

Statische Codeanalyse — wo ist der Fehler in meinem Programm?

Wolfgang Dautermann

10. Januar 2014

1 Motivation

Speziell bei Open-Source Software arbeiten oft Teams, die in der ganzen Welt verteilt sind (und sich im Real-Life manchmal gar nicht kennen), gemeinsam an einem Projekt. Was kann man tun um gemeinsam zu einem einheitlichen Coding-Stil zu finden und Fehler zu vermeiden?

2 Geplante Struktur des Vortrags

- Beispiele von Fehlermeldungen (und insbesondere **nicht vorhandenen Fehlermeldungen!**) von diversen C-Compilern.
- für C-Programmierer: Splint, cppcheck
- für Perl: Perlritic
- für PHP: PHP Code Sniffer
- für \LaTeX : chktex, nag
- Tools für weitere Programmiersprachen
- Diverse Online-Services

3 Zielgruppe

Entwickler und Programmierer, die daran interessiert sind, Fehler in ihren Programmen zu finden. Teams, die an einem gemeinsamen Open-Source Projekt arbeiten und einen gemeinsamen Codingstil finden wollen. Programmierkenntnisse (z.B. C, Perl, PHP, . . .) sind wünschenswert.

4 Praktische Vorführung

Zu den vorgestellten Tools wird es auch jeweils eine kurze praktische Vorführung geben.

5 Literatur/Weblinks

- Artikel “Splint und Co: Tools zur statischen Code-Analyse” in LinuxUser 11/2012
- <http://www.linux-community.de/Internal/Artikel/Print-Artikel/LinuxUser/2012/11/Splint-und-Co-Tools-zur-statischen-Code-Analyse>