

# Metrics

## Einleitung:

Ich habe mir einen SonarQube Server aufgesetzt und eine kleines Tic Tac Toe Spiel programmiert, da ich keinen Code von der Arbeit verwenden wollte. SonarQube habe ich mit Hilfe von Maven integrieren können. Ich habe absichtlich ein paar Fehler eingebaut um zu sehen wie gut die Korrektur funktioniert. Außerdem habe ich einfach programmiert, so wie ich immer programmieren würde um fest zu stellen welche Fehler sich bei mir eingeschlichen haben, über die Jahre.

## Das ist das Ergebnis:

```
12 ...      public static void main( String[] args )
13 ...      {
14 ...          logger = Logger.getLogger(Logger.GLOBAL_LOGGER_NAME);
15 ...          logger.setLevel(Level.ALL);
16 ...          System.out.println( "Der Print Error" );
17 ...
18 ...
19 ...          Title();
20 ...          Bot bot = new Bot();
21 ...          bot.answer(startGame());
22 ...
23 ...          sayWhatyouWant( "fasfasfasa" );
```

Replace this use of System.out or System.err by a logger. Why is this an issue? 17 hours ago ▾ L16 🔗  
🔗 Code Smell 🚨 Major 🔵 Open Not assigned 10min effort 🔗 bad-practice, cert

Er kriedete mit System.out print an, weil er dachte ich würde damit loggen. Dies mache ich in dem Fall zwar nicht, da es eine CLI Anwendung ist, aber generell mache ich das sehr oft. Habe mich auch nie mit dem Logger beschäftigt. Ich finde ein guter Tipp.

```
... 🔗      public static void Title(){
```

Rename this method name to match the regular expression `^[a-z][a-zA-Z0-9]*$`. Why is this an issue? 16 hours ago ▾ L28 🔗  
🔗 Code Smell 🟢 Minor 🔵 Open Not assigned 5min effort 🔗 convention

Methode Falsch benannt ... ok hat er recht.

```
... 🔗      public static String startGame(){
```

Refactor this method to not always return the same value. Why is this an issue? 6 minutes ago ▾ L32 2 🔗  
🔗 Code Smell 🚨 Blocker 🔵 Open Not assigned 4min effort 🔗 No tags

```
    if (choise.contains("1") |choise.contains("1") | choise.contains("2") | choise.contains("3")){
        return choise;
    }else{
        System.out.println("Keine gültige Auswahl! ");
        startGame();
    }
    return choise;
```

Hier habe ich zwei mal Return benutzt. War wirklich keine gute Entscheidung. Wollte einfach schnell fertig werden. Gebe ich ihm auch recht.

```
if (choise.contains("1") |choise.contains("1") | choise.contains("2") | choise.contains("3")){
```

Correct this "|" to "&" and extract the right operand to a variable if it should always be evaluated.

29 minutes ago ▾ L39 🔗

Why is this an issue?

🐞 Code Smell 🚫 Blocker 🔵 Open Not assigned 5min effort

👤 cert

Das hätte ich nie als einen Fehler erkannt. Wusste gar nicht, das es sich hier bei um einen Bug handeln könnte.

```
public static void sayWhatyouWant(String text){  
  
    for (int i= 10 ; i > -1 ; i--){  
        int result = 10/i;  
        String Result = Integer.toString(result);
```

Dies ist ein von mir konstruierter Fehler. Variable verlangen, welche nicht benutzt wird. (text) Mir ist an dem Beispiel aufgefallen, das SonarQube nicht immer in die Logik schaut, da die Funktion hier eindeutig dazu führt, dass durch 0 geteilt wird. Dies wird aber nicht erkannt.

```
public static void sayWhatyouWantt(String text) {
```

```
    while (true){  
        logger.warning("While True Fehler");  
        System.out.println(text);
```

Replace this use of System.out or System.err by a logger. Why is this an issue?

17 hours ago ▾ L60 🔗

🐞 Code Smell 🚫 Major 🔵 Open Not assigned 10min effort

👤 bad-practice, cert

```
    }
```

Add an end condition to this loop. Why is this an issue?

17 hours ago ▾ L61 🔗

🐛 Bug 🚫 Blocker 🔵 Open Not assigned 15min effort

👤 cert

Bei while True jedoch reagiert SonarQube wieder.

Fazit:

Mir gefällt dieses System tatsächlich ganz gut. Zwar werde ich ungern von einem Programm hinsichtlich meines Programmierkünste kritisiert, jedoch waren die Lösungsvorschläge in den meisten Fällen angebracht. Gut finde ich auch, dass bei jedem Fehler eine Erklärung mitgeliefert wird und ein Vorschlag zum verbessern. Einige Dinge hätte ich nie als Bug erkannt und viele Dinge, die ich aus Unachtsamkeit tue, können so behoben werden. Trotzdem schützt das System nicht vor Dummheit :). Man sollte trotzdem wissen was man tut, siehe das geteilt durch 0 Beispiel.