



**viadee**<sup>®</sup>  
IT-Unternehmensberatung

# **Clean Code Adventure Gamification und statische Codeanalyse**

Christoph Meyer

Juni 2018

# Hello, Adventurer!



## Christoph Meyer, Level 39 Software Architect

XP



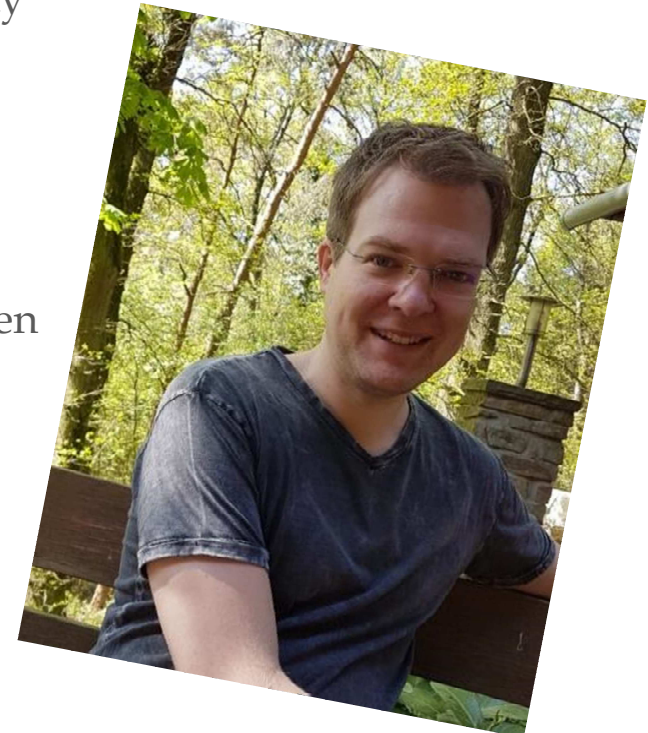
- Hauptquests
  - SWA Skills steigern
  - IT-Berater Quests suchen und lösen
  - Wissen vermitteln: Java, Softwarequalität mit Clean Code, Security



- Nebenquests
  - Familie zusammenhalten
  - Kids großziehen
  - Stärke erhöhen
  - Coole Games zocken



@javaxhr





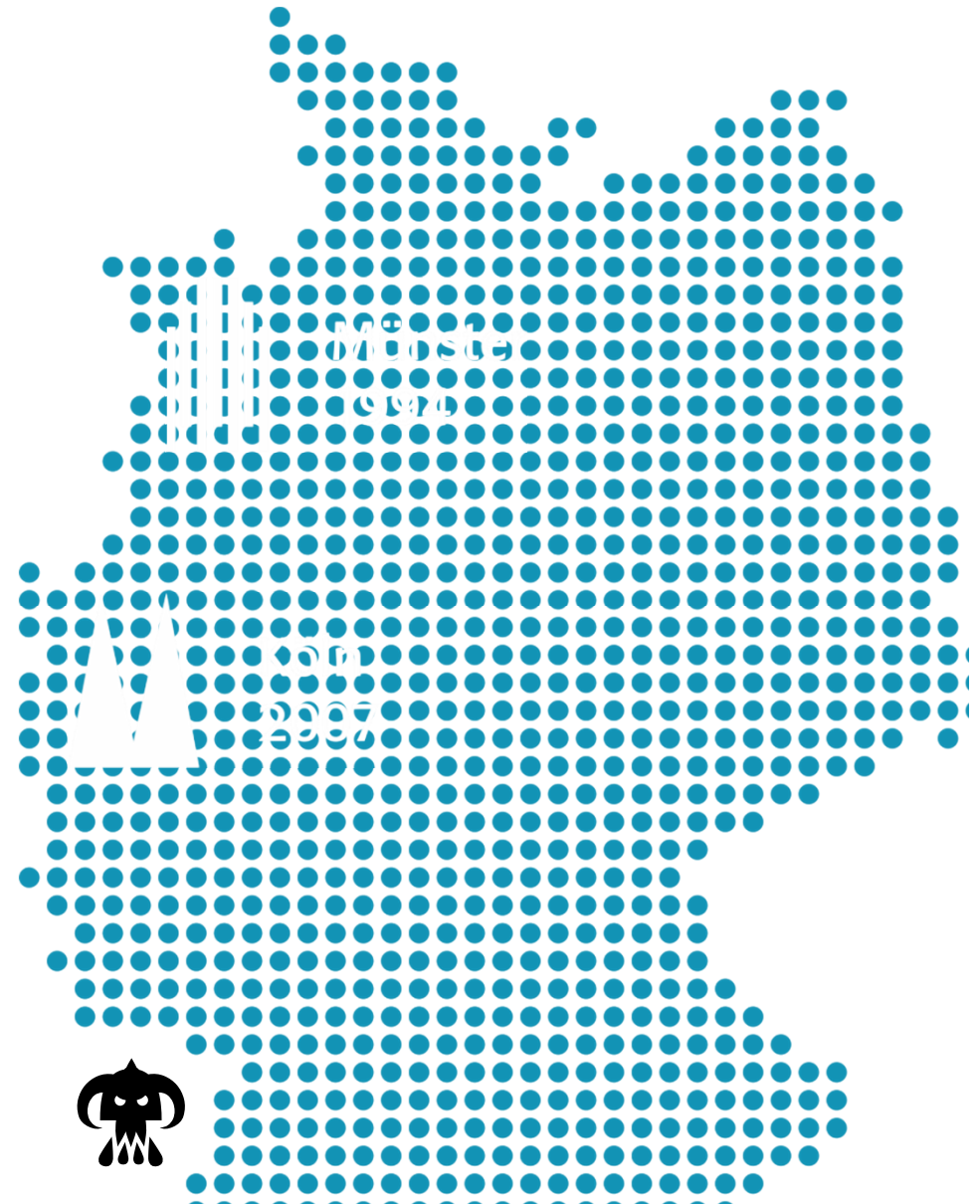
> 130 Mitarbeiter

> 18 Mio. Euro Umsatz

> 2.000 Beratungsprojekte

> 280.000 PT

Komplexität begreifen. Lösungen schaffen.



# Wartungsprojekte!



Wie **steuere** ich  
Wartungsaktivitäten?

**Motivation** für Wartung?



## Code Qualität



Codequalität ist ein Teilaspekt von Softwarequalität, mit dem insbesondere nicht-funktionale Anforderungen wie Konformität, Verständlichkeit, Analysierbarkeit, Modifizierbarkeit, Anpassbarkeit, Prüfbarkeit (so z. B. benannt und definiert nach ISO/IEC 9126) erfüllt bzw. unterstützt werden sollen.

*Wikipedia*

# Synonyme It. Wikipedia

1. Programmierstil
2. Codierstil
3. Codestruktur
4. Codeformat(ierung)
5. ... 



# Klassenunterschiede

```
1 /**
2  * @author [Dein Name hier] - Wir coden als Autoren, sagt Onkel Bob!
3  */
4 public class Gretchenfrage {
5
6     @SuppressWarnings("unused")
7     private boolean persoenlicherStil; // Stilfrage?
8
9     @SuppressWarnings("unused")
10    private boolean ansichtssache; // wirklich privat?
11
12    Level codingErfahrung; // vollstaendige Sicht?
13
14    // OpenJDK 10
15    // var rateMich = StaticUtils.generateValue();
16
17 }
```

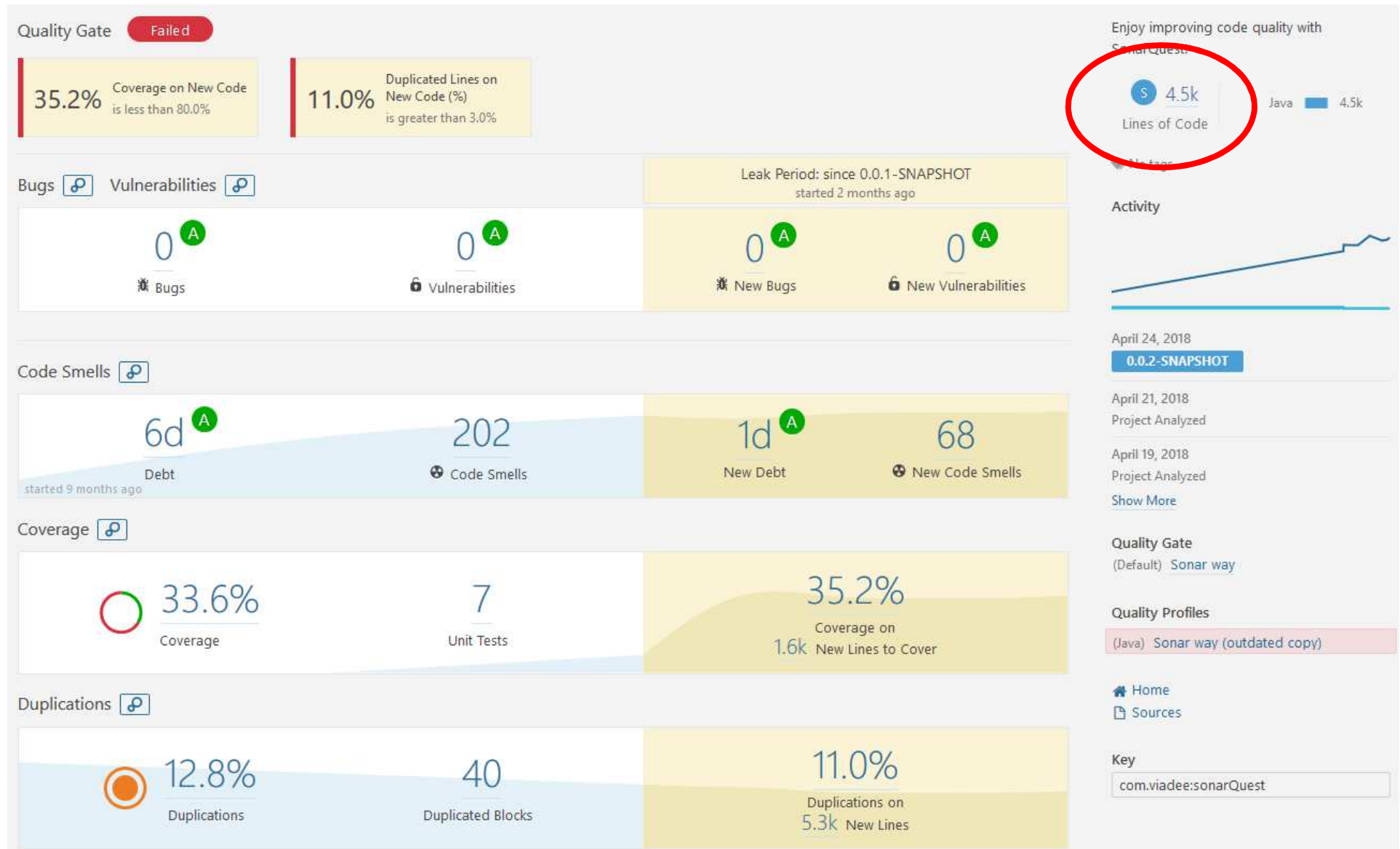


# Codequalität ermitteln

- Wenn ich was verändern will, muss ich die Codequalität objektiv messen!
- Statische Codeanalyse to the rescue!

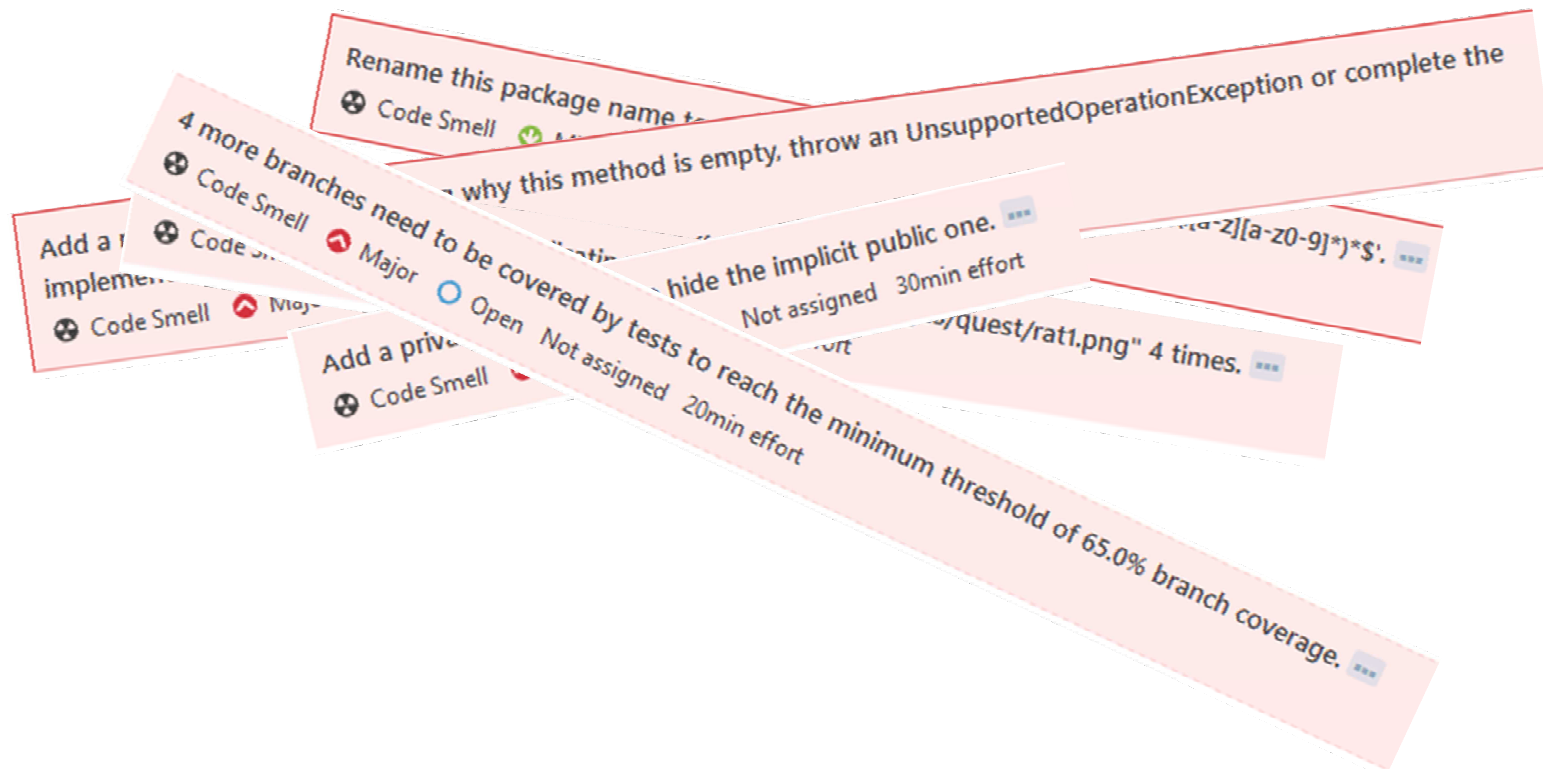


# SonarQube – niedliches neues Projekt



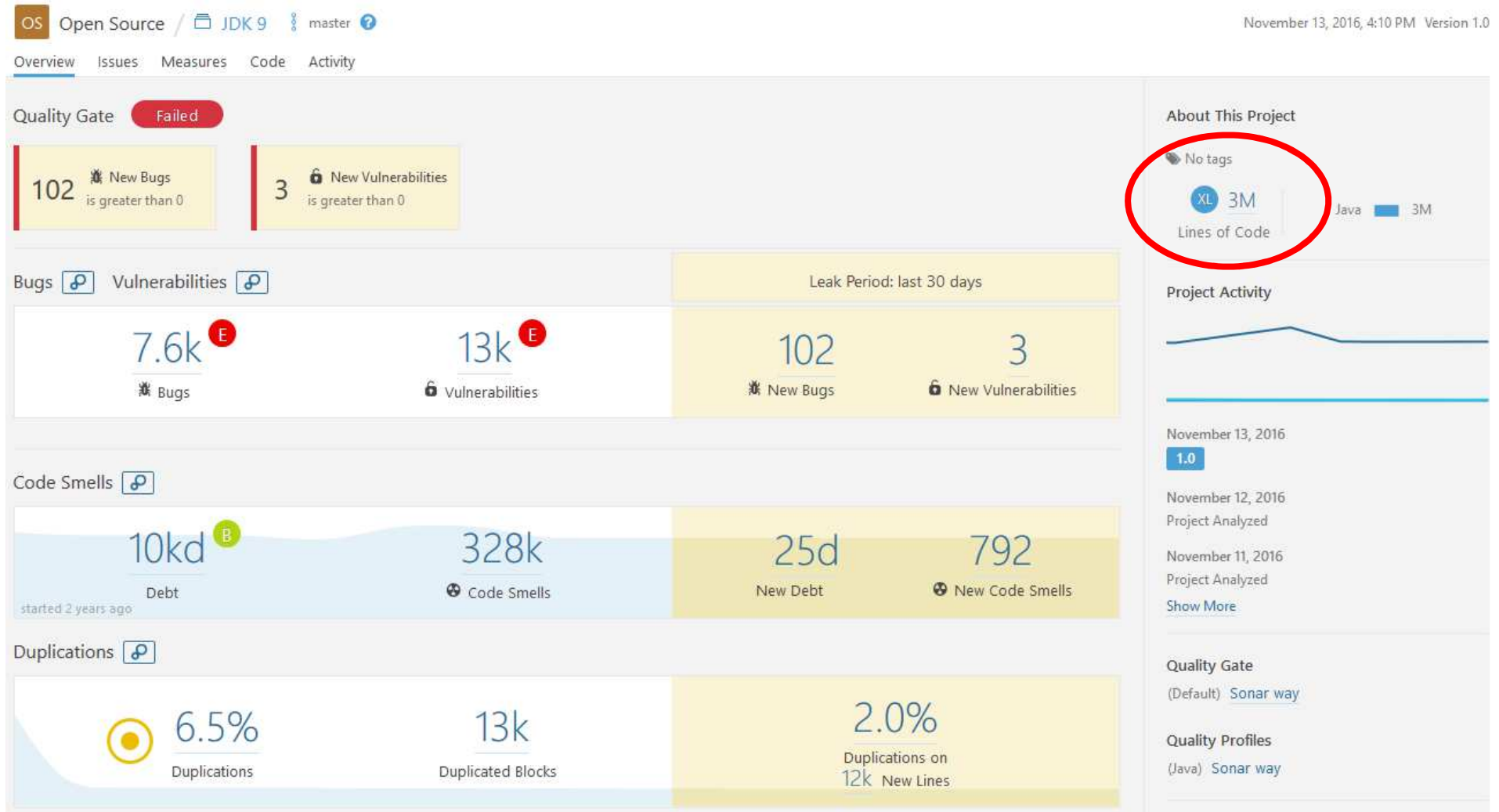
# SonarQube – Codequalität & Code Smells

Hauptindikator: Code Smells → Maintainability



vgl. SonarQube „Rules Explorer“, <https://rules.sonarsource.com/java/>

# SonarQube – nicht ganz so neues Projekt



<https://sonarcloud.io/dashboard?id=net.java.openjdk%3Ajdk9>

# Weitergedacht...

1. Ok, wir haben einen Haufen Code mit geringer Codequalität und das jetzt auch sehr gut dokumentiert – und jetzt?
2. Das Alltags- und Motivationsproblem! 
3. Wo und wie fange ich an, wenn ein Riesenberg Codeprobleme vor mir liegt?
4. Und wie behebe ich den „Wasserschaden?“

# Zitat (Realitätscheck!)

Wir alle haben uns das Chaos angeschaut, das wir gerade angerichtet hatten, und dann beschlossen, es an einem anderen Tag zu beseitigen.

Wir alle haben die Erleichterung gefühlt, zu sehen, dass unser chaotisches Programm lief, und beschlossen, dass ein laufendes Chaos besser wäre als nichts.

Wir alle haben uns vorgenommen, später zurückzukommen und das Chaos zu beseitigen.

*(Robert C. Martin, Clean Code, 2009)*



# Gesetz von Dave LeBlanc

Später  
=  
Niemals



## Level 2

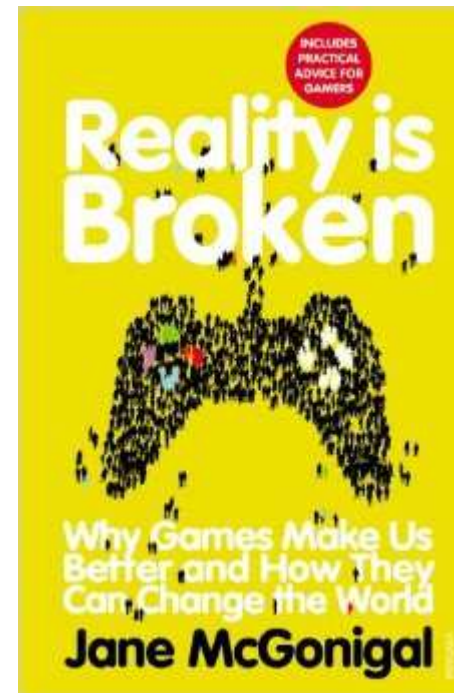
# Gamification



# Gamification

## Schlüsselfaktoren

1. Ziel
2. Herausforderung
3. Feedback-System
4. Freiwilligkeit



## Level 3

### SonarQuest



**Level 1 → Level 2 → Level 3**

Statische Codeanalyse + Gamification =



## **Die Idee: Wartung als RPG**

- OpenSource-Spiel für Entwickler, Agilisten, Manager & Gamer
- Statische Code Analyse liefert Aufgaben (Issues)
- Die Teamleitung (Gamemaster) überführt diese in Herausforderungen (Quests) & Abenteuer (Adventures)
- Entwickler werden zu einer Heldengruppe, die gemeinsam Abenteuer erlebt und Quests löst
- Mit der Zeit entwickeln sich die Charaktere (Avatare)
  - Erfahrungspunkte (XP)
  - Gold
  - Artefakte
  - ...



## **Spielkonzept**

- Quest
  - Aufgabe: 1 – n Issues lösen
  - Belohnung in Gold und Erfahrungspunkten
  - Umfang und Zeitrahmen: Story-Level - 1 Sprint
- Adventure
  - Verbindet mehrere Quests zu einer Geschichte
  - Belohnung: Sonderpunkte für Absolvieren des Adventures
  - Umfang und Zeitrahmen: Epic-Level – mehrere Sprints
- Sonderaufgaben (Special Tasks)
  - Zusätzliche Aufgaben neben den Issues
  - Motivationsfaktor für Spieler



## **Spielkonzept**

- Freiwilligkeit
  - Keine Automatismen, kein Tracking
  - Pull-Prinzip: Entwickler holen sich Quests aktiv
- Auswertung
  - Issues: Statusupdates aus sonarQube
  - Special Tasks: Gamemaster entscheidet über Sonderaufgaben
- Kooperatives Konzept
  - Bonuspunkte für alle Beteiligten bei Lösung von Quests & Adventures
  - Entwicklung individueller Charaktere & Fähigkeiten
  - keine Leaderboards, kein Malus-System





## **Storytelling**

- Jedes Projekt ist eine eigene Welt (World) mit eigenem Charakter
- Jede Quest ist eine Geschichte, jedes Adventure ein „Epos“
  - Bindung an Charaktere
  - „Cliffhanger-Effekt“ mit jedem Sprint
- Jeder Entwickler ist ein Held mit individuellen Eigenschaften und einer eigenen Geschichte

# Live-Demo



# 1 UP?



Wir suchen Mitstreiter für das  
Open Source Projekt **SonarQuest**!

<https://github.com/viadee/sonarQuest>

# Vielen Dank für die Aufmerksamkeit!

**viadee**<sup>®</sup>  
IT-Unternehmensberatung



**Christoph Meyer**

Christoph.Meyer@viadee.de

**viadee** Unternehmensberatung GmbH

Anton-Bruchhausen-Straße 8

48147 Münster

Tel: +49 251 7 77 77 171

**www.viadee.de**

Gibt's noch Fragen?

Dann jetzt!



# Links und Quellen

- Robert C. Martin, *Clean Code*, 2009
- SONAR Metrics  
<https://docs.sonarqube.org/display/SONAR/Metric+Definitions>
- Copyright-Freie Grafiken auf vielen Folien von [www.pixabay.com](http://www.pixabay.com)
- Gamification in „Reality is Broken“ von Jane McGonigal, Random House UK Verlag, 2012
- Icons aus <https://nagoshiashumari.github.io/Rpg-Awesome/> und <https://github.com/nagoshiashumari/Rpg-Awesome> mit Open Font Licence
- Gesetz von Dave LeBlanc <http://on-agile.blogspot.de/2007/04/why-you-wont-fix-it-later.html>
- „Water Leak“ in der SonarQube-Doc  
<https://docs.sonarqube.org/display/SONAR/Fixing+the+Water+Leak>
-