

Prüfungsprotokoll Informatik Praxis

Prüfer: Professor Lichter und Professor Borchers

- OOSK: Objektorientierte Softwarekonstruktion (V3: WS 2005/2006),
- SQS: Software-Qualitätssicherung (V3: SS 2006),
- DIS I: Designing Interactive Systems I (V4:WS 2005/2006),
- CTHCI: Current Topics in Human Computer Interaction (V2:SS 2006).

Note: 2,0

Dauer: 55 Minuten

Info

Ich bin recht kurzfristig auf den Trichter gekommen (übrigens durch ein Prüfungsprotokoll ;), dass man die Veranstaltungen von Prof. Lichter und Prof. Borchers für die Praxisprüfung miteinander kombinieren kann. Die Kombination der Fächer und Profs kann ich auch nach der Prüfung noch empfehlen. Die Prüfungssituation ist ganz angenehm. Beide Profs geben Hilfestellungen, wenn man sich mal festgefahren hat.

Prof. Lichter hat im Vergleich zu anderen Prüfungen einen wilden Ritt durch die verschiedenen Fachgebiete unternommen, angefangen bei der Softwaretechnik, über OOSK hin zu SQS. Die „Standardfragen“ kamen praktisch gar nicht. Prof. Lichter fragt grundsätzlich eher nach Zusammenhängen und stichprobenartig. Falls man etwas zu einem Thema nicht weiss, bekommt man in anderen Bereichen eine Chance.

Die Themen von Prof. Borchers kamen eher wie erwartet. Meiner Meinung nach lässt sich aus den Vorlesungen auch nur ein beschränkter Teil gut prüfen. Wie in anderen Protokollen bereits angedeutet, geht Prof. Borchers vor allem bei den mathematischen Modellen bis ins kleinste Detail, um zu erfahren, ob man die Hintergründe vollständig verstanden hat. Ich weiss nicht, ob das so für eine Praxisprüfung dringend notwendig ist, jedenfalls empfehle ich, u.a. diese Bereiche genau vorzubereiten (siehe unten).

Bei der Benotung hatte ich übrigens den Eindruck, dass Prof. Borchers mich besser bewertet hat, als ich mich selbst eingeschätzt hatte. Bei Prof. Lichter war dies anscheinend genau umgekehrt. ;) Am Ende war ich mit der Note jedenfalls absolut zufrieden.

Material und Vorbereitung

Der Stoff aller vier Vorlesungen wird in guter digitaler Form angeboten: Den Stoff von Prof. Lichter habe ich nach den Folien gelernt, die gut strukturiert und vollständig sind. Nicht alles aus diesen Folien ist relevant, aber das findet Ihr schnell selbst heraus.

Die Folien von Prof. Borchers sind eher zum Überblick zu gebrauchen. Ich habe vor allem aus den Vorlesungsvideos gelernt und einige Themen mit den weiteren Quellen vertieft.

Ich hatte DIS und SQS bereits teilweise gehört. Für die Prüfung habe ich noch einmal ca. 4 Wochen gelernt, davon ca. 2 Wochen „Hardcore“. Für eine bessere Note sollte man vielleicht noch mehr Aufwand investieren. Für mich war das so OK – ich bin Mechaniker, und kein Doktor. ;)

Tips

Vor allem für Professor Lichter gilt, dass man den Verlauf der Prüfung immer auch selbst mitbestimmt. Seine Fragen kommen in der Regel aus dem „Katalog“ von Antworten, die man vorher gegeben hat. Falls ihr also etwas erwähnt, solltet ihr auch darauf vorbereitet sein, es genauer erklären zu können.

Prof. Borchers möchte zu ausgewählten Themen alle Details wissen. Das oft gehörte „Werkzeugkasten-Prinzip“ konnte ich in meiner Prüfung nicht erkennen (wenn man davon absieht, dass ich meine Wasserflasche mit Linearer Algebra analysieren sollte). Ich empfehle, alle Modelle (und hier vor allem die mathematischen) sorgfältig vorzubereiten. Man sollte nicht nur die Hintergründe genau erklären, sondern auch anwenden können. Einiges davon wird in der Vorlesung gar nicht erläutert, oder von einem Assi in ein paar Minuten durchgewunken. Hier sollte man zu den Vorlesungsvideos die entsprechenden Quellen heranziehen.

Konkret (mindestens):

- Fitts'/Hick's Law und Varianten: Unterschiede, Anwendung, Details,
- CMN Model Human Processor: Details bis hin zu Decay- und Processing-Zeiten, Hintergründe (Bloch's Law, Penstroke-Test),
- GOMS: Unterschiede der 4 Varianten, KLM mit M-Regeln anwenden können,
- Matrizen (nur CTHCI): Hintergründe, Anwendungsmöglichkeiten (z.B. Undo, Beschreibungsmöglichkeit von Interaktionssequenzen, usw.), an Beispielen anwenden können.

Eine kleine Bemerkung

Ich habe im Gespräch mit anderen Studenten oft den Eindruck gehabt, dass alles andere als eine 1,x in einer Diplomprüfung als eine miserable Note gilt (vor allem, wenn man Prüfungsprotokolle liest, wo Leute eigene Vorlesungsskripte erstellen und dem Professor in der Prüfung noch einige Neuigkeiten beibringen). Ich habe das mal das „Thomas D. Prinzip“ getauft ;). Ich habe Respekt vor diesen Menschen und finde das Engagement natürlich gut. Ich denke aber auch, dass die Informatik-Welt nicht nur aus solchen Leuten besteht. Das gilt auch für die Prüfungsprotokolle. Bitte macht daher ein eigenes Protokoll eurer Diplomprüfungen, auch wenn diese vielleicht nicht sehr gut verlaufen ist.

Vielen Dank und viel Erfolg!

Professor Borchers (ca. 30 Minuten)

Wir hatten ja verschiedene Modelle. Können Sie die mal der Reihe nach aufzählen, vielleicht sortiert von stark mathematisch/formal bis hin zu wenig mathematisch/formal?

1. Fitts/Welford/Shannon/Hick's Law,
2. CMN Model Human Processor,
3. GOMS.

OK, wozu braucht man denn Fitts' Law & Co.?

Erklärt, dabei auch auf Unterschiede und Vorzüge der verschiedenen Varianten eingegangen. Hick's Law mit erwähnt.

Was macht man denn mit Hick's Law?

Auswahl aus mehreren Items. Zwei Varianten für Gleichverteilung und ungleichmäßige Verteilung der Items. Formeln freiwillig erwähnt bzw. aufgeschrieben.

Erklären Sie doch mal CMN Human Model Processor.

Erläutert und aufgemalt. Da der Midterm Test in DIS kurz vorher bereits im Detail auf die verschiedenen Processing und Decay Times eingegangen war, hatte ich diese auch hier parat.

Was wird denn länger im Short Term Memory gespeichert, Bilder oder Töne?

Definitiv falsche Antwort, konsequent falsch erklärt.

Wissen Sie, warum man Töne länger speichern kann als Bilder?

Auch hier habe ich mich verhaspelt, aber irgendwie auch was richtiges gesagt.

Wie kommt man denn auf das Modell des Human Processors?

Mir fiel nur der Penstroke Test ein.

Was ist denn der Penstroke-Test?

Erklärt, dabei zunächst nicht auf den „Wahrnehmungs-Anteil“ in den 240 ms eingegangen. Wichtig ist hier, dass man auf die Korrekturen der Overshootings eingeht, d.h. die Differenz zwischen den verschiedenen Peaks ist von großer Bedeutung.

Sagt Ihnen Bloch's Law etwas?

Gehört hatte ich davon im Kontext zum Human Processor. In der Vorlesung waren wir aber nicht im Detail darauf eingegangen. Viel sinnvolles konnte ich dazu nicht wiedergeben.

Was ist GOMS? Welche Varianten gibt es?

CMN, KLM, CPM, NGOMSL und EPIC aufgezählt und individuelle Unterschiede erläutert.

Was ist denn das besondere an CPM-GOMS?

Parallele Tasks lassen sich abbilden, Overlapping, Beispiel mit Jetfighter-Pilot gebracht.

Was sind denn die 7 Stages of Action? Können Sie die mal an einem Beispiel durchspielen?

Gulfs of Execution/Evaluation aufgezählt und angewendet. Dabei auf die Schwierigkeiten eingegangen, mit denen man einige der Schritte voneinander unterscheiden kann.

Was sind Affordances und Constraints? Können Sie Constraints einmal an einem Beispiel beschreiben?

Constraints sind das Gegenteil von Affordances. Beispiel für physikalische/ semantische/ logische Constraints gegeben.

Wir hatten ja auch Matrizen zur Modellierung von Interaktion. Wo würden Sie denn dieses Modell auf der Mathe-Skala einordnen?

Top 1: Lineare Algebra at its finest.

(Ich hatte eine Wasserflasche dabei, die oben eine Ziehverschluss hatte).

Stellen Sie sich einmal vor, der Verschluss Ihrer Wasserflasche hätte drei Zustände: Zu, halb-offen und auf (in dieser Reihenfolge). Können Sie die drei „Tasten“ und die gesamte Funktion des Verschlusses einmal als Matrix modellieren?

Gemacht. Null Fehler.

Was kann man denn mit Hilfe von Matrizen so alles zeigen in Bezug auf Interaktionsmöglichkeiten?

Einige Highlights erwähnt, wie Existenz einer Undo-Funktion, Abbildung von Interaktionssequenzen, usw.

Nun stellen Sie sich einmal vor, die Wasserflasche hätte noch einen Lichtschalter mit den Zuständen an/aus. Wie müssten Sie Ihre Matrix erweitern, um alle Interaktionsmöglichkeiten abzubilden?

(!?) Das habe ich nicht richtig hinbekommen und wir haben dann irgendwann abgebrochen. Richtig war, dass die neue Matrix zwei Mal diagonal in einer neuen Gesamtmatrix auftaucht. Beispiele dazu gibt es im Vorlesungsvideo und im entsprechenden Paper.

Professor Lichter (ca. 25 Minuten)

Wenn Sie Software entwickeln wollen, welche Tätigkeiten müssen Sie denn da so durchführen?

Verschiedene Teile erwähnt, u.a. PM, CM, CRM, Requirements Engineering, etc.

Was macht man denn im Requirements Engineering genau?

Anforderungen erheben und verwalten, Abgleich zwischen Anforderungen und Ergebnissen. Ganz vollständig war die Antwort scheinbar nicht.

Jetzt habe ich meine Anforderungsspezifikation, wie komme ich denn da zu meinem Entwurf?

Reviews, Tiefenstruktur mit Hilfe linguistischer Methoden ermitteln, Formalisieren, z.B. mit UML (er wollte auf die Linguistik hinaus).

Können Sie mal ein paar linguistische Regeln nennen?

Ich kannte vor allem die drei Kategorien: Tilgungen, Generalisierungen, Verzerrungen. Er wollte Beispiele zu den Tilgungen hören, ich habe ihm stattdessen etwas zu den Generalisierungen angeboten, damit war er zufrieden.

Was gibt es denn so für Elemente im OO-Metamodell?

Klassen, Objekte, Beziehungen, Entwurfseinheiten (Klassenbibliotheken, Rahmenwerke, Cluster, Schichten).

Wo Sie gerade Schichten sagen: Was ist denn das und wie funktioniert das?

Sammlung logisch zusammenhängender Einheiten, z.B. Klassen, Cluster, Bibliotheken, etc.

Was sind die Vor- und Nachteile von Schichten?

Logische Trennung, dadurch eingeschränkte Kommunikationsmöglichkeiten über mehrere Schichten. Da fehlte ihm noch etwas in der Antwort.

Was kann man denn mit den Begriffen Kohäsion und Kopplung in Bezug auf Schichten anfangen?

Da wusste ich erst einmal nicht viel zu. Ich habe die beiden Prinzipien dann einfach von Klassen auf Schichten übertragen, das war scheinbar ganz OK.

Wie kommuniziert man denn in Schichten?

Immer nur von einer tieferen zu einer nächsthöheren Schicht.

Und wenn Sie von der tiefsten zu der höchsten Schicht kommunizieren möchten?

Dann muss man sich durch die Schichten „hangeln“.

(Hier kam der Shift zu SQS).

Wenn Sie sich jetzt mal vorstellen, Sie haben alles zur Verfügung, was man zum Testen braucht (Spezifikation, Sourcecode, ..). Wie würden Sie denn da vorgehen?

Ich habe zunächst Reviews erwähnt und wollte dann auf einen Whiteboxtest hinaus, der war ihm aber zu teuer. Da blieb dann nicht viel übrig. ;) Blackboxtest beschrieben und auf die verschiedenen Überdeckungsprinzipien eingegangen.

Warum ist gerade die Funktionsüberdeckung so interessant?

Meine Antwort war nicht ganz zufriedenstellend. Es lief wohl darauf hinaus, dass es in der Softwaretechnik zwischen Kunde und Lieferant vor allem darum geht, dass die vereinbarten Funktionen auch laufen.

Wie komme ich denn jetzt zu meinen Testfällen?

Äquivalenzklassenbildung mit Repräsentanten beschrieben. Dabei einige kleine Fehler gemacht.

Angenommen, ich habe zu allen Funktionen Äquivalenzklassen gebildet und zu jeweils einem Repräsentanten getestet, so dass ich keine Fehler mehr finde. Kann ich dann sagen, dass das Programm fehlerfrei ist?

Nein, Programme sind nicht vollständig testbar. Erläutert, warum.

Können Sie mir Beispiele für Produkt- und Prozessmetriken geben?

Prozess: Dauer, Aufwand. Produkt: Wartbarkeit? Letzteres war ein mächtig falscher Ansatz, was ich auch direkt zu spüren bekam:

Nennen Sie mir doch einmal eine Produktmetrik zu Wartbarkeit/Testbarkeit.

Das war eine kleine Fangfrage. Es gibt nämlich keine. Zum Glück habe ich hier nicht geraten, sondern eher versucht zu erläutern, wie ich zu einer solchen Metrik kommen könnte. Jedenfalls war das eine Sackgasse.

Welche Komplexitätsmetriken gibt es denn?

Komplexitätsmetrik nach McCabe mit zyklomatischer Zahl erläutert. Ausserdem grob die OO-Metrik-Suite von Chidamer & Kemerer beschrieben und auf WMC eingegangen, wo auch die zyklomatische Zahl berechnet wird.