

## Werkzeugunterstützung zur Vermeidung von Doppelimplementierungen

- Bachelorthesis  
 Masterthesis

### Aufgabenstellung

Bei der Entwicklung und Pflege großer Softwaresysteme kommt es immer wieder zu Doppelimplementierungen, d.h. es werden Funktionalitäten neuimplementiert, die bereits anderswo im System oder in einer Bibliothek existieren; in der Regel weiß ein Entwickler sich dessen nicht bewusst ist. Gerade bei unklarer Projektstruktur, Fluktuation im Team und langjähriger Weiterentwicklung kommt dies häufig vor und stellt nicht nur einen unnötigen Entwicklungs-Aufwand dar, sondern verursacht auch Folgekosten in Test, Fehlerbeseitigung und der späteren Pflege.

Da sich die beiden Implementierungen mitunter syntaktisch unterscheiden werden sie von üblichen Clone-Detektoren evtl. nicht gefunden. Hierfür sind andere bzw. erweiterte Algorithmen erforderlich. Zudem sind die zu suchenden Muster häufig relativ kurz, was leicht zu false positives bei der Suche führt. Um Doppelimplementierungen effizient und frühzeitig erkennen und Folgekosten vermeiden zu können ist jedoch ein zeitnahes, möglichst präzises Feedback an die Entwickler notwendig; am besten vor dem „Commit“ einer Änderung.

### Zielsetzung der Arbeit

Ziel der Arbeit ist die Konzeption, prototypische Umsetzung und Erprobung eines Werkzeugs (IDE-Plugin) zur Identifikation solcher Doppelimplementierungen. Hierfür soll aufgebaut werden auf existierende Ansätze, insb. eine Masterarbeit zur Lokalisierung von Code Pattern im Repository von <http://stackoverflow.com>, die unter <http://codekoan.org> bereits als Suchmaschine verfügbar ist.

Das Werkzeug soll Entwickler darauf hinweisen, dass die Funktionalität, die sie gerade umsetzen, anderswo bereits vorhanden ist. Als Suchraum dient bei der vorliegenden Arbeit eine Sammlung gängiger Bibliotheken wie Apache Commons (zur Vermeidung von „Neuerfindungen des Rads“). Der Abgleich mit dem in Entwicklung befindlichen System selbst (zur Vermeidung von Duplikaten im System) ist nicht Inhalt dieser Arbeit sondern ggf. einer zukünftigen Folgearbeit. Für diese Arbeit soll zudem fokussiert werden auf die Programmiersprache Java mit der IDE Eclipse und den Einsatz in der Entwicklung und Pflege von Informationssystemen.

Hierfür sind vsl. erforderlich:

- Literaturrecherche zum Stand der Forschung und zu den Grundlagen von <http://codekoan.org>
- Analyse realer Informationssysteme / Befragung von itestra-Experten bzgl. relevanter Beispiele
- Konzeption und prototypische Umsetzung eines Werkzeugs:
  - Zerlegung von Sourcecode in für die Analyse geeignete Einheiten (Patterns).
  - Realisierung eines IDE-Plugins (für Eclipse, optional zusätzlich auch für IntelliJ) für die Erkennung, dass ein Entwickler eine ausreichend große, zusammenhängende Funktionalität implementiert hat, Prüfungsanfrage an codeKoan Server, geeignete Darstellung der Ergebnisse für den Entwickler in der IDE und Möglichkeit zur Sammlung von (anonymisiertem) Feedback zur Verbesserung der Trefferquote.
- Erprobung anhand realer Projekt(e), insb. im Hinblick auf in der Praxis akzeptable Trefferquoten und den Nutzen für Entwicklung und Wartung.

### Unternehmensprofil

Diese Arbeit wird in Kooperation mit der itestra GmbH ([www.itestra.de](http://www.itestra.de)) angeboten. Die itestra GmbH ist ein innovativer, unabhängiger und international tätiger Software-Dienstleister im Bereich unternehmenskritischer Prozesse, Systeme und Anwendungen. Zum Portfolio gehören die Analyse von Geschäftsprozessen, der Entwurf und die Realisierung leistungsfähiger Lösungen in Projekten sowie das strategische Management bereits vorhandener Softwaresysteme.

### Betreuung (itestra GmbH)

Jonathan Streit ([streit@itestra.de](mailto:streit@itestra.de))

### Betreuung (LMU)

Prof. François Bry, Yingding Wang