# Aufgabe 5: Entwurf 1

## Fragen

5.2.1)

Beschreiben Sie in eigenen Worten den Unterschied zwischen Architektur (Grobentwurf) und Design (Feinentwurf). Warum unterscheidet man zwischen diesen beiden?

Unter Systemarchitektur versteht man eine strukturierte und hierarchische Anordnung der Systemkomponenten sowie Beschreibung ihrer Beziehungen. Der Feinentwurf beschreibt dann bereits die Spezifikation der Algorithmen und Datenstrukturen um diese Komponenten und Beziehungen umzusetzen. Man unterscheidet diese beiden Stufen, um klar zwischen Funktionalität und Umsetzung zu unterscheiden. Gerade um zu klären, ob alle Komponenten enthalten sind, sind die Informationen des Designs bereits viel zu feingranular.

5.2.2)

Beschreiben Sie mit eigenen Worten den Unterschied zwischen den folgenden Begriffen; geben Sie jeweils ein Ihnen bekanntes Beispiel dafür an: Komponente, Paket, Framework

* Komponenten:

Eine Komponente ist eine Spezialisierung einer Klasse und ein modularer Teil eines Softwaresystems. Sie besitzt Strukturmerkmale (wie z.B. Attribute und Funktionen) und kann mit anderen Komponenten in Beziehung gesetzt werden.

*Beispiel:* Bei einem Emailsystem kann der Emaileingang eine eigene Komponente darstellen.

* Paket:
* Pakete helfen Komponente zusammenzufassen und zu strukturieren. Dabei kommen Komponente die aufgrund ihrer Eigenschaften zusammenwirken in dasselbe Paket.

*Beispiel:* Es können bei einem Emailsystem die Klassen für Emailausgang und Emailschreiben in ein Paket gelegt werden.

* Framework:

Ein Frameworks ist ein Paket aus verschiedenen, kooperierenden Klassen, die anpassbar sind und eine Vorlage zur Nutzung in einem bestimmten Anwendungsfall sind.

*Beispiel:* Das JUnit-Framework stellt Standardfunktionen zur Testung von in Java geschriebenen Programmen bereit, die für den konkreten Testfall angepasst werden können.

5.2.3)

Warum sind verschiedene Sichten auf eine Architektur sinnvoll? Sehen Sie Nachteile mehrerer Sichten? Falls ja, welche?

Verschiedene Sichten auf eine Architektur sind weitestgehend sinnvoll da somit die Architektur durch verschiedene „Brillen“ gesehen wird und somit Schwachstellen an verschiedenen Stellen identifiziert werden können. Dadurch kann eine Architektur verbessert werden und durchdachter sein.

Jedoch gibt es auch Nachteile. Man kann sich so leicht im Detail verlieren und damit das Projekt stark verzögern.

5.2.4)

Was fanden Sie schwierig, oder haben Sie nicht verstanden? Es mag sein, dass Sie hier nichts angeben können. Dann antworten Sie bitte mit „nichts“.

5.2.5)

Beschreiben Sie, was Sie am interessantesten oder gewinnbringend fanden.

5.2.6)

Welche Anknüpfungspunkte sehen Sie zwischen diesem Stoff und dem, was Sie bereits wissen?

5.2.7)

Wie lange haben Sie gelesen, Fragen beantwortet, Aufgaben bearbeitet? Gefragt ist jeweils der Gesamtaufwand aller Gruppenmitglieder.

## Aufgaben

5.3.1)

Eine Architektur-Beschreibung soll unter anderem

* relevant
* effizient pflegbar
* verständlich und nachvollziehbar

sein: Was bedeutet das? Geben Sie für jedes Merkmal ein positives und ein negatives Beispiel an.

5.3.2)

Vergleichen Sie Abbildung 1 „Systemkontext notiert in UML“ des Textes von Stefan Zörner mit der bisher besprochenen Form der Kontextdiagramme (vgl. Balzert, Abschnitt 16.3 „Kontext und Überblick“; vgl. Foliensatz im Wiki) und bewerten Sie kritisch: Was kommt von der RE-Sicht zur Architektursicht hinzu? Fehlen Ihnen in Abbildung 1 (weitere) Informationen (welche)? Sollte es ein gemeinsames Kontextdiagramm für Requirements Engineering und Architekturentwicklung oder zwei verschiedene geben? Begründen Sie Ihre Meinung kurz.

5.3.3)

Überlegen Sie sich eine Software-Architektur für das System zur Organisation von Tischtennisturnieren und zeichnen Sie ein UML-Verteilungsdiagramm dafür. Sie dürfen das Diagramm (korrekt!) per Hand zeichnen oder ein UML-Tool Ihrer Wahl verwenden, das die UML-Elemente jedoch korrekt darstellen muss.