

## Vorlesungen, Prüfungen und Seminare in Software Engineering an der Universität Stuttgart

*J. Ludewig*

*Institut für Informatik der Universität Stuttgart*

*Breitwiesenstr. 20-22, 7000 Stuttgart 80*

### Lehrangebot

Die folgende Tabelle zeigt das Lehrangebot unserer Abteilung. Dabei sind Veranstaltungen, die nur aushilfsweise übernommen sind, weggelassen, ebenso Kurse in Programmiersprachen (Ada, Smalltalk, Eiffel) und individuelle Aufgaben (SoPras, Studien- und Diplomarbeiten). Angegeben ist der aktuelle Stand (SS 91 bis 92). Rechts stehen der Typ (Vorlesung, Übung, Seminar, Hauptseminar, Praktikum und Kolloquium) und die Zahl der Semesterwochenstunden.

### Vorlesungen

Software Engineering	4V 1Ü
Konzeption und Aufbau objektorientierter Programme	2V 2Ü
Methoden und Notationen der frühen Phasen	2V
Realisierung spezieller Software-Systeme	2V
Software-Verwaltung und -Wartung	2V
Software-Metriken	2V 1Ü

### Seminare und Hauptseminare über 2S/2HS

Methoden der Software-Entwicklung; CASE; Metriken;  
Methoden und Werkzeuge für die frühen Phasen;  
Software-Wartung und Wiederverwendung; Prozeß-Modelle;  
Formale Methoden und Notationen für die Software-Entwicklung

Fachpraktikum Software-Projektdurchführung 4P

Kolloquium Durchführung wissenschaftlicher Arbeiten 2K

### Prüfungen

Um das schwierige Problem der Stoffabgrenzung in den Griff zu bekommen, habe ich seit einiger Zeit das Verhältnis zwischen Vorlesung und Prüfung umgekehrt und beginne nun jeweils mit der Bekanntgabe der Prüfungsfragen; das ist derzeit ein Katalog von 26 Fragen mit bis zu sieben Unterpunkten, insgesamt knapp hundert Fragen. Während der Vorlesung ist jederzeit klar, um welche Frage es gerade geht (siehe Beispiel unten). So verringert das Korsett des Fragenkatalogs auch die Gefahr, allzu weitläufig zu berichten, statt präzise Aussagen zu machen.

### 18. CASE

18.1 Was ist CASE? Welche Anforderungen gibt es an CASE-Systeme? Wie sind CASE-Systeme zu klassifizieren?

18.2 Wie sind typische Spezifikationssysteme heute aufgebaut?

18.3 Worin besteht das Speicherungsproblem?

18.4 Worin besteht das Tracing-Problem?

18.5 Was leisten heutige Spezifikationssysteme, und was nicht?

Was ist bei Auswahl und Einführung von CASE zu beachten?

Durch Prüfungen können wir Wissen, Fähigkeiten und Verständnis der Zusammenhänge feststellen. Wissen läßt sich sehr gut schriftlich prüfen, notfalls sogar im Multiple Choice-Verfahren. Fähigkeiten können ebenfalls in einer Klausur demonstriert werden, wenn es gelingt, entsprechend kleine Aufgaben zu stellen, und wenn eine objektiv richtige Lösung bekannt ist. Das Verständnis der Zusammenhänge läßt sich m.E. nur im Gespräch feststellen; die Alternative wären hier Aufsätze, deren Durchsicht und Bewertung noch schwieriger wäre als die Bewertung mündlicher Prüfungen. Mündliche Prüfungen haben den weiteren Vorteil, daß sie zeigen, wieweit die Kandidaten in der Lage sind, sich mit Einwänden auseinanderzusetzen und ihre Position zu vertreten - im Software Engineering sicher keine praxisferne Aufgabe. Aus diesen Gründen wird Software Engineering hier mündlich geprüft, und ich bin nicht daran interessiert, es zum Kernfach zu machen, denn dadurch wäre der Prüfungsmodus vorgegeben.

### Vorlesungen

Alle Vorlesungen - auch die Grundvorlesung Software Engineering mit etwas mehr als einhundert Teilnehmern - sind seminarartig, d.h. es sind jederzeit Einwände und Diskussionen möglich, und ich rege sie auch durch Fragen an. Bei diesem Ansatz ist zwar das Pensum reduziert, aber die Aufmerksamkeit und das Interesse der meisten Teilnehmer deutlich erhöht.

### Seminare

Informatik-Studenten der Universität Stuttgart benötigen für das Hauptdiplom jeweils ein Seminar und ein (benotetes) Hauptseminar. Wir verfolgen dabei zwei spezielle Ziele: Einerseits bemühen wir uns intensiv (und mit einigem Erfolg), die Vortragstechnik der Teilnehmer zu verbessern. Dazu trägt auch eine Lehrveranstaltung bei, in der neben den Prinzipien des wissenschaftlichen Arbeitens auch der Aufbau von Berichten und Vorträgen behandelt wird (Deiningert et al., 1992).

Zweitens gestalten wir die Aufgaben so, daß die Beteiligung an den Vorträgen der übrigen Teilnehmer im Interesse aller Beteiligten liegt. Dazu erhält jeder Teilnehmer neben seinem Vortragsthema auch ein "Querschnittsthema", das im günstigsten Fall orthogonal zu den Vortragsthemen liegt. Wird beispielsweise in jedem Vortrag eine bestimmte Methode vorgestellt, so wäre ein Querschnittsthema die zu begründende Auswahl einer Methode unter bestimmten Rahmenbedingungen.

### Offene Fragen

Vorlesungen sind keineswegs die ideale Form der Lehre (Goldschmid, 1986). Das gilt verstärkt im Software Engineering, wo es bis heute nur wenig kodifiziertes, allgemein anerkanntes Wissen gibt. Hier wäre die gemeinsame Arbeit als Unterrichtsform angemessen ("learning by doing").

Dieser Ansatz scheitert aber schon am Zahlenverhältnis zwischen Professoren und Assistenten auf der einen, Studenten auf der anderen Seite: in Stuttgart ca. (1+4) : 100. Wissenschaftliche Hilfskräfte, sonst oft eine akzeptable Notlösung, helfen hier kaum, weil sie auf einem so schwankenden Gebiet allzu große Mühe haben und auch von ihren Kommilitonen nicht als Lehrer anerkannt werden.

Das "Wang-Institute of Graduate Studies" in Tyngsboro, Massachusetts, hat uns vor knapp zehn Jahren demonstriert, wie ideale Bedingungen für den Unterricht in Software Engineering aussehen: Mehrjährige Praxiserfahrung als Voraussetzung der Einschreibung, eine Relation von etwa 6:1 zwischen Lernenden und Lehrenden, unter diesen bekannte Namen wie Susan Gerhard und Richard Fairley, und eine perfekte räumliche und technische Ausstattung. Es wäre unfair, in der allgemeinen Mangelsituation solche Bedingungen zu fordern. Wie aber können wir trotz des Mangels dem Ideal möglichst nahekommen?

### Quellen, Literatur

Deininger, M., H. Lichter, J. Ludewig, K. Schneider (1992): Studien-Arbeiten - Anleitung zur Vorbereitung, Durchführung und Betreuung von Studien-, Diplom- und Doktorarbeiten am Beispiel Informatik.  
vdf, Zürich, und Teubner, Stuttgart, 1992.

Goldschmid, M. (1986): Unterlagen zu einem hochschuldidaktischen Seminar an der ETH Zürich, mit einer Liste von ca. 50 Unterrichtsformen; unveröffentlicht.