

## Aufgabenblatt 5

### **Ziel dieses Blatts**

Nun wird es langsam Ernst mit der konkreten Arbeit an Design und Implementierung des eigenen Projekts. Nach den intensiven Vorarbeiten, die im Rahmen der ersten vier Aufgabenblätter zu absolvieren waren, sind Sie (hoffentlich) gut vertraut mit der Problematik Ihres Themas. In diesem Arbeitsblatt werden die Anforderungen an einen ersten Designentwurf Ihres Modells spezifiziert. Dieser Designentwurf ist zugleich Gegenstand der zweiten Review.

### **Definitions- und Entwurfsphase (Designentwurf)**

Dieses Thema war zentral in der Vorlesung Software Engineering im letzten Semester nimmt den meisten Raum im [1] ein, so dass Sie die einschlägigen Konzepte kennen. Da die zu verwendenden Architekturkonzepte weitgehend klar bzw. in der Themenstellung vorgegeben sind, liegt der Schwerpunkt dieser Etappe auf der Definitionsphase und dort genauer auf der Objekt-Orientierte Analyse (OOA) [1, LE 13].

#### **Aufgabe:**

Entwickeln Sie stufenweise und in mehreren Iterationsrunden das Modell Ihrer Anwendung.

Entwerfen Sie zunächst als **statisches Modell** Ihres Systems die Klassen, Attribute, Assoziationen und Vererbungsstrukturen sowie die Zuordnung der Klassen zu Paketen und halten Sie die Gründe für Ihre Entscheidungen als Teil der zu erstellenden Design-Beschreibung fest (so viel oder wenig detailliert, dass ein Neueinsteiger diese Entscheidungen nachvollziehen kann ohne sich in Details zu verlieren). Erstellen Sie dazu geeignete UML-Diagramme.

Entwickeln Sie auf dieser Basis das **dynamische Modell** Ihres Systems. Spielen Sie an Hand geeigneter Anwendungsfälle und Szenarios alle wichtigen Interaktionskonstellationen durch, identifizieren Sie die Methoden, welche die einzelnen Klassen zur Verfügung stellen müssen und überlegen Sie, wie sich der Fluss der Botschaften durch das System gestaltet. Erstellen Sie dazu geeignete UML-Diagramme und ergänzen Sie die Design-Beschreibung entsprechend.

Beachten Sie, dass die beiden Modellierungsphasen im Sinne eines evolutionären, iterativen Entwicklungsprozesses mehrfach durchlaufen werden (siehe auch die Ausführungen zur Implementierung) und dass den Phasen unterschiedliches Gewicht zukommt<sup>1</sup>. Aktualisieren Sie den Designentwurf im Rhythmus der Wochenscheiben (siehe unten).

**Abzugeben:** Eine erste Version der Design-Beschreibung (im PDF-Format), aus welcher alle wichtigen Aspekte sowohl des statischen als auch des dynamischen Modells der ersten Iteration Ihres Systems ersichtlich sind. Halten Sie sich dazu an die im Blatt 3 angegebene Gliederung einer Design-Beschreibung mit dem Schwerpunkt auf den Abschnitten 3 und 4. Nutzen Sie sowohl textuelle als auch grafische Elemente (UML-Diagramme geeigneten Typs), um die Logik Ihres Systems darzustellen.

<sup>1</sup> „Die entstehende Spezifikation besteht aus einem statischen und einem dynamischen Modell. Welches dieser beiden Modelle in der Systemanalyse das größere Gewicht besitzt, hängt wesentlich von der jeweiligen Anwendung ab. Das statische Modell ist bei Datenbank Anwendungen besonders wichtig. Das dynamische Modell ist bei stark interaktiven Systemen sowie bei technischen Systemen von besonderer Bedeutung.“ [1, S. 378]

## Testkonzept

Als ein Maß für die Qualität der ausgeführten Implementierung hat es sich bewährt, die einzelnen Komponenten sowie das Gesamtsystem ausführlichen Tests zu unterziehen. Diese beiden Aspekte, *Komponententests* und *Systemtests*, sind bereits im Zuge der Modellierung vorzubereiten, um alle „Ecken“ des Systems auch wirklich zu erwischen.

Für die **Komponententests** stehen mit Unit-Frameworks (siehe Aufgabenblatt 3) Konzepte und Werkzeuge zur Verfügung, mit denen derartige Tests weitgehend automatisiert durchgeführt werden können. Es ist wichtig, die auszuführenden Testreihen genau zu planen, um die Gewähr zu haben, dass bei späteren Modifikationen am Design bzw. Quelltext alle explizit und implizit bestehenden Abhängigkeiten und Voraussetzungen auch weiterhin berücksichtigt sind.

### Aufgabe:

Komponenten- und Systemtests sind fester Bestandteil des Integrationsprozesses einzelner Programmteile in das Gesamtsystem. Deren Planung, Vorbereitung und Ausführung koordiniert die/der **Verantwortliche für Tests** Ihrer Projektgruppe. Dazu ist ein **Testkonzept** zu erstellen, das entsprechende konzeptionelle (Was wird wie getestet? Wo kommen die Testdaten her?) und organisatorische (Wer macht was und wie ist der Ablauf? Wie werden durchgeführte Tests mit Ihren Resultaten festgehalten?) Festlegungen fixiert. Erstellen Sie außerdem innerhalb Ihres Code-Projekts ein eigenes Package `test`, in der Sie alle Testbeispiele aufsammeln, damit sich diese einfach vom restlichen Quelltext trennen lassen. Orientieren Sie sich hierzu an den Vorgaben von Maven (java-basiertes Build Management Tool; <https://maven.apache.org/>).

**Abzugeben:** Das Testkonzept, das ein Antwort auf obigen Fragen (was, wer, usw., organisatorisch UND technisch) liefert und als Anleitung für das eigene Team dient.

## Ablauf der zweiten Review

**Gegenstand des zweiten Reviews** ist der aktuelle Stand der Arbeiten an Ihrem Projekt. Grundlagen des Reviews sind die mit diesem Aufgabenblatt abzugebende erste Version der Design-Beschreibung sowie die Test- und Dokumentationskonzepte.

Voraussetzung für das **Review** ist die termingerechte Abgabe der zu begutachtenden Materialien (Designbeschreibung, Testkonzept, evtl. überarbeitetes Dokumentationskonzept)

Das Review läuft wieder wie im Merkblatt „Review“ beschrieben ab (20 Min. Präsentation, 20 Min. Diskussion, 5 Min. Auswertung) und findet zum vereinbarten Termin statt.

**Vergessen Sie die aktuelle Aufwandsanalyse und Aktualisierung Ihrer Webseiten nicht!**

## Literaturverzeichnis

1. Balzert, Helmut; Liggesmeyer, Peter (2011): Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb. 3. Aufl. Heidelberg: Spektrum Akademischer Verlag (Lehrbücher der Informatik).