1794 & 1798
Masterprüfung (Master of Comp. Sc.)
zu den Kursen Softwareengineering II & Softwarearchitektur
Prüfer: Prof. Six
Februar 2006

Wie ist eine HTTP-URL aufgebaut? GET-Mechansimus?
Was gibt es noch ? (POST...). Wann ist GET, wann POST besser?

Was sind Sessions? Wozu braucht man Sessions? Wie kann man sie technisch realisieren?

Was ist J2EE? Aus welchen Teilen besteht es? (Anmerkung von Prof. Six als ich sagte dass J2EE <u>auch</u> für Webapplikationen geeignet sei: es ist <u>besonders</u> für Webapplikationen geeignet).

MVC erklären, wo wird es im Zusammenhang mit J2EE verwendet (Model2-Architektur).

Welche Mechanismen gibt es, um Dynamik in JSPs zu ermöglichen?
Wie werden Daten zwischen Servlets und JSPs transportiert?
Es liegen spezielle Objekte im Scope, nämlich?
Dann kurze Diskussion ob der Scope zum Anwendungskern gehört - in der KE 2 steht (Model2-Architecture, Kursversion 10/05), daß in der Literatur meist zum Anwendungskern gerechnet. Das habe ich auch so formuliert, er hat meine Skepsis wohl bemerkt und gemeint, daß man das auch anders sehen kann.

Definition / Beschreibung von Softwarearchitektur? (Komponenten und Interaktion, globale Festlegung). Nach welchen Gesichtspunkten kann man Architekturkomponenten definieren?

Wie unterscheiden sich Komponenten in Architektur und Entwurf?
(Architekturkomponenten = globale Gliederung, Entwurf folgt der von der Architektur vorgegebenen Gliederung, Entwurfskomponenten passen sich in die Architektur ein).

Schichtenarchitektur, wozu, Vorteile? (Er meinte dazu noch, die schöne Idee mit der lediglich lokalen Anpassung einzelner Komponenten klappt in der Praxis nicht immer so ganz).

Kann bei Schichtenarchitektur eine Schicht auch mit anderen Schichten als der direkt darunterliegenden interagieren, ohne daß das schlechtes SoftwEng. ist?
(Ja, siehe 5Schicht-Modell...).
Wieso gibt es eigentlich eine Richtung bei der Interaktion, die aufgerufenen Komponenten antworten doch? (Methodenaufruf, im Sinne von Client-Server gibt es eine Richtung).

Transformationen hin zum Entwurfsklassenmodell?
Hier wollte er als Einstieg zunächst mal als die Eigenschaften der **entwurfs**orientierten Aktivitätsdiagramme. Ich habe ihn missverstanden (ich glaube, weil sagte, er wolle über die anforderungsorientierten Aktivitätsdiagramme nichts wissen). Ich wusste jedenfalls nicht so recht, worauf er hinaus wollte. Habe etwas verworren erklärt, was ich alles über erweiterte / anforderungsorientierte / entwurfsorientierte Aktivitätsdiagramm wusste (Szenen, Sterotype...). Irgendwann fiel der Groschen und ich konnte die Eigenschaften der entwurfsorientierten ADiagr. schildern. (Subaktivitäten als wiederverwendbare Kompontenten. Auf jede Akteuraktion folgt immer eine Systemaktion, die Umkehrung dazu gilt aber nicht, usw.).
Dann war die Zeit zuende, als nächstes wären wohl die 6 Trafos hin zum Entwurfsklassenmodell dran gewesen.

Fazit:
Ein fairer und beruhigender Prüfer, der das Thema SoftwareEngineering nicht nur aus der Theorie kennt.
Er will sichergehen, daß der Kandidat lebendiges Wissen und echtes Verständnis für den Stoff hat. Dafür verzeiht er kleinere Mißverständnisse oder wenn im Eifer des Gefechtes mal ein Versprecher vorkommt („Aktion“ und „Aktivität“ verwechselt...).

Man sollte Grundkenntnisse in OOP haben (nur Prinzipien notwendig, keine Details) und vielleicht ein bisschen was von „Formaten im Internet“ verstehen. Software Engineering I verstanden zu haben empfiehlt sich.

Konzeptpapier lag vor mir auf dem Tisch war aber nicht notwendig (rein mündliche Prüfung).

Empfehlung: Bei Unklarheiten sofort und höflich nachfragen (wollen Sie auf... hinaus?). Wenn der Stoff verstanden ist kann einem eigentlich nicht viel passieren (ausser einer guten Note).
Er meinte allerdings, daß er demnächst die Maßstäbe etwas anheben wird (ich war an der Kippe zwischen zwei Noten und das hätte mich beinahe eine drittel Note gekostet, womit ich mich immer noch fair behandelt gefühlt hätte).

1794 & 1798
Masterprüfung zu den Kursen Softwareengineering II & Softwarearchitektur
Prüfer: Prof. Six
Januar 2006

Allgemein:
Die Prüfungsatmosphäre ist sehr entspannt, eher wie ein normales Gespräch. Prof. Six erzählt zwischendurch auch ein bißchen was, um die Fragen bzw. die Antworten in einen größeren Rahmen zu setzen. Im Gegensatz zu anderen Prüfungen bekommt man nichts zu schreiben, d.h. man muss alles erzählen. Prof. Six prüft eher in die Breite als in die Tiefe und konzentriert sich sehr stark auf seinen eigenen Kurs (Softwareengineering II), es gab keine direkten Fragen zum Stoff von Softwarearchitektur (die Fragen zur Architektur basieren eher auf dem Architektur-Teilen des Kurses SW-Engineering II).

Fragen:

Wie ist eine URI aufgebaut (allgemein)? Wie ist speziell eine http-URI aufgebaut? Wie sieht der Query-Teil aus? Wozu braucht man den? (Zum übertragen von Formulardaten) Wie kann man noch Formulardaten übertragen? (bei POST im http-Body) Vorteile / Nachteile von GET und POST?

Was sind Sessions? Wozu braucht man Sessions? Wie kann man sie technisch realisieren?

Was ist J2EE? Aus welchen Teilen besteht es (sowohl logisch als auch technisch, er wollte zum einen auf JSPs, Servlets & EJBs, zum anderen auf Browser und Applicationserver hinaus)?

Wenn man sich JSPs, EJBs und Servlets so ansieht, welches Muster lässt sich hier in der Architektur erkennen? (MVC, M=EJBs, V=JSPs, C=Servlets)

Welchen Grund gibt es noch (außer dem Pattern), JSPs und Servlets zu trennen? (Unterschiedliche Entwickler, JSPs = WebDesigner, Servlets = Javaentwickler)

Welche Mechanismen gibt es, um Dynamik in JSPs zu ermöglichen?

Warum sind Scriptlets nicht wünschenswert? (Führen leicht zu Model-1-Architekturen, d.h. verwischen die Grenzen zwischen V und C)

Wie werden Daten zwischen Servlets und JSPs transportiert? (Scopes)

Es liegen spezielle Objekte im Scope, nämlich? (JavaBeans)

Was macht eine JavaBean aus?

Definition / Beschreibung von Softwarearchitektur? (Ihm war hier insbesondere der Begriff Komponente wichtig, und dass die Summe der Komponenten dann die Architektur bildet)

Wie unterscheiden sich Komponenten in Architektur und Design? (Weitgehend Granularität, ich hab dann noch ein bißchen was zu Komponentenframeworks erzählt)

Was sind Aktivitätsdiagramme? Wofür kann man sie benutzen?

Welche Erweiterungen gibt es für Webapplikationen?

Kurs Software Engineering II (1794) Fassung vom SS 2005, Software Architektur 1798
Fachprüfung Master of Comp. Science

Prüfer: Prof. Dr. Six

- Was steckt hinter J2EE
- Was macht ein Servlet (Entgegennahme von Requests)
- Was macht eine JSP
- Wird für eine JSP unbedingt Java benötigt (hier wollte Prof. Six auf die EL hinaus)
- Reicht EL aus -> Taglibs
- Wann muss das Servlet auf das Model zugreifen
- Gibt es Möglichkeiten, ohne Modelzugriffe auszukommen
- Wie können Szenen bei akteurorientierten Aktivitätsdiagrammen zusammengefasst werden
- Wie grenzt sich der Entwurf von der Architektur ab (hier ging es dann auch etwas ins Detail)
- Muss sich die Architektur dem Entwurf anpassen oder umgekehrt (Architektur ist das globale)
- Was macht eine gute Architektur aus (-> Nennung der zwei Punkte aus dem Kurs)
- Struts kam nicht direkt dran, habe aber trotzdem das fünfstufige Modell zur Implementierung einer Action eingestreut

Fazit:
Am Anfang hat Prof. Six etwas Freiraum gelassen, um zu sehen, wie die Materie sitzt. Im Laufe der Prüfung kamen konkrete Fragen und auf diese Fragen wollte Prof. Six auch eine kurze und konkrete Antwort. Insgesamt kann ich Prof. Six als Prüfer empfehlen. Wobei er auch auf die theoretischen Grundlagen und Hintergründe Wert legt.