

(Image: Christopher Kuszajewski / pixabay.com, Lizenz: CC 0 Public Domain)

Automatisierte Auswertung von Programmier-Praktika

01 Mar 2016

Tags
Share

Degree:	Bachelor (B. Sc.)
Study programme:	Informatik/Softwaretechnik
State:	announced
Author:	Timo Hamann
Company:	CoSA (http://www.cosa.fh-luebeck.de)
Language:	German

Der Einsatz von Continuous Integration Lösungen gehört mittlerweile zum bewährten Vorgehen in der Softwareentwicklung. Diese Arbeit soll den Einsatz von Continuous Integration Lösungen im Rahmen der Hochschullehre (Programmierausbildung) untersuchen. Zur Unterstützung von Programmierlehrveranstaltung - insbesondere von Programmierpraktika - soll hierzu eine Lösung zur automatisierten Auswertung von wöchentlichen Abgabeaufgaben auf Basis von

existierenden Continuous Integration Lösungen geschaffen werden. Auf diese Weise sollen Praktikumsbetreuer/innen in Zeiten stetig größer werdender Programmierkurse zeitlich entlastet werden und Studierenden eine ergänzende Möglichkeit des Feedback-gesteuerten Selbststudiums gegeben werden.

Automatische Auswertung von Abgaben

Die automatische Auswertung von Abgaben soll folgende Aspekte umfassen:

- **Korrektheit** der Abgabe mittels Unit-Tests (eine geeignete Struktur für Unit-Tests zu finden ist Gegenstand der Arbeit)
- **Code Coverage** (zur Ermittlung unnötigen Codes für die Aufgabenstellung, hierfür sind geeignete und frei verfügbare Code Coverage Lösungen zu ermitteln)
- **Komplexität** der Abgabe mittels Komplexitätsmetriken (geeignete und einfach zu erhebende Metriken zu finden, ist Gegenstand der Arbeit)
- Einhaltung von **Programmierkonventionen** (z.B. mittels Checkstyle, geeignete Frameworks zu finden ist Gegenstand der Arbeit)

Studierende sollen die automatisierte Auswertung von Abgaben idealerweise aus der Eclipse Entwicklungsplattform heraus anstoßen können. Entsprechend geeignete Plugins sind zu analysieren und deren Eignung zu demonstrieren.

Automatisiertes Feedback für Studierende

Studierende sollen auf Basis dieser Aspekte ein automatisiertes Feedback pro Abgabe

- **in Relation zur Musterlösung**
- aber auch **in Relation zur Peergroup** der anderen Studierenden

erhalten. Details sind im Verlaufe der Arbeit zu klären.

Leistungsübersicht für Studierende und Lehrende

Lehrende und Studierende sollen auf Basis einer solchen Auswertung schneller einen Überblick über den Leistungsstand erhalten. Die Auswertung soll

- für eine Abgabe eines Studierenden (Detailansicht einer Abgabe: für den Studierenden und Lehrende),
- für eine Abgabe über alle Studierenden (Leistungsübersicht einer Abgabe: nur für Lehrende),
- über alle Abgaben eines Studierenden (Leistungsübersicht eines Studierenden: für den Studierenden und Lehrende),
- über alle Abgaben aller Studierenden (Leistungsübersicht eines Semesters: nur für Lehrende)

möglich sein. Details sind im Verlaufe der Arbeit zu klären.

Teilaufgaben

Hierzu müssen im Detail folgende Teilaufgaben bearbeitet und dokumentiert (Bachelorarbeit) werden:

- Durchführen einer **Anforderungsanalyse**.
 - Dies beinhaltet die **Analyse des bestehenden Aufgabenkatalogs** der Programmierveranstaltung Prog I (Inf).
 - Ferner sind insbesondere bestehende Continuous Integration Lösungen hinsichtlich Eignung für den dargestellten Einsatzzweck zu analysieren.
- **Aufbau einer Continuous Integration** Lösung zur automatisierten Evaluation von Abgaben im Rahmen von Programmier-Praktika.
 - Die Lösung soll in einer containerisierten Form mittels Docker ausgebracht werden können.
 - Die Lösung muss Accounts für eine Vielzahl von Studierenden schnell und einfach erzeugen können.
 - Die Lösung muss die oben genannten Auswertungs-Aspekte abdecken.
 - Die Lösung muss sowohl ein Feedback bezogen auf die Musterlösung als auch auf die Peergroup generieren können.
 - Die Lösung muss eine Leistungsübersicht für Studierende und Lehrende darstellen können.
- Der **Nachweis der Realisierbarkeit** soll anhand mindestens 8 Aufgaben des Aufgabenkatalogs Prog I erfolgen.
 - Die zu betrachtenden Aufgaben sollen dabei aus dem ersten, dem zweiten, dem dritten und dem vierten Viertel des ersten Semesters stammen.
 - Ergänzend sollen drei Aufgaben aus dem ersten, zweiten und dritten Drittel des zweiten Semesters betrachtet werden, um abzuleiten inwiefern das Konzept auf freiere und komplexere Aufgabenstellungen übertragbar ist.
 - Auf Basis dieser Aufgaben soll ein "Kochrezept" abgeleitet und

dokumentiert werden, wie beliebige Aufgaben des Aufgabenkatalogs umgewandelt/umgeschrieben werden müssen, damit sie einer automatisierten Evaluierung unterworfen werden können.

- Es sind entsprechende Empfehlung zur Umgestaltung des Aufgabenkatalogs abzuleiten und zu dokumentieren.
- Es ist ein Template zu erstellen, nach dem neue Aufgaben für den Aufgabenkatalog erstellt werden können.
- **Begleitende Dokumentation** der oben angegebenen Schritte, Designentscheidungen und Ergebnisse (Bachelorarbeit)
 - Die Dokumentation muss zusätzlich zu den üblichen Anforderungen an eine Bachelorarbeit, die folgenden Punkte umfassen:
 - Die Dokumentation muss insbesondere einen Anhang beinhalten, wie die Lösung betrieben und konfiguriert werden kann.
 - Die Dokumentation muss ein Kochrezept beinhalten, wie Aufgaben formuliert und gestaltet sein müssen, um diese einer automatisierten Auswertung unterwerfen zu können.

Sollten sich während der Bearbeitung dieser Arbeit einzelne Aspekte als schwer realisierbar erweisen, kann die Aufgabenstellung - nach plausiblen Nachweis der Schwierigkeit - angepasst werden.