

### **Warum ist das interessant?**

Insbesondere der Begriff Code Smell [1] ist in der Software Entwicklung sehr geläufig und beschreibt funktional korrekten Code, welcher aber auf Grund der schlechten Struktur in der Regel zu Problemen im weiteren Verlauf der Entwicklung führt. So lassen auch wir dieses Konzept früh in unsere Veranstaltungen einfließen um die Qualität der Ausbildung zu verbessern und bringen z.B. den Teilnehmern des Praktikums *Programmieren in Java* bei duplizierten Code oder zu lange Methoden zu vermeiden. Abseits von Code Smells gibt es aber noch weitere „üble Gerüche“, die zu Problemen in anderen Phasen der Softwareentwicklung führen können, z.B. während der Anforderungsanalyse [2] oder des Architekturdesigns [3].



### **Was soll ich tun?**

In dieser Arbeit sollen Sie systematisch, basierend auf der wissenschaftlichen Literatur, eine Übersicht über existierende Smells im Kontext der Softwareentwicklung und deren Bedeutungen erstellen. Dazu ist es erforderlich zunächst eine umfassende Recherche über existierende Arbeiten bzgl. Smells durchzuführen.

### **Welche Ergebnisse werden erwartet?**

Erwartet wird eine Mapping Study [4] über das Thema von Smells im Kontext der Softwareentwicklung.

### **Was bringt mir das?**

Neben den Kreditpunkten und der Note für den oben genannten Typ dieser Arbeit:

- Umfassende Kenntnisse über gute Software Entwicklung und wie man Fallstricke vermeidet, über den gesamten Software Lebenszyklus hinweg
- Wissen über die Erstellung einer wissenschaftlichen Recherche
- Ggf. eine wissenschaftliche Publikation im Anschluss der Arbeit

[1] M. Fowler, K. Beck, J. Brant, W. Opdyk, D. Roberts. Refactoring: Improving the Design of Existing Code. Addison Wesley, 1999.

[2] H. Femmer, D. M. Fernández, E. Juergens, M. Klose, I. Zimmer, J. Zimmer. Rapid requirements checks with requirements smells: two case studies. In Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering, S. 10-19. ACM, 2014.

[3] M. Lippert, S. Roock. Refactoring in large software projects: performing complex restructurings successfully. John Wiley & Sons, 2006.

[4] D. Budgen, M. Turner, P. Brereton, B. Kitchenham. Using mapping studies in software engineering. Proceedings of PPIG. Vol. 8., S. 195-204, Lancaster University 2008.

### **Kontakt**

Sascha El-Sharkawy (M. Sc.)  
elscha@sse.uni-hildesheim.de

### **Aufteilung der Arbeit**

Theorie	Implementierung	Literatur
20%	0%	80%