* 1. Abgabe in EiSE – Übungszettel 5 – WiSe 2015/16

Von Adam Shafei, Len Williamson & Steffen Pegenau

Tools zur Softwarequalitätssicherung

* + 1. FindBugs

<http://findbugs.sourceforge.net/downloads.html>

* + - 1. Beschreibung:
      2. FindBugs inspiziert den Java Byte Code und versucht mit statischen Analysemethoden Fehlermuster („Bug Patterns“) zu finden, die vorher hinterlegt sein müssen.
      3. Unpräzise Ergebnisse:
      4. Probleme :

1. Problem  
   Beschreibung:  
   Mögliche Lösung:
2. Problem  
   Beschreibung:  
   Mögliche Lösung:
3. Problem  
   Beschreibung:  
   Mögliche Lösung:
   * 1. PMD

<https://pmd.github.io/>

* + - 1. Beschreibung:
      2. PMD untersucht den Quellcode anhand von Regelwerken auf typische Programmierfehler.
      3. Unpräzise Ergebnisse:
      4. Probleme :

1. Problem  
   Beschreibung:  
   Mögliche Lösung:
2. Problem  
   Beschreibung:  
   Mögliche Lösung:
3. Problem  
   Beschreibung:  
   Mögliche Lösung:
   * 1. CheckStyle

<http://checkstyle.sourceforge.net/>

* + - 1. Beschreibung:
      2. CheckStyle prüft Quellcode auf die Einhaltung von (selbstgesetzten) Programmierregeln.
      3. Unpräzise Ergebnisse:
      4. Probleme :

1. Problem  
   Beschreibung:  
   Mögliche Lösung:
2. Problem  
   Beschreibung:  
   Mögliche Lösung:
3. Problem  
   Beschreibung:  
   Mögliche Lösung:
   * 1. JDepend

<http://clarkware.com/software/JDepend.html>

* + - 1. Beschreibung:
      2. JDepend generiert nach einer Quelltextanalyse Metriken zur Messung der Design-Qualität (Erweiterbarkeit, Wiederverwendbarkeit, Wartbarkeit, Abhängigkeiten).
      3. Unpräzise Ergebnisse:
      4. Probleme :

1. Problem  
   Beschreibung:  
   Mögliche Lösung:
2. Problem  
   Beschreibung:  
   Mögliche Lösung:
3. Problem  
   Beschreibung:  
   Mögliche Lösung:
   * 1. Dependency Finder

<http://depfind.sourceforge.net/>

Beschreibung:

* + - 1. DependencyFinder erzeugt aus Basis einer Bytecode-Analyse Abhängigkeitsgraphen und Qualitätsmetriken (z.B. Codezeilen / Methode)
      2. Unpräzise Ergebnisse:
      3. Da DependencyFinder nur Metriken und Abhängigkeitsgraphen erzeugt, liegt die Gefahr vor allem in einer unpräzisen Interpretation durch den Anwender. Grenzwerte müssen für die einzelnen Aspekte sinnvoll angepasst werden, um die Interpretation zu erleichtern.
    1. Checker Framework

<http://types.cs.washington.edu/checker-framework/>

Beschreibung:

* + - 1. Das Checker Framework erweitert das Typsystem von Java mit Annotationen, um dem Nutzer das Finden und Vermeiden von Fehlern zu erleichtern.
      2. Unpräzise Ergebnisse:

Drei Funde des Checker Frameworks