

# Gedächtnisprotokoll Diplomprüfung Praktische Informatik

Prüfungsinhalt: Einführung in die Softwaretechnik  
Software-Qualitätssicherung (SS 2002)  
Einführung in Datenbanken (Kemper/Eicker, Kap. 1-7, 13, 16, 18)  
E-Learning Konzepte & Systeme  
Prüfer: Lichter, Schroeder

Datum: 12.5.2004  
Prüfling: Lutz Ibler

Literatur: siehe unten!

*Herr Lichter hat gerade angerufen, er steckt im Stau und kommt ein paar Minuten später. Lassen Sie uns doch schonmal die formalen Sachen klären.*

(Das war gut, da das Erledigen der Formalitäten meine Nervosität gemindert hat. Als Lichter dann nach ein paar Minuten kam, hatten Schroeder und ich schon eine lockere Stimmung drauf. Vielleicht hilft das Leuten mit großer Prüfungsnervosität – man könnte sich nach Absprache ja ein wenig früher mit dem Professor treffen und schonmal locker einsteigen. Nur so eine Idee, die mir gerade gekommen ist.)

*Welche Reihenfolge wünschen Sie sich denn?*  
Keine Ahnung.

*Also antworten müssen Sie aber irgendwas.*

Gut, da Herr Prof. Schroeder gerade schon das Blatt vor sich liegen hat, würde ich spontan gerne mit Softwaretechnik anfangen.

## **Einführung in die Softwaretechnik**

*Was gehört denn so alles zur Softwaretechnik?*

Anforderungsanalyse, Systemmodellierung, Prozessmanagement, Projektmanagement, Qualitätssicherung...

*Sie haben Prozessmanagement gesagt. Was ist denn ein Prozess?*

Die Aneinanderreihung von Aktivitäten während der Softwareerstellung.

*Und unterscheidet sich das Prozessmanagement vom Projektmanagement?*

Ja, wesentlich! Projektmanagement ist die Verwaltung von Ressourcen, um ein Ziel innerhalb eines Zeit- und Kostenrahmens zu erreichen.

*Bleiben wir beim Prozess. Sie haben Prozessmodelle kennen gelernt. Welche gibt es da?*

Das Wasserfallmodell, das Spiralmodell, das Prototypenmodell, das Transformationsmodell...

*Wie sieht das Wasserfallmodell aus?*

(Wasserfallmodell erklärt. Erwähnt, dass man in frühere Phasen zurückspringen kann.)

*Nicht zurückspringen, das wäre ja ein Wandern in der Zeit. Mehr so Wiederaufnahmen von Tätigkeiten. Und was ist die Kritik am Wasserfallmodell?*

Dass es sehr dokumentenorientiert ist. Und dass es schwierig ist, einen Zeitplan einzuhalten, weil man immer wieder alte Tätigkeiten aufnehmen kann.

*Ja, aber vor allem, dass Fehler erst so spät entdeckt werden. Requirements Engineering, was ist das denn?*

Das Feststellen und Festschreiben von dem, was ein Kunde für ein Softwareprodukt haben will.

*Wie erheben Sie denn Anforderungen?*

(Einige Möglichkeiten erklärt: Interviews usw.)

*Was stellt man für Forderungen an Anforderungsdokumente?*

Verständlichkeit, Genauigkeit, Überprüfbarkeit, zur Kommunikation geeignet.

*Und natürlich die Standardanforderungen Vollständigkeit und Konsistenz. Es gibt funktionale und nichtfunktionale Anforderungen. Nennen Sie mal Beispiele für nichtfunktionale Anforderungen.*

Benutzerfreundlichkeit.

*Und?*

Zeitverhalten.

*Ja. Wie notieren Sie denn funktionale Anforderungen?*

Zum Beispiel mit einer algebraischen Notation.

*Da kommen Sie aber nur selten dazu, die anzuwenden.*

Dann mit Hilfe von Diagrammen.

*Das wäre schön. Haben Sie schonmal ein Anforderungsdokument erstellt?*

Ich würde sagen: ja.

*Und, wie haben Sie die Anforderungen da formuliert?*

–klick– Natürlichsprachlich!

*Genau!*

## **Software-Qualitätssicherung**

*Welche Maßnahmen zur Sicherung von Qualität gibt es?*

Konstruktive und analytische.

*Und was bedeutet das?*

Konstruktive Maßnahmen sorgen dafür, dass ein Qualitätsprodukt entsteht. Analytische messen, inwieweit Qualität besteht.

*Was ist denn Qualität?*

Die Erfüllung von vorher festgelegten Zielen.

*Ja. Aber analytische Maßnahmen sollen nicht messen, sondern Fehler finden. Das ist das eigentliche Ziel. – Eine Methode zum Sichern von Qualität ist das Software-Testen. Wir haben die Weyuker-Axiome kennen gelernt. Sagen Sie mal, was Ihnen zu Anti-Extensionalität, Anti-Komposition und Anti-Dekomposition einfällt. Fangen Sie mit Anti-Extensionalität an.*

(Erläutert: Semantisch gleiche Programme nicht mit gleicher Testfallmenge testbar.)

*Und warum kann man die nicht gleich testen?*

Wegen der möglicherweise unterschiedlichen Implementierungen.

*Beim Black-Box-Test ist das natürlich egal. Was ist mit den anderen beiden Axiomen?*

(Erläutert.)

*Zum Messen braucht man Metriken. Was ist eine Metrik?*

Eine Abbildung von Gegenständen auf eine Skala, plus eine Anleitung zur Interpretation.

*Sie kennen ein historisches Komplexitätsmaß. Welches ist das?*

Äh, da muss ich passen. Mir fiel jetzt nur ein objektorientiertes ein.

*Na gut, wie ist denn das objektorientierte?*

*Kopplung.*

Wie definiert man Kopplung?

(Kopplung erklärt.)

*Ja, genau. Und jetzt zum historischen Maß... McCabe, sagt Ihnen der was?*

Schwach... der orientiert sich an Ablaufdiagrammen?

*In etwa. (Erläutert McCabe.) – Sie haben objektorientierte Maße angesprochen. Sagt Ihnen Lack of Cohesion in Methods irgendwas, mit dem man den Zusammenhalt einer Klasse zu beschreiben versucht?*

Gehört habe ich das schonmal.

*Haben Sie schonmal eine Klasse implementiert? Woraus besteht die?*

Aus Variablen und Methoden, die die Variablen modifizieren.

*Genau. Und jetzt, wo könnte man da mit dem Zusammenhalt ansetzen?*

–klick– Wenn Methoden verzahnt auf die Variablen zugreifen, dann hat die Klasse einen hohen Zusammenhalt.

*Genau. Das ist dann ein Maß für die Komplexität der Klasse.*

*Schroeder: Echt?*

*Lichter: Na ja, das ist so eine Idee.*

## **Einführung in Datenbanken**

*Wozu braucht man denn Datenbanken?*

Zur einfachen Verwaltung von Informationen.

*Aber das könnten Sie doch auch alles mit Dateien machen.*

Ja, aber dann müsste sich jede Anwendung wieder neu um die Dateiverwaltung kümmern.

*Aber es gibt doch z.B. in Java Serialisierbarkeit?*

Ja. Damit kann man aber noch immer keine Sichtbarkeitsregeln definieren.

*Das kann man mit dem Dateisystem.*

Nur auf gesamte Dateien, nicht auf einzelne Informationen. Und das Dateisystem kann auch keine Mehrbenutzer-synchronisation, also z.B. konkurrierende Updates verwalten.

*Und noch was? Wenn Sie mehrere Operationen ausführen wollen, die Daten verändern?*

Dann braucht man Transaktionen.

*Wie wird das gehandhabt?*

*Transaktion gelten als atomare Operation, bis man sie abschließt.*

*Und wenn mehrere die gleiche Transaktion ausführen?*

Dann kann eine nicht ausgeführt werden, wenn sie mit der anderen konkurriert.

*Rollback ist da das Stichwort, genau. Eben haben Sie den Softwareentwurf beschrieben. Wo ist denn der Unterschied vom Softwareentwurf um Datenbankentwurf?*

RE usw. sind gleich. Das Ergebnis des Datenbankentwurfs ist ein konzeptuelles Modell.

*Wovon denn?*

Von Daten. Man erhält ein Datenbankschema. Bzw. Sichten, die dann konsolidiert werden müssen.

*Genau, also man modelliert beim Datenbankentwurf Datenstrukturen, weniger Abläufe. Was bietet sich denn da an, um so ein Schema zu notieren?*

ER, UML.

*Wo ist der Unterschied?*

UML ist mächtiger und modelliert Verhalten mit.

*Woraus besteht denn ein ER-Diagramm?*

Aus Entities, Relations und Attributen.

*Hier ist so ein ER-Diagramm. (Zeigt eine Kopie aus dem Buch.) Erläutern Sie das mal.*

(Erläutert.)

*Sagen Ihnen Multiplizitäten was?*

(1:1, 1:M, M:N erläutert.)

*Und es gibt da noch eine Notation, welche ist das?*

Die (min, max)-Notation.

*Wo ist denn der Unterschied zwischen den beiden Notationen? Sind die gleich mächtig?*

Nein, (min, max) ist ausdrucksstärker und je nach Anwendungsgebiet praktisch.

*Versuchen Sie mal, in diesem Beispiel Multiplizitäten einzutragen.*

(Getan.)

*Hier gibt es Professoren und Studenten. Das würde man ja normalerweise zusammenfassen. Wie denn?*

Mittels Vererbung. In ER mit is-a dargestellt.

*Ja, so ein sechseckiger Kasten. Warum macht man das hier nicht?*

Äh?

*In UML würde man das doch machen. Bei ER-Diagrammen kommt man so schnell nicht auf die Idee.*

Keine Ahnung, warum nicht. Vielleicht, weil man mit ER eher verschiedene Sichten modelliert und die Generalisierung erst bei der Konsolidierung einführt?

*Das könnte ein Grund sein. Wie übertragen Sie diesen Entwurf in ein relationales Schema?*

(Verfahren erläutert.)

*Was sind die Operatoren der relationalen Algebra?*

*(Erläutert.)*

*Und reichen diese aus?*

Ja, die sind relational vollständig.

*Es gibt aber noch weitere.*

Ja, den Join, die Division.

*Die Division, ja. Und den Join – was macht der denn?*

*(Join erklärt.)*

*Genau. Wie implementieren Sie denn den Join in SQL?*

Weiss nicht genau. Mir schwebt da was mit JOIN ON vor, aber die genaue Syntax kenne ich nicht.

*Also davon habe ich noch nie gehört. Vielleicht benutzen Sie da irgendeinen überirdischen SQL-Dialekt. Wie würden Sie das denn sonst machen?*

–klick– mit SELECT ... FROM ... WHERE Joinbedingung.

*Richtig. Erst das Kreuzprodukt im FROM und dann die Selektion im WHERE. Sie kennen Relationenkalküle – welche Varianten denn? Und sind die gleich mächtig wie die Algebra?*

Tupel- und Domänenkalkül. Beide sind gleich mächtig bei Beschränkung auf sichere Ausdrücke.

*Schnell noch was zu Normalformen. Welche kennen Sie und welche sind praxisrelevant?*

Erste bis vierte und BCNF. Praxisrelevant bis zur dritten, weil danach Abhängigkeiten verloren gehen können.

*Wozu macht man die Normalisierung?*

Zur Vermeidung von Anomalien. Es gibt Einfüge-, Lösch- und Aktualisierungsanomalien. Diese werden durch Redundanzen verursacht.

*Genau. Worauf muss man da achten, bei so einer Normalisierung? Insbesondere bei der Dekomposition? (Zeigt eine Kopie der verlustigen Kemper-Eickler-Biertrinker-Relationszerlegung.)*

Man muss auf die Verlustlosigkeit achten, die hier im Beispiel nicht gegeben ist. (Verlustlosigkeit erläutert.)

*Sagen Ihnen Integritätsbedingungen was?*

Es gibt Bedingungen auf Attributdomänen, also z.B. „nur Werte zwischen 1 und 10“ und Fremdschlüsselbeziehungen, wo man Aktualisierungen zum Teil über mehrere Relationen hinweg verfolgen muss.

*Referentielle Integrität heisst das.*

## **E-Learning Konzepte & Systeme**

*Im E-Learning haben wir uns mit verschiedenen Lerntheorien beschäftigt. Können Sie mir diese aufzählen?*

Behaviorismus, Kognitivismus und Konstruktivismus.

*Und wo liegen da die Unterschiede?*

Beim Behaviorismus wird der Mensch als etwas betrachtet, das auf Reize reagiert.

*Stichwort Konditionierung. Die steht im Vordergrund.*

Beim Kognitivismus wird Wissen als ein abstraktes Gebilde betrachtet, das der Mensch erforschen und als kognitive Struktur nachbilden muss.

*Das kann man so stehen lassen. Kennen Sie den Namen von jemanden, der an der Entstehung der Idee des Kognitivismus beteiligt war?*

Piaget.

*Herr Piaget hat zwei Ansätze entwickelt. Wissen Sie, welche?*

Ein Ansatz verfolgt die Idee der Akkomodation und Akkumulation. Akkomodation ist die Anpassung von Wissen an neue Situationen und Akkumulation ist die Aufnahme von neuem Wissen unter Berücksichtigung aktuellen Wissens.

*Das wäre ein Ansatz. Erläutern Sie doch noch den Konstruktivismus.*

Der geht davon aus, dass Wissen nicht strukturiert vorhanden ist, sondern vom Lernenden erst konstruiert werden muss.

*Sie kennen die Lernzieltaxonomie von Bloom. Wie wird da klassifiziert?*

Nach kognitiven, affektive und sensormotorische Zielen.

*Und was sind die kognitiven?*

Wissen, Verständnis, Anwendung, ...

*... (unterbricht) eine aufbauende Folge also. Wie bringen Sie das mit dem Konstruktivismus in Verbindung?*

Der Konstruktivismus setzt sich darüber hinweg, indem er diese Ziele vereint.

*Integriert, ja. Es gibt verschiedene Theorien zum Gehirn. Hier sehen Sie ein solches Modell. Erläutern Sie mir doch mal die Prozesse, die hier ablaufen. (Zeigt eine unbeschriftete Version des Wolke-Tabelle-Datenbank-Bildes zum Information Processing Model.)*

Es gibt das Ultrakurzzeitgedächtnis, das Kurzzeitgedächtnis und das Langzeitgedächtnis... (so ging es weiter – Prof. Schroeder wollte die Prozesse wissen, die Wissen vom einen in den anderen Speicher bringen.)

*Und wie kommt es dazu, dass sich Wissen hier im Langzeitgedächtnis ablagert?*

Äh, vielleicht durch Wiederholung?

*Wozu soll die gut sein?*

Zum Festigen der kognitiven Strukturen.

*Na ja. Vielleicht. Geht das denn nur in eine Richtung, dieses Einlagern von Informationen?*

Nein, auch andersrum. Zum Beispiel vom Langzeit- in das Kurzzeitgedächtnis...

*... weil man ja immer wieder neue Eindrücke klassifizieren muss. Herr Paivio hat eine weitere Theorie entwickelt. Können Sie mir diese Abbildung erläutern? (Zeigt Paivios Dual Coding-Modell.)*

Paivios Ansatz ist, dass der Mensch über zwei kognitive Systeme verfügt: eins für nonverbale Informationen, z.B. für Bilder, und eins für verbale Informationen. Das Erinnern wird erleichtert, wenn beide Systeme verknüpft werden, z.B. durch die Begleitung eines Vortrags mit einer Präsentation.

*Zum Was ist da wichtig?*

Dass man die Mittel angemessen einsetzt.

*Das heisst, die eindeutige, deutliche Verknüpfung von Begriff und visuellem Eindruck. Sonst haben Sie das Problem der kognitiven Überladung. Ich denke, damit sind wir fertig – warten Sie bitte einen Moment lang draußen!*

**Fazit:**

Beide Prüfer sind sehr angenehm und fair und waren auch „im Team“ gut.

Wichtig ist nach übereinstimmender Mitteilung beider Prüfer in der Vorbesprechung bei einer praktischen Prüfung die Breite, weniger die Tiefe. Sich beim Lernen ständig die Fragen zu stellen „Warum wurde diese Methode/Technik eingeführt? Welchen Nutzen hat man davon? Was wird dadurch verbessert?“ hilft ungemein, auf die Prüfungssituation vorbereitet zu sein. Man muss in der Lage sein, Zusammenhänge darzustellen. Ergo: zeigen, dass man sich in dem Gebiet auskennt und den Überblick hat. Wenn dann mal eine Antwort fehlt, kann man ausweichen und über ein paar Fragen des Prüfers vielleicht doch noch die Antwort herleiten. Ich denke, dass diese Beobachtung in etwa für alle Prüfer und alle Prüfungen in Praktischer Informatik gilt.

Viel Erfolg euch allen!

P.S.: Ja, das Protokoll ist lang. Ich habe ausführliche Protokolle vermisst. Bei den stichwortartigen habe ich immer nur gedacht, dass der Prüfling offenbar alles gewusst hat und alles präzise formulieren konnte. Das zu können ist aber absolut nicht notwendig, wenn natürlich nicht von Nachteil.

## Literatur:

Die folgende Liste enthält alle Bücher, die ich zur Vorbereitung benutzt habe. Ich habe mir das Weglassen der bibliographischen Angaben bis auf Autor und Titel sowie eine Kommentierung erlaubt, die sich rein auf den Nutzen der Werke zu meiner persönlichen Prüfungsvorbereitung bezieht.

### Einführung in die Softwaretechnik

*Balzert, Helmut: Lehrbuch der Software-Technik. Band 1.* Ganz gut zum Nachlesen der strukturierten Analyse (SA). Sonst m.E. weniger gut, da die Themen der Vorlesung nur angeschnitten, aber nicht vertieft werden (zumindest in diesem ersten Band. Der zweite Band ist in dieser Hinsicht besser). Das ist aber eine sehr persönliche Einschätzung. Das Buch wird oft hoch gelobt und sehr empfohlen.

*Sommerville, Ian: Software Engineering.* Der Klassiker. Gut für die Prozessmodelle.

*Nagl, Manfred: Softwaretechnik: methodisches Programmieren im Großen.* Die ersten beiden Kapitel sind m.E. die ausformulierte Version der Vorlesungs-Einführung.

*Nagl, Manfred: building tightly integrated software development environments: the IPSEN approach.* Fand ich für die Vorbereitung völlig unbrauchbar.

### Software-Qualitätssicherung

*Myers, Glenford J.: methodisches Testen von Programmen.* Der Klassiker zum Black-Box-Test. Muss man lesen.

*Riedemann, Eike H.: Testmethoden für sequentielle und nebenläufige Software-Systeme.* Was der Mann alles weiß! Das Buch ist prima für Black- und White-Box-Tests. Wer lernen will, wie man Testfälle erstellt, lernt das hier. Verschiedene Betrachtungen zur Testwirksamkeit und zum Aufwand der Testfallerstellung sind ebenfalls enthalten.

*Wallmüller, Ernest: Software-Qualitätsmanagement in der Praxis.* Super. M.E. eine der Vorlagen für Prof. Lichters Vorlesung.

*Frühau, Karol et al.: Software-Prüfung: eine Fibel.* Umfangreiche Darstellung von Reviews. Das Management-Kapitel ist gut!

*Balzert, H.: Lehrbuch der Software-Technik. Band 2.* Erläutert ausführlich die Prozessmodelle, das CMM und sehr, sehr viel zum Projektmanagement. Für letzteres lohnt es sich, ein wenig Zeit zum Lesen mitzubringen.

*Bashir, Imran, Goel, Amrit L.: Testing Object-Oriented Software: Life-Cycle Solutions.* Kaum brauchbar zur Vorbereitung. Allerdings motiviert der vorgestellte Ansatz des Klassentests, mal eingehender über FREE nachzudenken.

### E-Learning Konzepte & Systeme

*Jank, Meyer: didaktische Modelle.* Die Einführung ist super und erklärt einiges zur Didaktik. Ist offenbar auch ein „Klassiker“.

*Edelmann, Walter: Lernpsychologie.* Sehr hilfreich, um sich einen Überblick zu verschaffen.

*Hops, Daniel: Motivation in E-Learning-Systemen. Diplomarbeit am LuFG Informatik IX.* Erläuterung des ARCS-Modells.

<http://tip.psychology.org/>. Erklärt alle Lerntheorien aus der Vorlesung auf jeweils einer bis zwei Seiten, und dazu noch ungefähr 40 weitere. Wunderbar als Referenz.

... bevor ich weiteres aufzähle: man sollte sich die Mühe machen, für E-Learning das Internet zu durchforsten. Den Hinweisen auf Webseiten aus der Vorlesung sollte man ebenfalls nachgehen. Im Internet findet man (irgendwie erwartungsgemäß) eine ganze Menge zu E-Learning-Konzepten und Lerntheorien.