

Automatische Erzeugung von Zusammenfassungen für Quellcode-Änderungen

Hintergrund

Code-Reviews [1] sind ein wichtiges Verfahren zur Software-Qualitätssicherung. In der Wirtschaft finden Code-Reviews überwiegend „änderungsbasiert“ statt, d.h. ein Softwareentwickler (der „Reviewer“) prüft alle Quellcode-Änderungen, die ein anderer (der „Autor“) im Rahmen einer Entwicklungsaufgabe vorgenommen hat. Etwas technischer ausgedrückt erfolgt also ein Review von „Patches“ bzw. Mengen von „Commits“ z.B. in git oder Subversion. Um ein effizientes Code-Review zu ermöglichen ist es sinnvoll, dem Reviewer zu Beginn des Reviews eine Zusammenfassung der wichtigsten Änderungen zu geben. Um den Autor zu entlasten, soll dessen textuelle Beschreibung durch eine automatisiert erzeugte Zusammenfassung ergänzt werden.

Aufgabe

Es soll ein Programm entwickelt werden, das die für den Reviewer wichtigsten Teile einer Menge von Commits extrahiert und textuell aufbereitet. Hierzu ist zunächst zu klären, wie solche Zusammenfassungen aussehen sollen und welches die „wichtigsten Teile“ sind. Dabei kann u.a. auf Kooperationspartner des Betreuers aus der Wirtschaft zurückgegriffen werden. Zusätzlich soll eine Einarbeitung in den aktuellen Stand der Forschung zum Thema erfolgen (z.B. [2, 3]). Ausgewählte Teile der identifizierten Anforderungen sind anschließend in Software umzusetzen und die erzielten Ergebnisse zu bewerten. Idealerweise erfolgt die Umsetzung in Java und durch Erweiterung des am FG SE entwickelten Code-Review-Tools CoRT [4].

Zu der Arbeit ist ein Bericht zu erstellen, in dem die Vorgehensweise, Konzepte und Ergebnisse beschrieben werden.

Organisatorisches

Betreuer: Tobias Baum, tobias.baum@inf.uni-hannover.de, G 308

Beginn: ab sofort

Literatur

- [1] Tobias Baum, Olga Liskin, Kai Niklas, Kurt Schneider: A Faceted Classification Scheme for Change-Based Industrial Code Review Processes, In QRS, 2016.
- [2] Buse, R. P. & Weimer, W. R. Automatically documenting program changes. Proceedings of the IEEE/ACM international conference on Automated software engineering, 2010, 33-42
- [3] Liu, Y.; Sun, X.; Liu, X. & Li, Y. Supporting program comprehension with program summarization. Computer and Information Science (ICIS), 2014 IEEE/ACIS 13th International Conference on, 2014, 363-368
- [4] <https://github.com/tobiasbaum/reviewtool>