

# Praktikum Software-Engineering 1

Wintersemester 2014/2015 – Aufgabenblatt 3

---

Prof. Dr. Stefan Sarstedt <[stefan.sarstedt@haw-hamburg.de](mailto:stefan.sarstedt@haw-hamburg.de)>

Raum 1085, Tel. 040/42875-8434

Norbert Kasperczyk-Borgmann <[nkb@informatik.haw-hamburg.de](mailto:nkb@informatik.haw-hamburg.de)>

Raum 1189, Tel. 040/42875-8417

## Bearbeitungshinweise

- Es gelten dieselben Hinweise wie bei den letzten Praktikumsblättern.

## Aufgabe 3.1: Use Case und User Story

Formulieren Sie für SOLE zwei Anwenderinteraktionen mit dem System:

- a) Eine einfache Tätigkeit in Form einer **User Story**,
- b) sowie eine „komplexe“ Tätigkeit in Form eines **ausführlichen Use Cases** (Anwendungsfall). Dieser soll aus mehreren Schritten (Erfolgsfall) bestehen und auch Fehlerfälle enthalten. Siehe hierzu vorab auch den Hinweis in Aufgabe 3.2a).

## Aufgabe 3.2: Entwurf – Systemoperationen und Schnittstellen

- a) Definieren Sie für Ihren Use Case und Ihre User Story aus Aufgabe 3.1 entsprechende **Systemoperationen**. Verwenden Sie die **Schnittstellenkategorie „A“**. Es sollte mindestens eine „komplexe Systemoperation“ dabei sei, deren Implementierung eine Interaktion zwischen mindestens zwei Komponenten erfordert (falls Sie noch keine finden, passen Sie Ihren Use Case aus Aufgabe 3.1b) entsprechend an!).
- b) Entwerfen Sie die **Schnittstellen des Anwendungskerns**. Weisen Sie dazu Ihre Systemoperationen aus Aufgabenteil a) jeweils einer entsprechenden verantwortlichen Komponente zu. Definieren Sie dazu **Komponenten-Interfaces in Java** mit ausführlichen **Javadoc**-Angaben.
- c) Zeichnen Sie für Ihre „komplexe Systemoperation“ ein **UML Sequenzdiagramm**, welches die Interaktion zwischen den Komponenten grafisch darstellt. Die sich ergebene(n) neue(n) Operation(en) nehmen Sie in Ihre Interfaces (inkl. Javadoc) auf.

## Aufgabe 3.3: Implementierung und Test

**Implementieren** Sie die komplexe Systemoperation aus Aufgabe 3.2c). Hierzu

- a) **Implementieren Sie die entsprechende Komponente**, welche die komplexe Systemoperation anbietet, inklusive Datenbankzugriffen (mittels JDBC) in Java. Orientieren Sie sich an dem Fallbeispiel aus der Vorlesung. Die andere(n) Komponente(n), mit denen die Systemoperation interagiert, können Sie als **Dummy-Komponente(n)** implementieren.

*Hinweise zur Datenbank:* Sie können eine eigene Installation verwenden (bewährt hat sich z.B. MySQL) oder die vorhandene Oracle-Installation an der HAW. Informationen hierzu finden Sie unter <https://userdoc.informatik.haw-hamburg.de/doku.php#oracle>. Verwenden Sie als Server `ora14.informatik.haw-hamburg.de`.

- b) Schreiben Sie einen **Komponententest** (als JUnit-Test) für Ihre Systemoperation. Konfigurieren Sie Ihr „System“ wieder mittels Dependency Injection und befüllen Sie die Datenbank vor Testausführung mit Stammdaten, die Ihre Tests benötigen.

Viel Spaß!