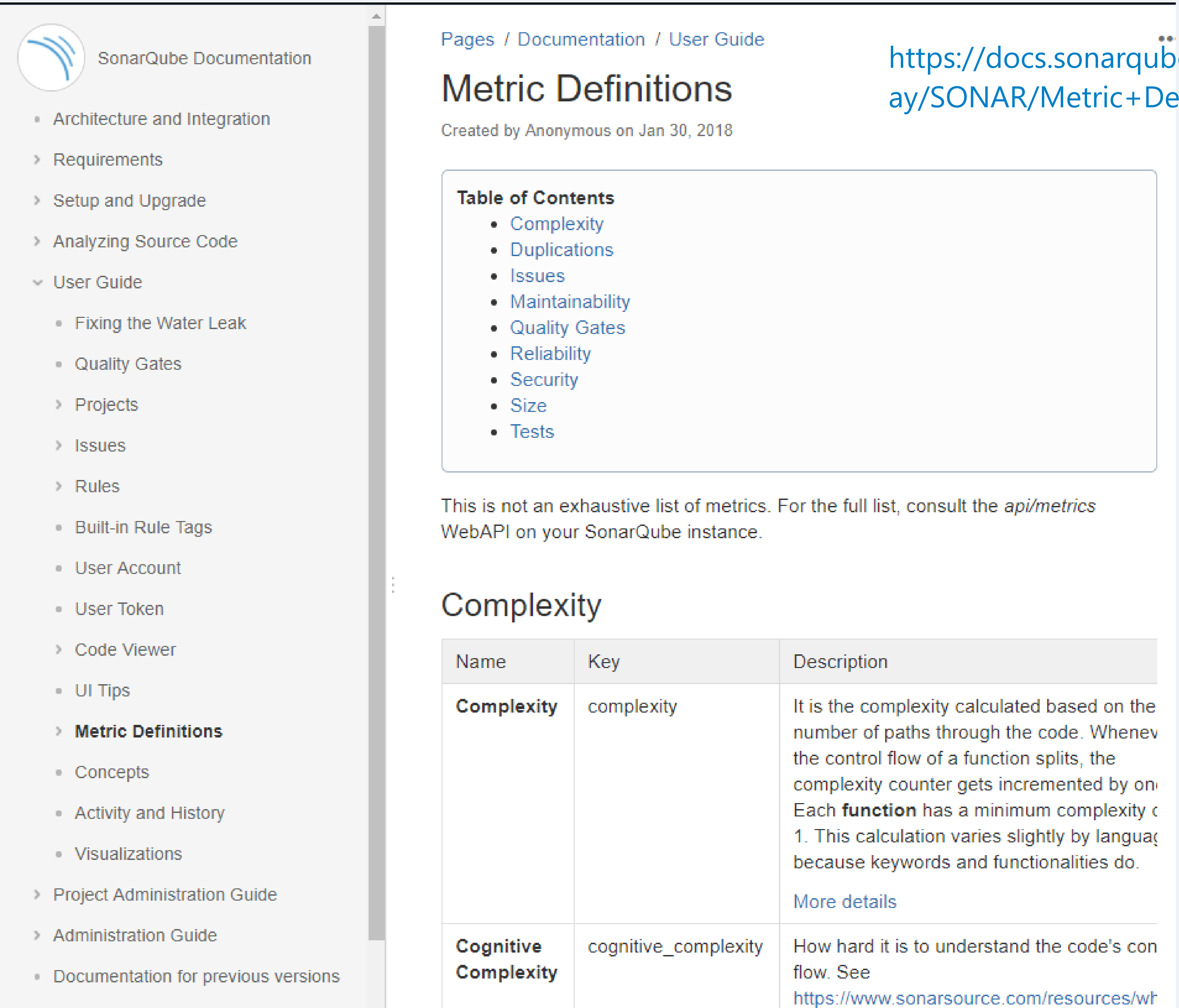
Bug

- An issue that represents something wrong in the code. If this has not broken yet, it will, and probably at the worst possible moment. This **needs to be fixed**. Yesterday.

Vulnerability

- A **security-related issue** which represents a potential backdoor for attackers.

https://docs.sonarqube.org/display/SONAR/Metric+Definitions



SonarQube Documentation

- Architecture and Integration
- Requirements
- Setup and Upgrade
- Analyzing Source Code
- User Guide
  - Fixing the Water Leak
  - Quality Gates
  - Projects
  - Issues
  - Rules
  - Built-in Rule Tags
  - User Account
  - User Token
  - Code Viewer
  - UI Tips
  - **Metric Definitions**
  - Concepts
  - Activity and History
  - Visualizations
- Project Administration Guide
- Administration Guide
- Documentation for previous versions

Pages / Documentation / User Guide

Metric Definitions

Created by Anonymous on Jan 30, 2018

Table of Contents

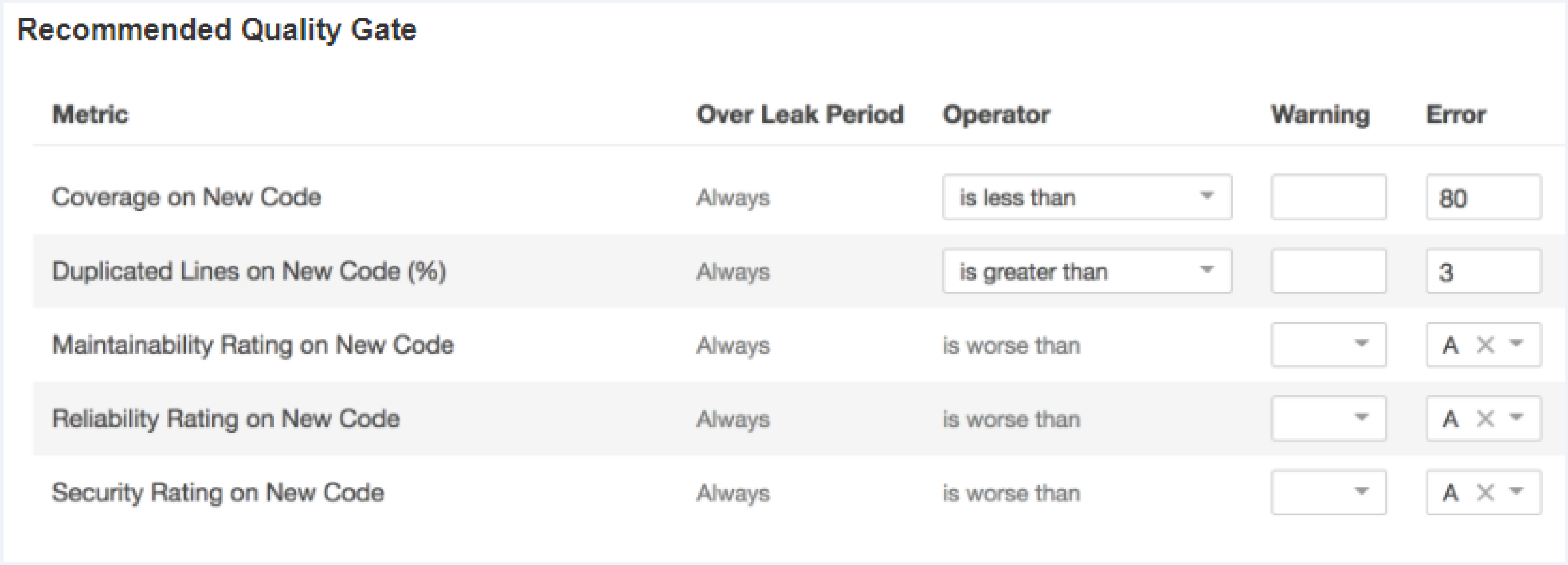
This is not an exhaustive list of metrics. For the full list, consult the *api/metrics* WebAPI on your SonarQube instance.

Complexity

Name	Key	Description
Complexity	complexity	It is the complexity calculated based on the number of paths through the code. Whenev the control flow of a function splits, the complexity counter gets incremented by on Each **function** has a minimum complexity c 1. This calculation varies slightly by languag because keywords and functionalities do. More details
Cognitive Complexity	cognitive_complexity	How hard it is to understand the code's con flow. See https://www.sonarsource.com/resources/wh

Quality gates

- Ready for delivery? Yes, if QG is met.



Recommended Quality Gate

Metric	Over Leak Period	Operator	Warning	Error
Coverage on New Code	Always	is less than		80
Duplicated Lines on New Code (%)	Always	is greater than		3
Maintainability Rating on New Code	Always	is worse than		A
Reliability Rating on New Code	Always	is worse than		A
Security Rating on New Code	Always	is worse than		A

Configurable quality gates on SonarQuebe