TRIOLOGY

simply smart solutions

Codequalitätskontrolle mit SonarQube

Wer ist das?

- Josha von Gizycki
- Goslarer
- Seit 2008 bei der TRIOLOGY GmbH
- Java, Oracle, Webentwicklung, …
- Hacker, Zocker, Rocker

Warum der?

Inspected with

sonarqube

Typische Fehler im Alltag

- Keine Tests
- Zeitdruck
- Legacy Code
- Basteln

Was ist das Ergebnis?

- Big Ball of Mud?
- Gasfabrik?
- Spaghetticode mit Copy Pasta?
- Innere Plattform?
- Sumo-Hochzeit?

Was macht statische Codeanalyse?

- Bytecode oder Quelltext
- Nutzt Metriken
- Generiert Statistiken

Was ist statische Codeanalyse?

- Bytecode
- Bugs
- Harte Fehler

- Quelltext
- Ineffizienz
- Handwerk

checkstyle

- Quelltext
- Codestil
- Konventionen

Metriken?

Lat. „ars metrica“: Lehre von den Maßen

Softwaremetriken

Mathematische Funktion → Code → Vergleichbarkeit

„Eine Softwarequalitätsmetrik ist eine Funktion, die eine Software-Einheit in einen Zahlenwert abbildet, welcher als Erfüllungsgrad einer Qualitätseigenschaft der Software-Einheit interpretierbar ist.“

– IEEE Standard 1061, 1998

McCabe – Zyklomatische Komplexität

- Anzahl unterschiedlicher Möglichkeiten, durch ein Stück Code zu laufen
- 1 + Anzahl Kontrollstrukturen + Boolesche Operatoren
  - `if, while, do, for, ?:, catch, case`
  - `&&, ||`
- Grenzwerte
  - 10 für Methoden
  - 200 für Klassen

McCabe – Zyklomatische Komplexität

```
String wochentagsName(int nummer) {
  switch(nummer) {
    case 1: return "Montag";
    case 2: return "Dienstag";
    case 3: return "Mittwoch";
    case 4: return "Donnerstag";
    case 5: return "Freitag";
    case 6: return "Samstag";
    case 7: return "Sonntag";
  }
  return "";
}
```

```
String wochentagsName(int nummer) {
  String[] tage = new String[] {
    "Montag",
    "Dienstag",
    "Mittwoch",
    "Donnerstag",
    "Freitag",
    "Samstag",
    "Sonntag"};
  int len = tage.length;
  if (nummer >= 1 && nummer <= len) {
    return tage[nummer - 1];
  }
  return "";
}
```

Mehrwert

- Quantitative / Qualitative Einschätzung der Codebasis
- Argumentationsgrundlage gegenüber Entscheidern
- Risikoabschätzung
- Teamorganisation

SonarQube

- Scanner
- Datenbank
- Web-Komponente

SonarQube

- OpenSource unter LGPL v3
- Integriert unter anderem PMD, Checkstyle, Findbugs
- Plugins
  - Sprachen (Java, Javascript, PHP, …)
  - SCM (Git, Subversion, Mercurial, …)
  - GitHub (Authentifikation, Pull Requests)

Code Smell

- „Cyclomatic Complexity“
- „Deprecated code should be removed eventually“
- „Silly math should not be performed“

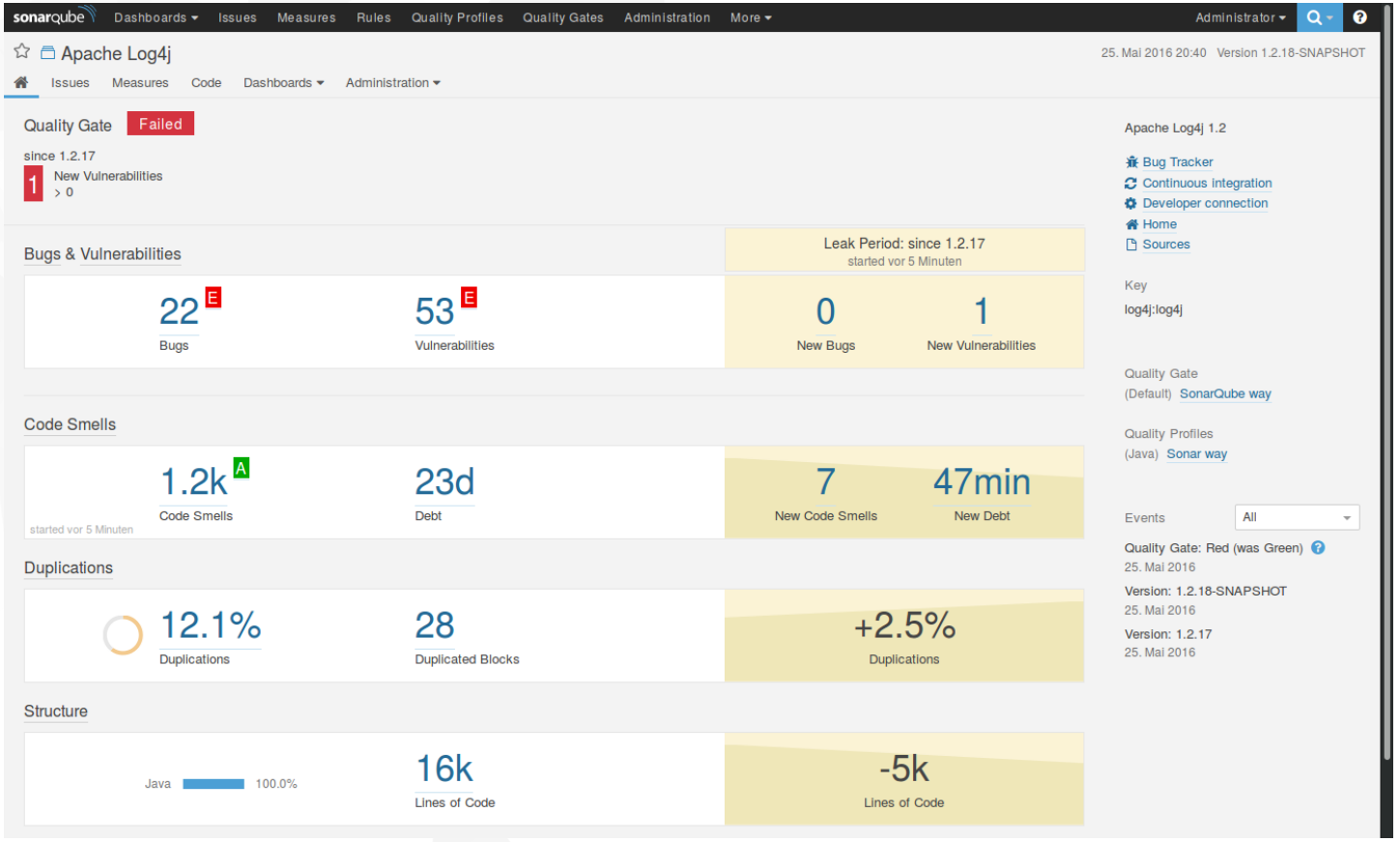
Vulnerability

- „public static fields should be constant“
- „Credentials should not be hard-coded“
- „Values passed to SQL commands should be sanitized“

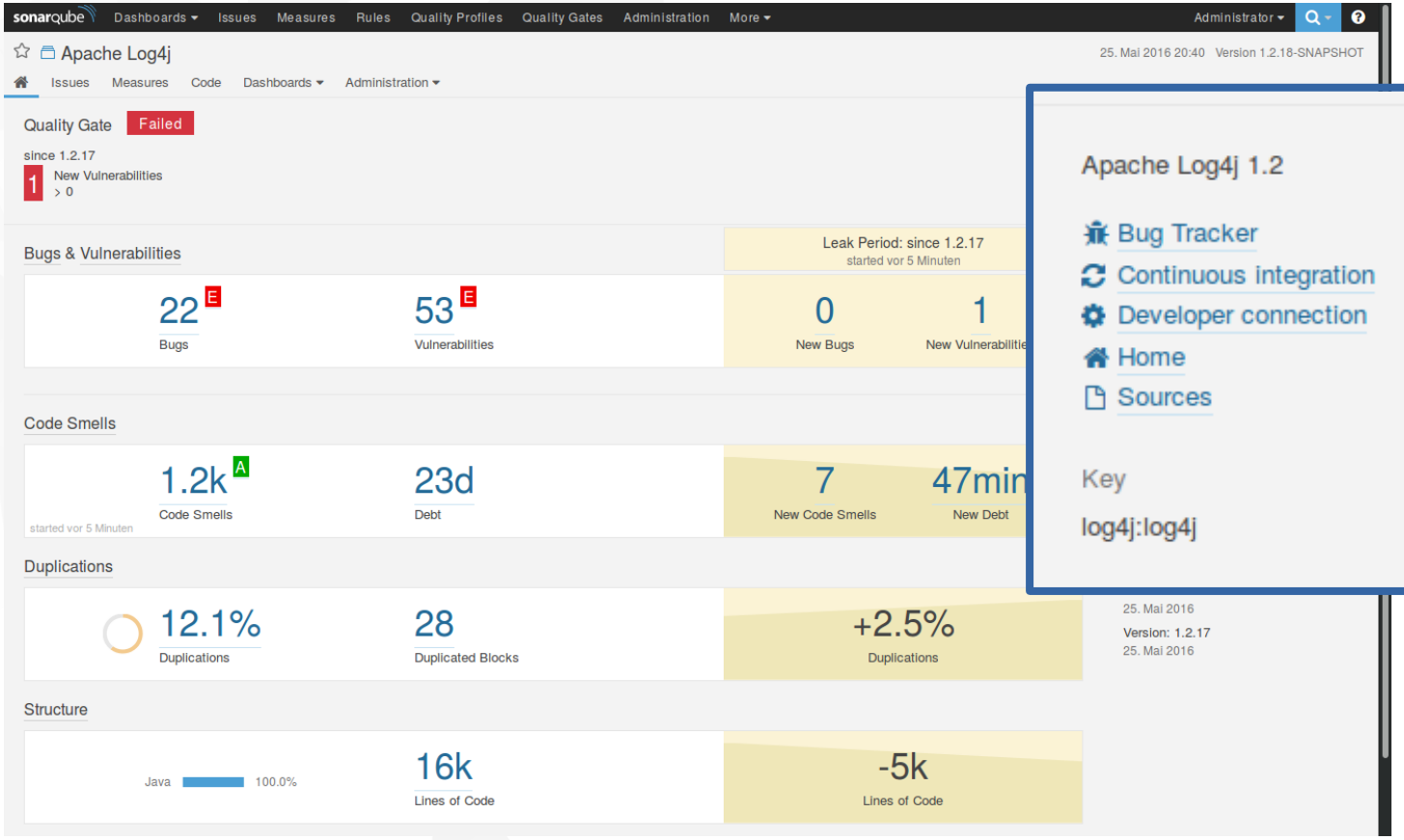
Bugs

- „Classes should not be compared by name“
- „Loops should not be infinite“
- „Null pointers should not be dereferenced“

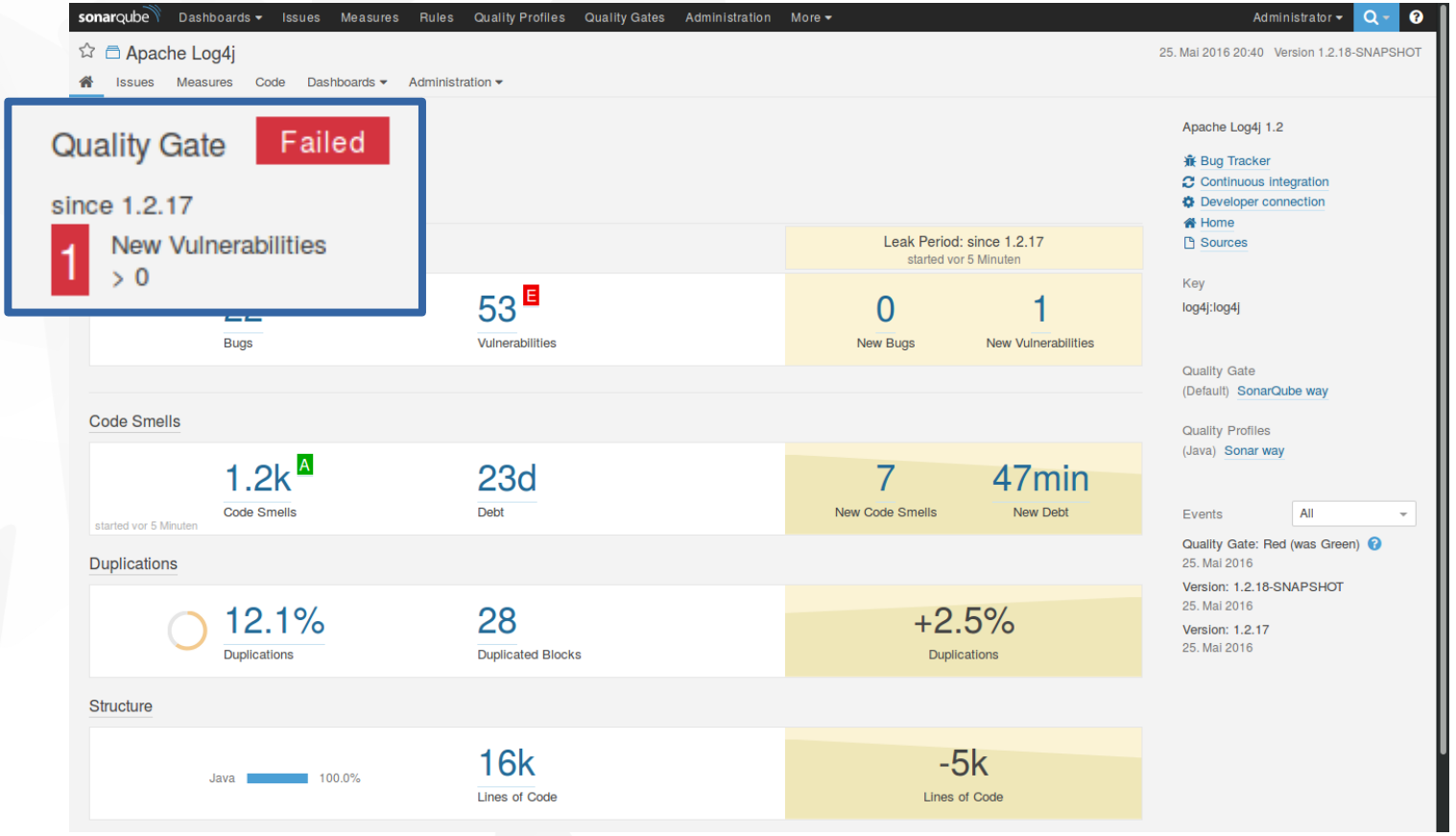
Auswertung


sonarqube
Dashboards
Issues
Measures
Rules
Quality Profiles
Quality Gates
Administration
More
Administrator
Apache Log4j
25. Mai 2016 20:40 Version 1.2.18-SNAPSHOT
Issues
Measures
Code
Dashboards
Administration
Quality Gate
Failed
since 1.2.17
1 New Vulnerabilities
> 0
Apache Log4j 1.2
Bug Tracker
Continuous integration
Developer connection
Home
Sources
Key
log4j:log4j
Bugs & Vulnerabilities
Leak Period: since 1.2.17
started vor 5 Minuten
22 E
Bugs
53 E
Vulnerabilities
0
New Bugs
1
New Vulnerabilities
Code Smells
1.2k A
Code Smells
23d
Debt
7
New Code Smells
47min
New Debt
started vor 5 Minuten
Quality Gate
(Default) SonarQube way
Quality Profiles
(Java) Sonar way
Events
All
Quality Gate: Red (was Green)
25. Mai 2016
Version: 1.2.18-SNAPSHOT
25. Mai 2016
Version: 1.2.17
25. Mai 2016
Duplications
12.1%
Duplications
28
Duplicated Blocks
+2.5%
Duplications
Structure
Java
100.0%
16k
Lines of Code
-5k
Lines of Code

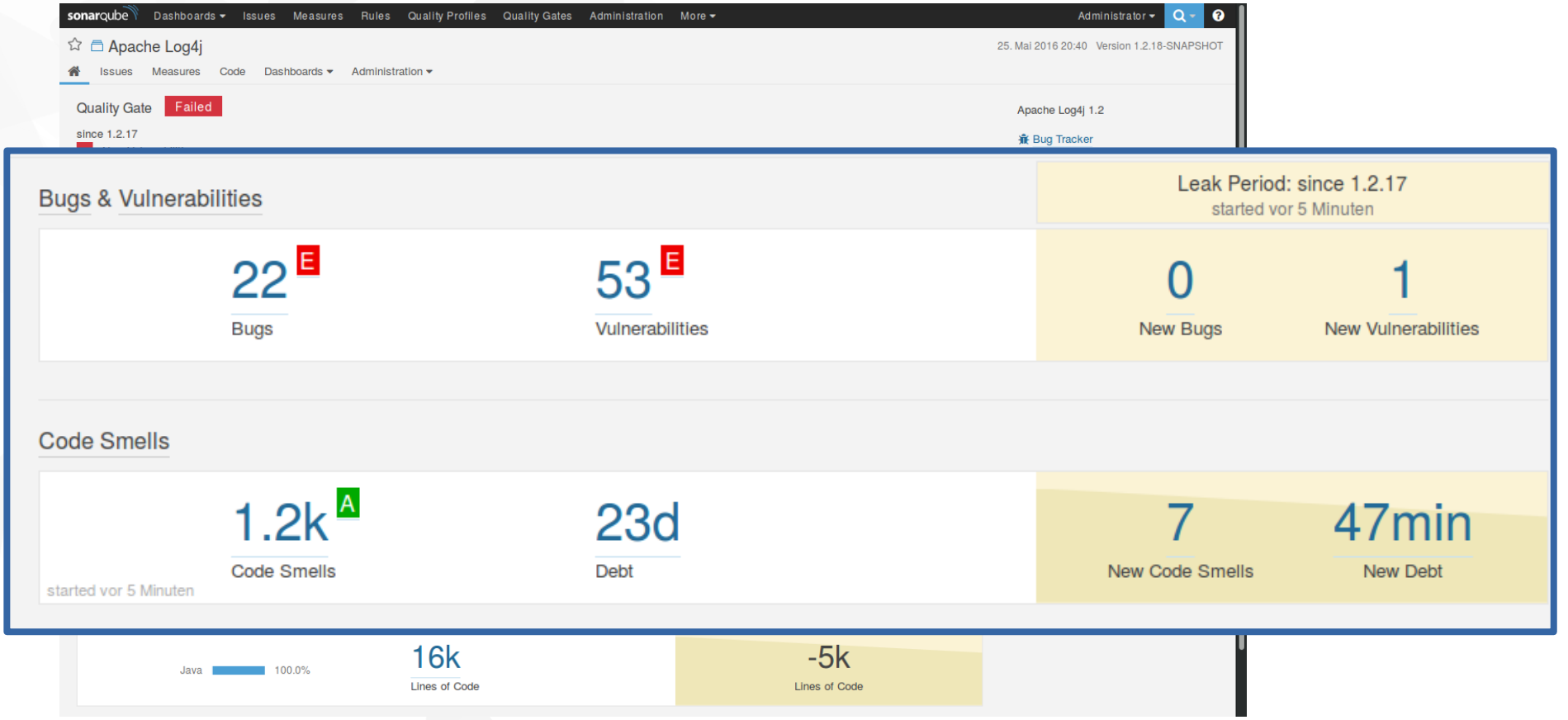
Auswertung


sonarqube
Dashboards
Issues
Measures
Rules
Quality Profiles
Quality Gates
Administration
More
Administrator
Apache Log4j
25. Mai 2016 20:40 Version 1.2.18-SNAPSHOT
Issues
Measures
Code
Dashboards
Administration
Quality Gate
Failed
since 1.2.17
1 New Vulnerabilities
> 0
Bugs & Vulnerabilities
Leak Period: since 1.2.17
started vor 5 Minuten
22 E
Bugs
53 E
Vulnerabilities
0
New Bugs
1
New Vulnerabilitie
Code Smells
1.2k A
Code Smells
23d
Debt
7
New Code Smells
47min
New Debt
started vor 5 Minuten
Duplications
12.1%
Duplications
28
Duplicated Blocks
+2.5%
Duplications
25. Mai 2016
Version: 1.2.17
25. Mai 2016
Structure
Java
100.0%
16k
Lines of Code
-5k
Lines of Code
Apache Log4j 1.2
Bug Tracker
Continuous integration
Developer connection
Home
Sources
Key
log4j:log4j

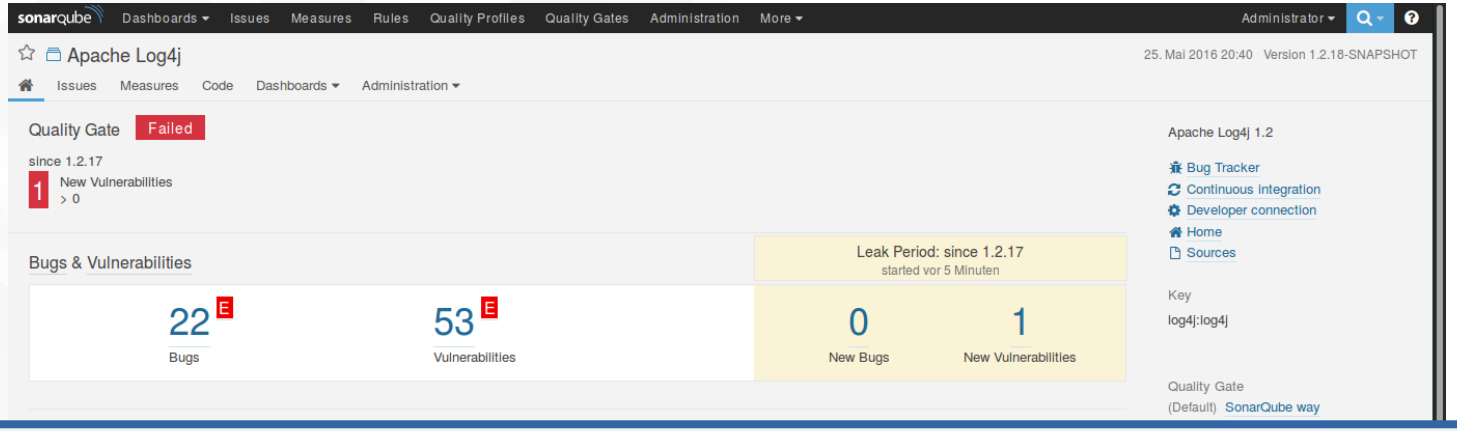
Auswertung

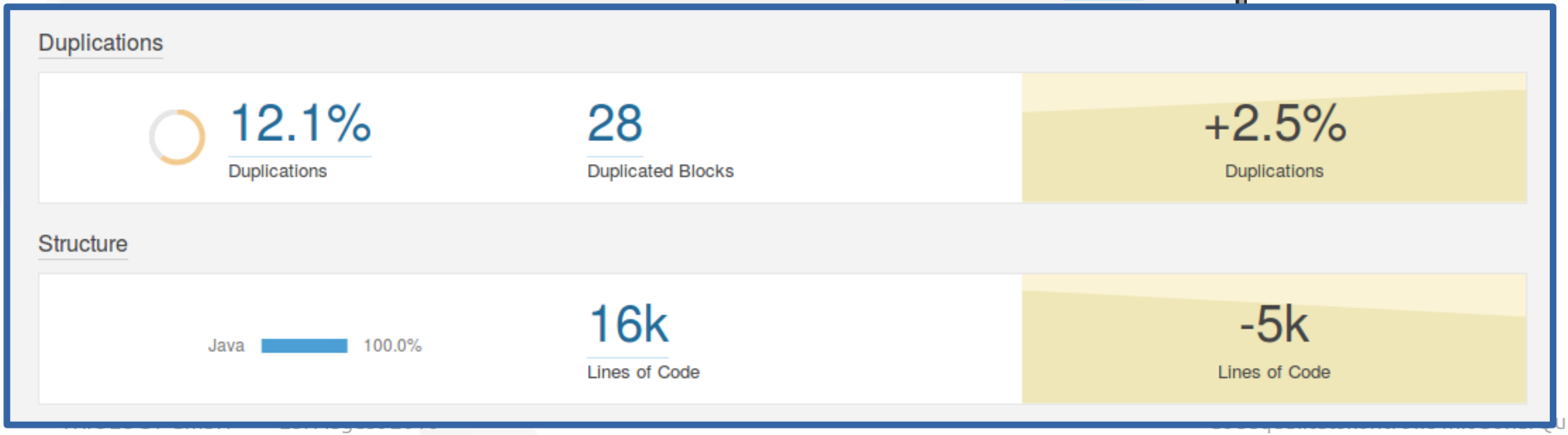

sonarqube
Dashboards
Issues
Measures
Rules
Quality Profiles
Quality Gates
Administration
More
Administrator
Apache Log4j
25. Mai 2016 20:40 Version 1.2.18-SNAPSHOT
Issues
Measures
Code
Dashboards
Administration
Quality Gate
Failed
since 1.2.17
1
New Vulnerabilities
> 0
Leak Period: since 1.2.17
started vor 5 Minuten
53
E
Vulnerabilities
0
New Bugs
1
New Vulnerabilities
Code Smells
1.2k
A
Code Smells
23d
Debt
7
New Code Smells
47min
New Debt
started vor 5 Minuten
Duplications
12.1%
Duplications
28
Duplicated Blocks
+2.5%
Duplications
Structure
Java
100.0%
16k
Lines of Code
-5k
Lines of Code
Apache Log4j 1.2
Bug Tracker
Continuous integration
Developer connection
Home
Sources
Key
log4j:log4j
Quality Gate
(Default) SonarQube way
Quality Profiles
(Java) Sonar way
Events
All
Quality Gate: Red (was Green)
25. Mai 2016
Version: 1.2.18-SNAPSHOT
25. Mai 2016
Version: 1.2.17
25. Mai 2016

Auswertung

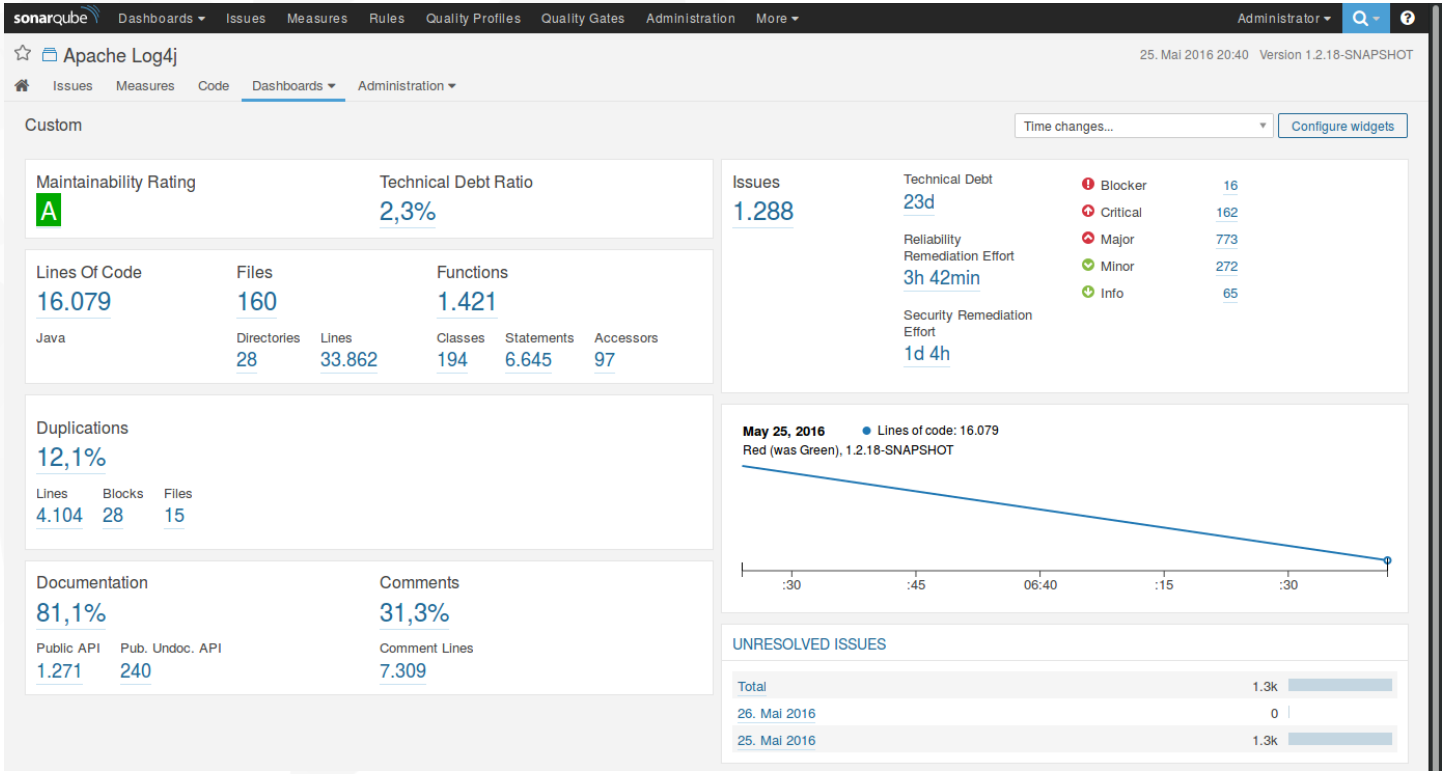

sonarqube
Dashboards
Issues
Measures
Rules
Quality Profiles
Quality Gates
Administration
More
Administrator
Apache Log4j
25. Mai 2016 20:40 Version 1.2.18-SNAPSHOT
Issues
Measures
Code
Dashboards
Administration
Quality Gate
Failed
since 1.2.17
Apache Log4j 1.2
Bug Tracker
Bugs & Vulnerabilities
Leak Period: since 1.2.17
started vor 5 Minuten
22
E
Bugs
53
E
Vulnerabilities
0
New Bugs
1
New Vulnerabilities
Code Smells
1.2k
A
Code Smells
23d
Debt
7
New Code Smells
47min
New Debt
started vor 5 Minuten
Java
100.0%
16k
Lines of Code
-5k
Lines of Code

Auswertung

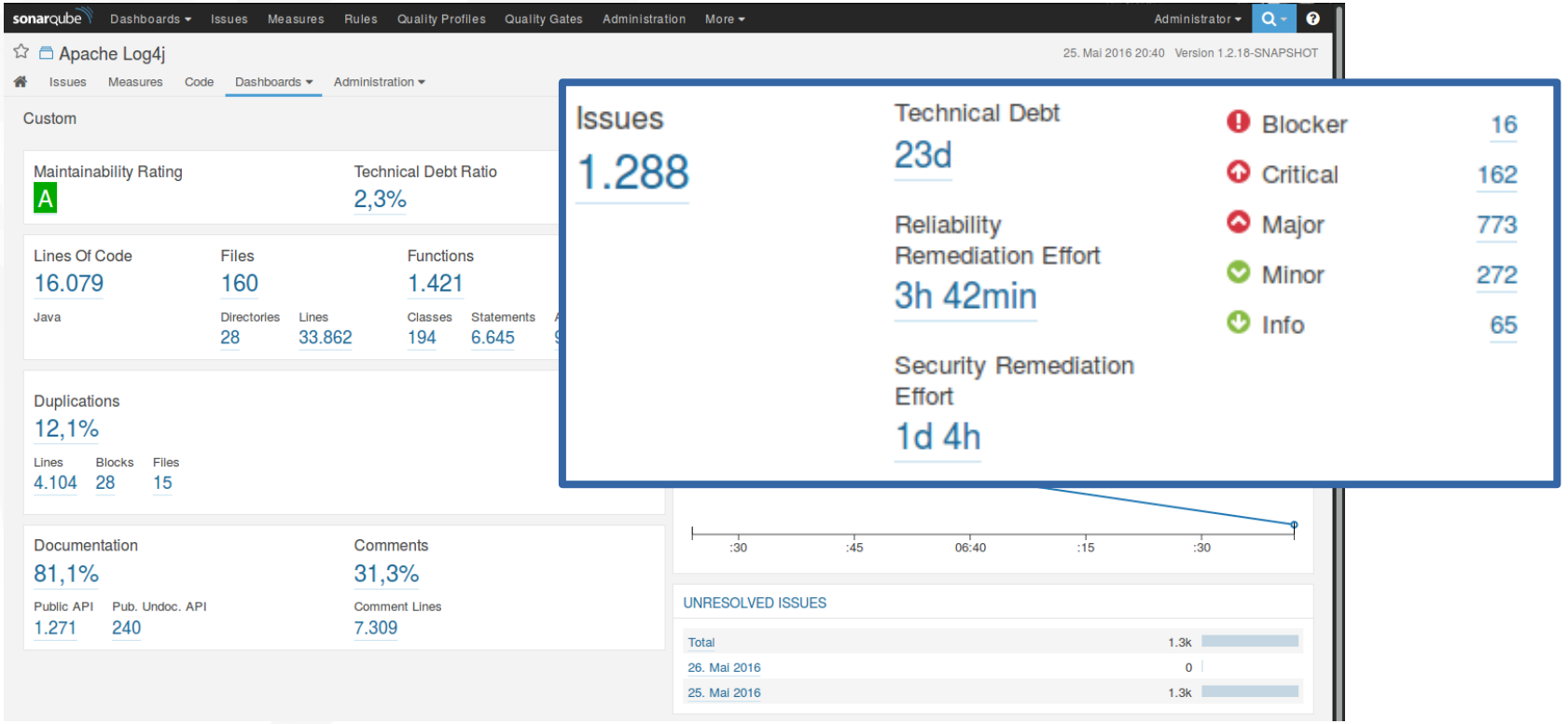

sonarqube
Dashboards
Issues
Measures
Rules
Quality Profiles
Quality Gates
Administration
More
Administrator
Apache Log4j
25. Mai 2016 20:40 Version 1.2.18-SNAPSHOT
Issues
Measures
Code
Dashboards
Administration
Quality Gate
Failed
since 1.2.17
1
New Vulnerabilities
> 0
Bugs & Vulnerabilities
22 E
Bugs
53 E
Vulnerabilities
Leak Period: since 1.2.17
started vor 5 Minuten
0
New Bugs
1
New Vulnerabilities
Apache Log4j 1.2
Bug Tracker
Continuous integration
Developer connection
Home
Sources
Key
log4j:log4j
Quality Gate
(Default) SonarQube way


Duplications
12.1%
Duplications
28
Duplicated Blocks
+2.5%
Duplications
Structure
Java
100.0%
16k
Lines of Code
-5k
Lines of Code

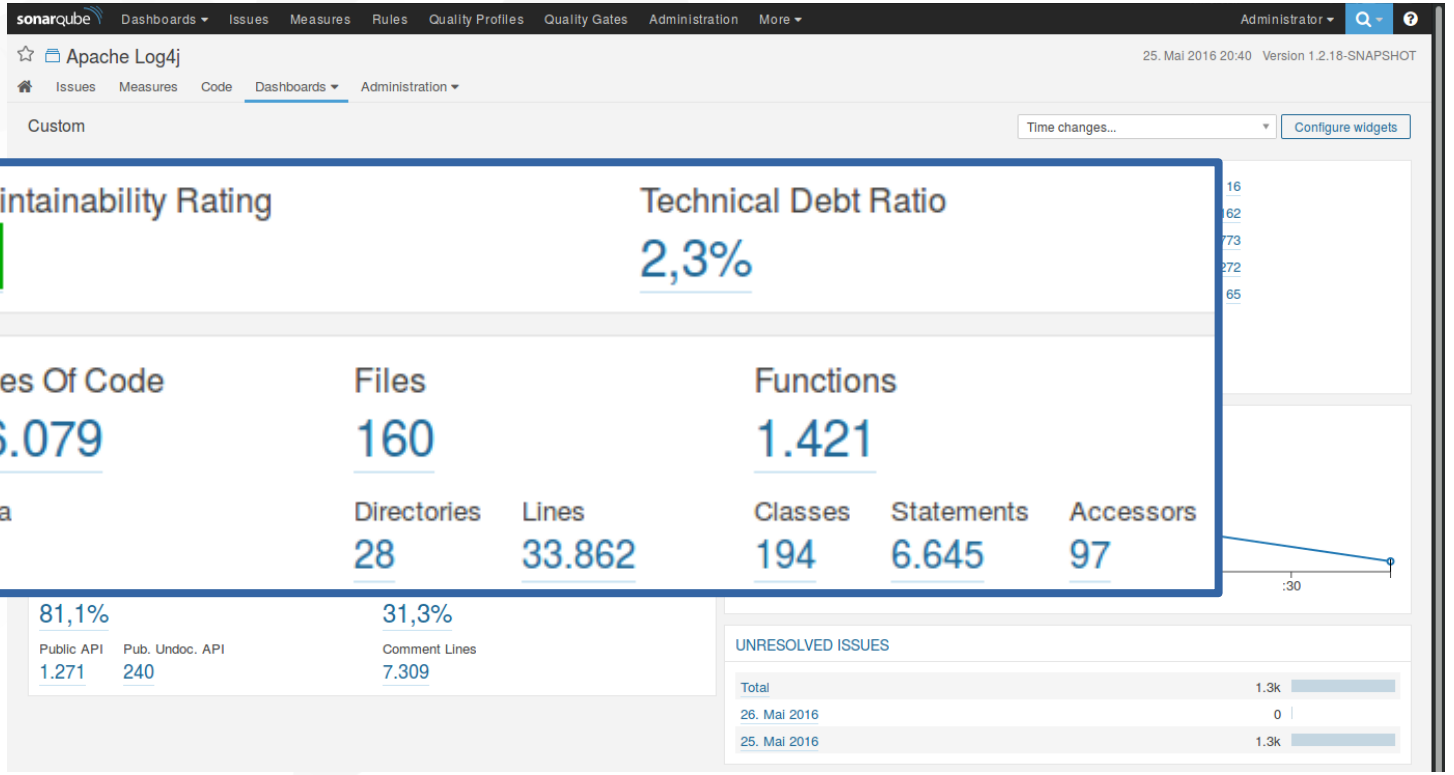
Auswertung


sonarqube
Dashboards
Issues
Measures
Rules
Quality Profiles
Quality Gates
Administration
More
Administrator
Apache Log4j
25. Mai 2016 20:40 Version 1.2.18-SNAPSHOT
Issues
Measures
Code
Dashboards
Administration
Custom
Time changes...
Configure widgets
Maintainability Rating
A
Technical Debt Ratio
2,3%
Lines Of Code
16.079
Java
Files
160
Directories
28
Lines
33.862
Functions
1.421
Classes
194
Statements
6.645
Accessors
97
Duplications
12,1%
Lines
4.104
Blocks
28
Files
15
Documentation
81,1%
Public API
1.271
Pub. Undoc. API
240
Comments
31,3%
Comment Lines
7.309
Issues
1.288
Technical Debt
23d
Reliability Remediation Effort
3h 42min
Security Remediation Effort
1d 4h
Blocker 16
Critical 162
Major 773
Minor 272
Info 65
May 25, 2016
Lines of code: 16.079
Red (was Green), 1.2.18-SNAPSHOT
:30
:45
06:40
:15
:30
UNRESOLVED ISSUES
Total 1.3k
26. Mai 2016 0
25. Mai 2016 1.3k

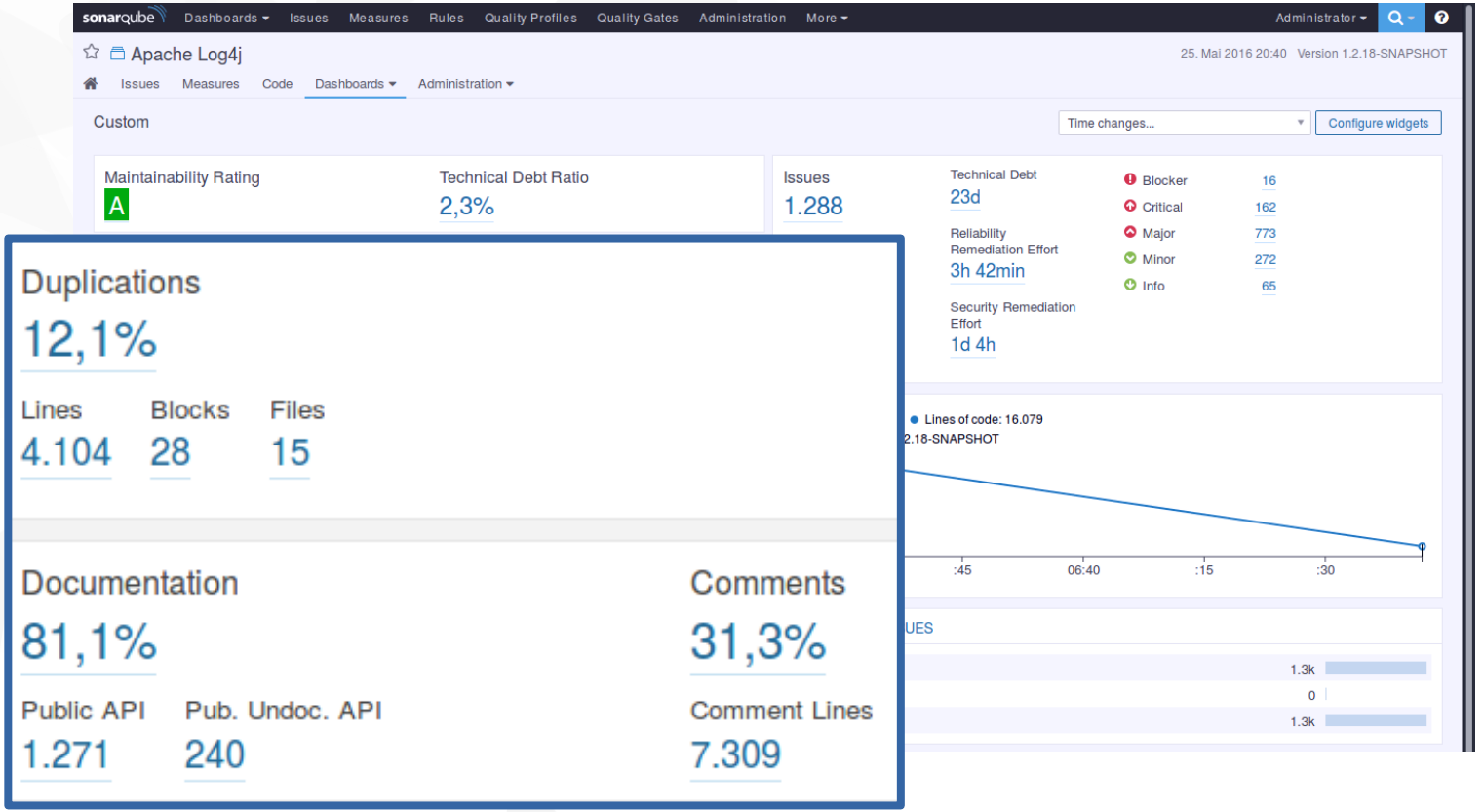
Auswertung


sonarqube
Dashboards
Issues
Measures
Rules
Quality Profiles
Quality Gates
Administration
More
Administrator
Apache Log4j
25. Mai 2016 20:40 Version 1.2.18-SNAPSHOT
Issues
Measures
Code
Dashboards
Administration
Custom
Maintainability Rating
A
Technical Debt Ratio
2,3%
Lines Of Code
16.079
Java
Files
160
Directories
28
Lines
33.862
Functions
1.421
Classes
194
Statements
6.645
Duplications
12,1%
Lines
4.104
Blocks
28
Files
15
Documentation
81,1%
Public API
1.271
Pub. Undoc. API
240
Comments
31,3%
Comment Lines
7.309
Issues
1.288
Technical Debt
23d
Reliability Remediation Effort
3h 42min
Security Remediation Effort
1d 4h
Blocker 16
Critical 162
Major 773
Minor 272
Info 65
:30
:45
06:40
:15
:30
UNRESOLVED ISSUES
Total 1.3k
26. Mai 2016 0
25. Mai 2016 1.3k

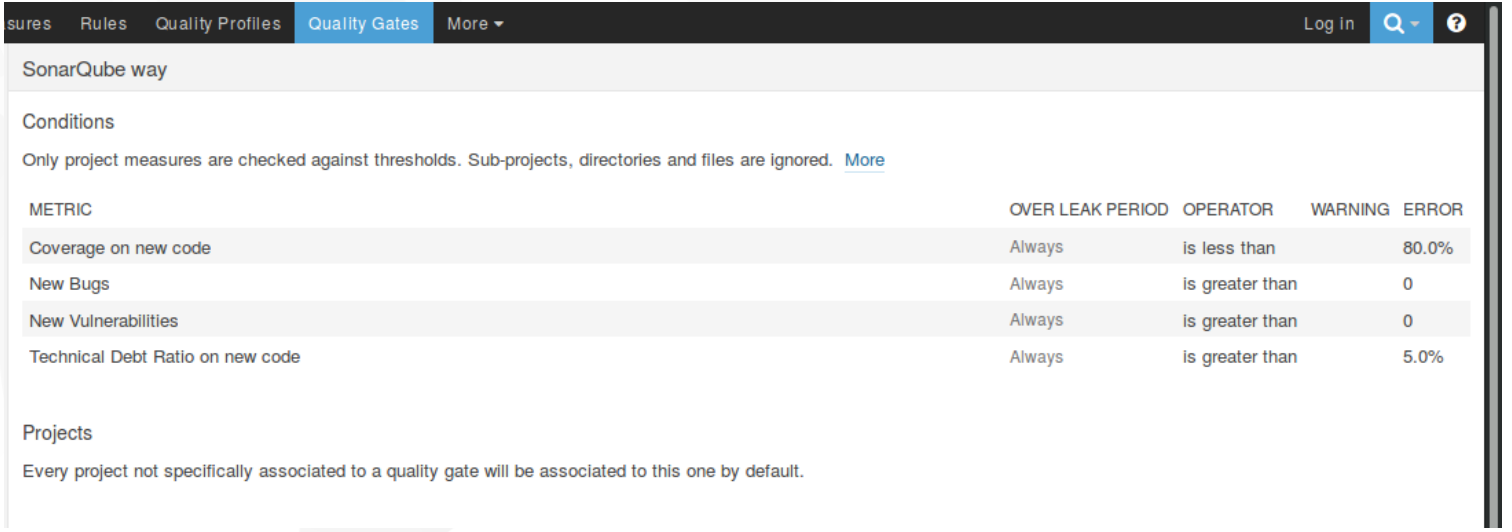
Auswertung


sonarqube
Dashboards
Issues
Measures
Rules
Quality Profiles
Quality Gates
Administration
More
Administrator
Apache Log4j
25. Mai 2016 20:40
Version 1.2.18-SNAPSHOT
Issues
Measures
Code
Dashboards
Administration
Custom
Time changes...
Configure widgets
Maintainability Rating
A
Technical Debt Ratio
2,3%
Lines Of Code
16.079
Java
Files
160
Directories
28
Lines
33.862
Functions
1.421
Classes
194
Statements
6.645
Accessors
97
81,1%
Public API
1.271
Pub. Undoc. API
240
31,3%
Comment Lines
7.309
:30
UNRESOLVED ISSUES
Total
1.3k
26. Mai 2016
0
25. Mai 2016
1.3k

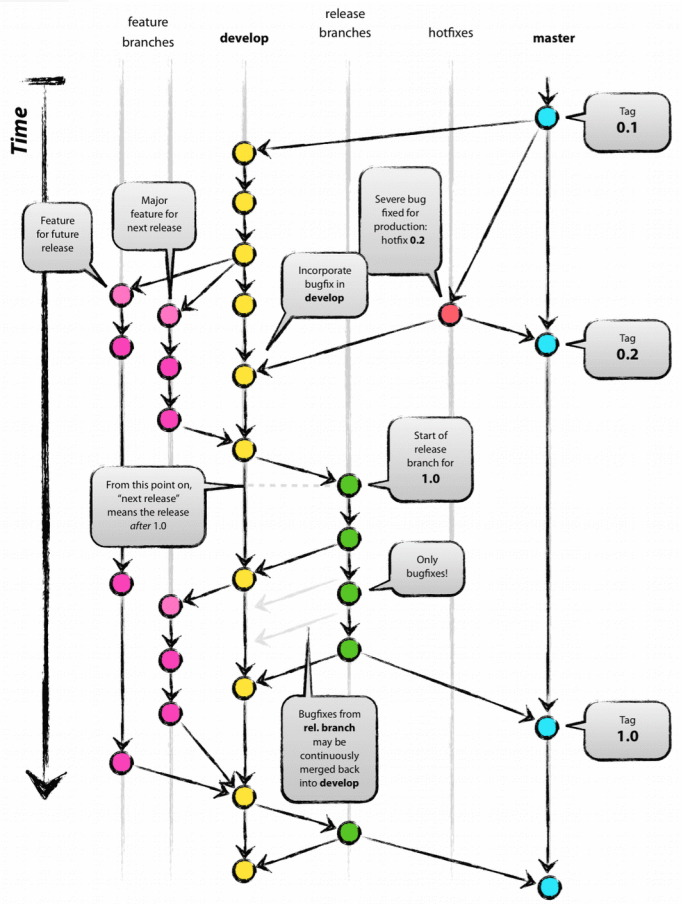
Auswertung


sonarqube
Dashboards
Issues
Measures
Rules
Quality Profiles
Quality Gates
Administration
More
Administrator
Apache Log4j
25. Mai 2016 20:40
Version 1.2.18-SNAPSHOT
Issues
Measures
Code
Dashboards
Administration
Custom
Time changes...
Configure widgets
Maintainability Rating
A
Technical Debt Ratio
2,3%
Issues
1.288
Technical Debt
23d
Reliability Remediation Effort
3h 42min
Security Remediation Effort
1d 4h
Blocker 16
Critical 162
Major 773
Minor 272
Info 65
Lines of code: 16.079
2.18-SNAPSHOT
:45
06:40
:15
:30
UES
1.3k
0
1.3k
Duplications
12,1%
Lines
4.104
Blocks
28
Files
15
Documentation
81,1%
Public API
1.271
Pub. Undoc. API
240
Comments
31,3%
Comment Lines
7.309

Quality Gates

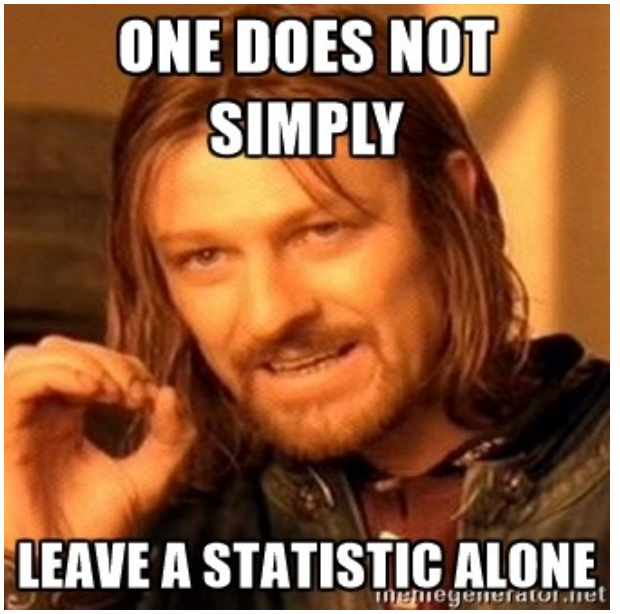


Quality Gates - GitFlow



Interpretation

- Relative Werte > Absolute Werte
- Änderungen > Status Quo
- Metriken verstehen
- Metriken in Relation zueinander stellen



Lines of Code

- Lines of Code - LOC
- Source Lines of Code – SLOC
- Comment Lines of Code – CLOC
- Non-Comment Lines of Code – NCLOC
- Logical Lines of Code - LLOC

Metriken in Relation setzen

```
DTO search(List<List<DTO>> rawData, int id) {
  if(rawData != null) {
    for(List<DTO> sublist : rawData) {
      for(DTO dto : sublist) {
        if(dto.getId() == id) {
          return dto;
        }
      }
    }
  }
  return null;
}
```

Metriken in Relation setzen

```
DTO search(List<List<DTO>> rawData, int id) {
  if(rawData == null) {
    return;
  }

  for(List<DTO> sublist : rawData) {
    for(DTO dto : sublist) {
      if(dto.getId() == id) {
        return dto;
      }
    }
  }
  return null;
}
```

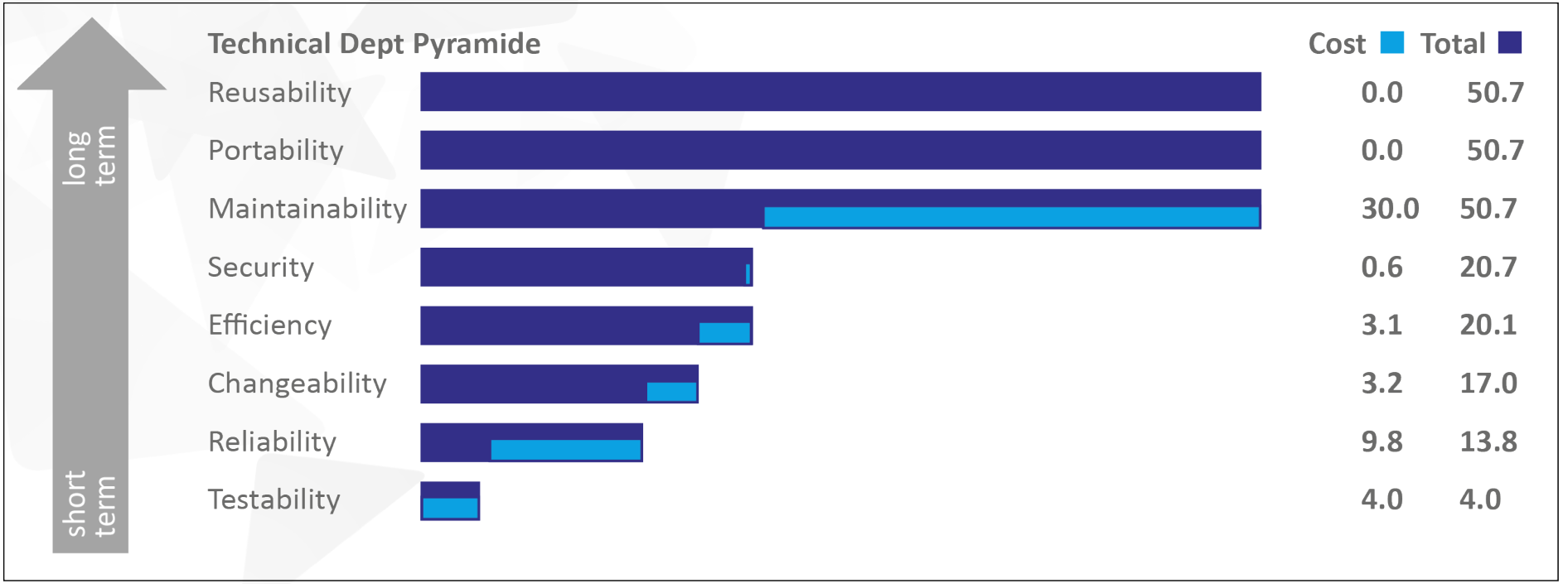
Metriken in Relation setzen

```
void printNumbers() {
  for(int i = 0; i < 10; ++i) {
    if(i % 2 == 0) {
      continue;
    }
    if(i % 3 == 0) {
      continue;
    }
    System.out.println(i);
  }
}
```

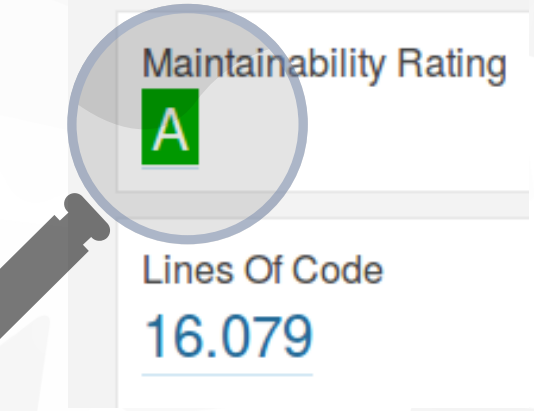
Metriken in Relation setzen

```
void printNumbers() {
  for(int i = 0; i < 10; ++i) {
    if(i % 2 == 0 || i % 3 == 0) {
      continue;
    }
    System.out.println(i);
  }
}
```

SQALE - ISO 9126



Maintainability Rating



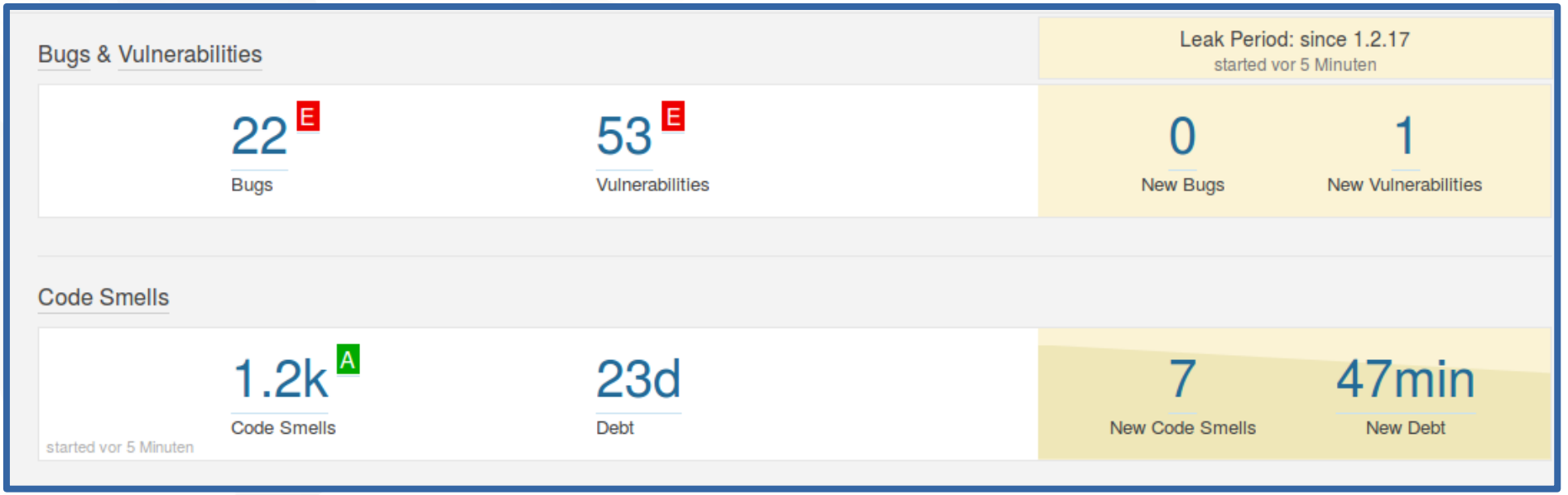
- A: 0 – 0,1
- B: 0,11 – 0,2
- C: 0,21 – 0,5
- D: 0,5 – 1
- E: > 1

- Technical Debt / Development Cost
- Projektgröße: 2.500 LOC
- Technische Schuld: 50 Tage
- 50 / (0.06 * 2.500) = 0,33

Erfahrungswerte

Mit ausreichend großer Codebasis erreichen
viele Legacy-Projekte ein A-Rating

Erfahrungswerte



Erfahrungswerte

„For every 30 rule violations, you can expect on average three minor bugs and one major bug“

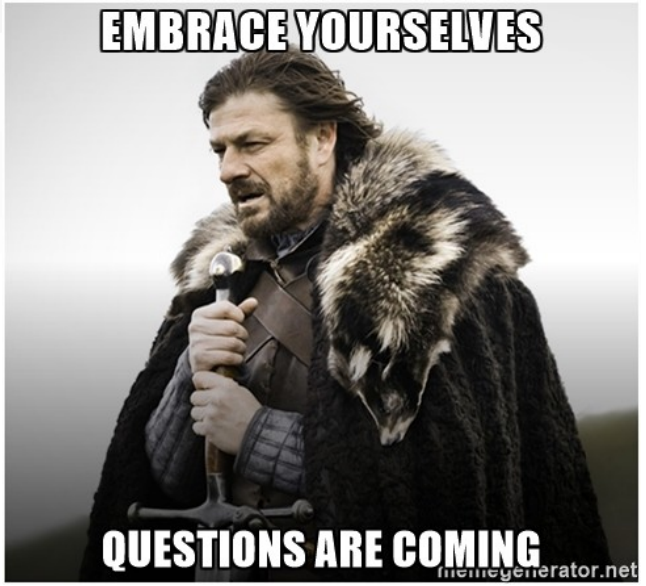
– Embedded Programmer

Erfahrungswerte

Live mit SonarQube zu arbeiten, kann zu Gasfabriken führen

Das heißt?

- Statistiken sind interessant
- Qualitätserhöhung
- Teammotivation
- Messbare Qualität für Kunden


EMBRACE YOURSELVES
QUESTIONS ARE COMING

Buffet ist eröffnet!

Ich hab da mal was vorbereitet

Josha von Gizycki
TRIOLOGY GmbH
josha.von.gizycki@triology.de

https://jug.cloudogu.com/sonar
Logindaten: jug / 20160825