Abgabe: Aufgabe 5 ST WS 14/15
HS Mannheim – Studiengang Informatik
Sebastian Schuler (1332256)
Thomas Kämmerling (1320565)

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“.

Nichts.

5.2.5)

Beschreiben Sie, was Sie am interessantesten oder gewinnbringend fanden.

Es ist gut die Definitionen der Begriffe einmal erarbeitet und sich darüber Gedanken gemacht zu haben. Vor allem die Unterscheidung zwischen Grob- und Feinentwurf wird in Projekten sicher Erleichterung bringen

5.2.6)

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

Aus unserer Sicht ist nun eine bessere Strukturierung der einzelnen Begriffe vorhanden was beim planen eines Projekts ein Vorteil bringt.

5.2.7)

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

6 Stunden.

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.

- Relevant: Eine Beschreibung ist relevant, wenn das beschrieben wird, was beschrieben werden soll. In einer Architekturbeschreibung soll also nur der Grobentwurf beschrieben sein. Positiv wäre es hier, wenn man sich an die Vorgaben zu Erstellung einer Architektur hält, also nur die in Frage 1 gezeigten Merkmale auch beschreibt. Negativ wäre es hier, bereits Datenstrukturen zu skizzieren oder aber Inhalte zu beschreiben die nicht zur Architektur, sondern zum Kontext gehören.
- Effizient pflegbar: Dies beschreibt, dass eine Architektur einfach zu aktualisieren und eben zu pflegen ist. Sprich, dass man einfach nachträglich Änderungen vornehmen kann oder aber die Architektur bereinigen. Wenig effizient pflegbar ist beispielsweise eine schlecht dokumentierte Architektur oder eine, die gar nicht aktiv geplant wurde. Gut ist es hingegen, wenn eine Architektur transparent dokumentiert ist.
- Verständlich und nachvollziehbar: Eine verständlich und nachvollziehbar, wenn auch dritte, die nicht an der Erstellung beteiligt waren die Architektur eigenständig bzw. nur mit wenigen Erklärungen verstehen können. Negativbeispiel hierzu wäre ein einfaches

Architekturdiagramm mit kryptischen Abkürzungen ohne Dokumentation. Positiv ist eine ausführlich dokumentierte Architektur, die einfach zu verstehen ist.

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.

Hinzu kommt in der Architektursicht die systemseitige Umsetzung. Das bedeutet, dass Systembausteine aufgeteilt werden und nicht mehr nur einfach eine Blackbox „System" zum Kontext gehört, sondern ganz konkrete Komponenten wie Mail-Server, Datenbanken usw. Außerdem werden hier schon Services eingebunden, wie SOAP oder JavaMail. Die Architektursicht ist also technischer als die Sicht des RE. Außerdem kommen bereits Kardinalitäten hinzu, sprich wie viele Nutzer zugreifen können.

In Abbildung 1 erscheint der Punkt „TicketSystem" zu allgemein. Diesen sollte man noch weiter in Unterkomponenten aufteilen. Außerdem könnte man auch hier einfügen, die dieser mit dem Ticketshop interagiert.

Die Trennung der beiden Kontextdiagramme ist sinnvoll. Dadurch hat man auf der RE-Ebene eine Sicht, die nicht zu sehr auf die Technik schaut, sondern sich auf Anforderungen und Stakeholder konzentriert. Man wird nicht in technische Details gezogen, die das RE einschränken. In der Architektur, die auf den Anforderungen beruht kann man dann die Anforderungen in eine technische Umsetzung übersetzen.

Abgabe: Aufgabe 5 ST WS 14/15
HS Mannheim – Studiengang Informatik
Sebastian Schuler (1332256)
Thomas Kämmerling (1320565)

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.