

# Blended Learning: Informatik-Lehre an der Hochschule Offenburg

**Prof. Dr. Claudia Schmidt**  
Fakultät Medien und  
Informationswesen (M+I)

Badstraße 24  
77652 Offenburg  
Tel.: 0781 205-133  
c.schmidt@fh-offenburg.de

**1963:** Geboren in Westerlohe  
**Bis 1992:** Informatikstudium an der Universität Karlsruhe  
**1992–1998:** Wissenschaftliche Mitarbeiterin und Promotion  
am Institut für Telematik der Universität Karlsruhe  
**Seit 1998:** Professorin für Telekommunikation  
an der Hochschule Offenburg



**Arbeitsgebiete:** Computernetze, Neue Entwicklungen im Internet,  
Kommunikationsunterstützung für multimediale Anwendungen,  
Management von Kommunikationsdiensten

## 4.3 Blended Learning: Informatik-Lehre an der Hochschule Offenburg

Prof. Dr.-Ing. Claudia Schmidt  
Prof. Dr. rer. pol. Volker Sängler

### Abstract

*Additionally to the courses Software Engineering, Computer Networks and Databases we provide web-based e-learning applications so that our students are able to repeat the contents of the courses independantly of time and place.*

*Those E-Learning applications – we call them M+I-Learning – have the same structure as the corresponding courses. Inside, the facts are textually and graphically presented, sometimes with animations. Especially quizzes and exercises in which the learner works actively are a helpful supplement to the learning by listening during the lecture.*

*To motivate the learners we offer furthermore an online learning game “Software Engineering in the Future” in which the student Ben travels through space and has to solve various problems. By solving the problems the player learns to understand and to handle Software Engineering models and concepts.*

*The applications are available under <http://mi-learning.mi.fh-offenburg.de>.*

### 1. Einleitung

Informatik-Veranstaltungen in der Fakultät Medien und Informationswesen vermitteln meist komplexe Inhalte, die anschließend in begleitenden Laborveranstaltungen praktisch und an konkreten

Beispielen vertieft werden. Allerdings benötigen die Studierenden für ein reiches Labor und die selbstständige Erarbeitung korrekter Lösungen einige Grundkenntnisse, die aus der jeweiligen Theorieveranstaltung mitgebracht werden müssen.

Um den Studierenden weiterhin die Möglichkeiten zu geben, den Stoff der Lehrveranstaltungen raum- und zeitunabhängig nachzuarbeiten und auch didaktisch aufbereitete Übungen virtuell durchzuführen, haben wir zu den Veranstaltungen Software Engineering, Computernetze und Datenbanken webbasierte E-Learning-Materialien konzipiert und erstellt. Diese Lernarrangements bieten den Lernenden die Möglichkeit, selbstbestimmt im eigenen Lernrhythmus und über unterschiedliche Medien einen Zugang zu der Thematik zu finden. Hybride Lernarrangements (Blended Learning) versuchen hier die Vorteile unterschiedlicher didaktischer Methoden und Medien miteinander zu kombinieren [1].

## 2. Die E-Learning-Applikationen

### 2.1 M+I-Learning-Lektionen

Zu den Vorlesungen Software Engineering, Computernetze und Datenbanken existieren E-Learning-Lektionen, die genau wie die zugehörigen Vorlesungen strukturiert sind – jedem Kapitel der Vorlesung entspricht ein Kapitel der zugehörigen E-Learning-Lektion. Die Lektionen basieren auf einem flexiblen Rahmenwerk, das realisiert wurde, damit die eigentlichen Inhalte der Lektionen von unterschiedlichen Personen erstellt werden können.

Der Rahmen der gesamten Lektion ist für alle Inhalte unveränderlich und damit für den Lernenden konsistent [2]. In Abbildung 4.3-1 sind die Kapitel der Lektion Software Engineering am oberen Bildrand erkennbar, die Unterkapitel erscheinen am linken Bildrand.

Neben den Fakten, die das Wissen aus der Vorlesung textuell, grafisch und zum Teil animiert darstellen, sind Übungen und Quizzes die zentralen Elemente der Lektionen, die in jedem Kapitel auftauchen. Sie sind visuell ansprechend realisiert, sodass sie die Lernenden zum Üben motivieren. Das Lernen erfolgt damit nicht wie in der Vorlesung durch Hören und Sehen, sondern durch selbst erfahren und selbst tun und stellt somit eine wirkungsvolle Ergänzung dar. Die Übungen zeichnen sich durch einen unterschiedlichen Grad der Interaktivität aus. Mit einfachen Übungen werden Algorithmen und Methoden visualisiert und können somit besser verstanden werden. Durch komplexe Übungen werden explorative Lernaktivitäten unterstützt. Die Lernenden können eigene Hypothesen aufstellen und diese anschließend durch eine gezielte Interaktion mit den Medien überprüfen [3].

Ergänzend und vertiefend absolvieren die Studierenden danach noch ein Praktikum (Software-Engineering) und Labore (Computernetze und Datenbanken), in denen etwas größere Aufgabenstellungen mit praxisnahen Werkzeugen gelöst werden. Im Falle der Datenbanken etwa wird eine Datenbank entsprechend einer Anforderungsanalyse in Oracle und deren Management-Tool SQLDeveloper [4] konzipiert, implementiert, mit Daten gefüllt und mit einem Web-Interface versehen.

Software Engineering Hochschule Offenburg  
University of Applied Sciences

Software **UML** Qualitätsmanagement ER-Modell Benutzerschnittstelle Vorgehensweisen

### Klassendiagramm Fakten

**Assoziationen**  
Beziehungen zwischen Klassen werden durch Assoziationen dargestellt. Über eine Assoziation wird die Kommunikation zwischen den Objekten beschrieben und damit ein Zugriff auf die Element (Attribute und Operationen) einer andere Klasse definiert.

Wir betrachten hier nur **binäre Assoziationen**, d.h. Beziehungen zwischen zwei Klassen. Diese werden in den Klassendiagrammen als durchgezogene Linie zwischen den Klassen dargestellt.

Über die folgenden optionalen Angaben können Assoziationen detaillierter spezifiziert werden:

- » **Assoziationsname:** Der Name einer Assoziation ist optional und wird an der Assoziationslinie – nicht zu nahe an den Enden, da es sonst zu einer Verwechslung mit Rollen kommen kann – angegeben. Da der Name oft nicht für beide Assoziationsrichtungen zutrifft, kann zusätzlich eine bevorzugte Leserichtung über ein ausgefülltes Dreieck hinzugefügt werden.
- » **Rollen:** an den Enden einer Assoziation kann über die Angabe von Rollen die spezifische Verwendung der Klassen in der Assoziation näher beschrieben werden. Besondere Bedeutung kommt den Rollen zu, wenn Klassen über mehrere unterschiedliche Assoziationen verbunden sind und somit nur über die Rolle eine Unterscheidung getroffen werden kann.

Assoziationen in UML

- » **Kardinalitäten:** Neben den Rollen können an den Assoziationsenden Kardinalitäten angegeben werden, die Mengenverhältnisse zur Laufzeit (wie viele Beziehungen darf ein Objekt mit wem besitzen?) beschreiben. Dabei wird die zu einer Rolle gehörende Kardinalität am gegenüberliegenden Assoziationsende angegeben.

Anzeigen: Alles

Prof. Dr. Claudia Schmid, c.schmid@h-offenburg.de Prof. Dr. Volker Säenger, volker.saenger@h-offenburg.de

Abb. 4.3-1: Kapitel der Lektion Software Engineering sind am oberen Bildrand erkennbar, die Untertitel erscheinen am linken Bildrand

Die Erfahrungen der vergangenen zwei Semester zeigten, dass zur Klausurvorbereitung erneut auf die E-Learning Lektionen zugegriffen wird, um den Stoff zu wiederholen, Details herauszuarbeiten und das Wissen zu festigen. Standardisierte Berichte über die Zugriffe, die in unserer Lernplattform Moodle zur Verfügung stehen, bestätigen dies.

## 2.2 Online-Lernspiel

Ein spezielles Problem der Veranstaltung Software-Engineering ist die Tatsache, dass die Studierenden für die teilweise abstrakten Themen dieser Disziplin wenig Interesse aufbringen und nicht zum Lernen motiviert sind. Wozu braucht man diese vielen Methoden für die relativ kurzen Programme, die am Anfang des Studiums realisiert werden? Die zu erstellenden Diagramme liefern keine greifbaren, sondern eher theoretische Zwischenergebnisse.

Um die genannten Probleme zu beheben, wurde ein Online-Lernspiel für das Software-Engineering konzipiert und implementiert. Das primäre Ziel des Spiels ist

es, die Studierenden zur Arbeit mit den wichtigen Methoden und Vorgehensweisen anzuregen. Zunächst soll durch das Angebot eines Lernspiels die Hemmschwelle gesenkt werden, sich überhaupt mit Software Engineering zu beschäftigen. Der Spaß, die Konzentration und die Selbstbestimmtheit beim Spielen sollen die Motivation verbessern und daraus soll ein besseres und vor allem lang andauerndes Verständnis resultieren.

Für die Lernenden sind klar definierte und gut erreichbare Ziele wichtig. Deshalb ist das Spiel in kleine Einheiten unterteilt, sogenannte Minispiele, in denen die Konzepte und Methoden des Software Engineering trainiert werden. Weil im Spiel alle Themen der gesamten Vorlesung vorkommen, soll es semesterbegleitend gespielt werden. Das sture Lernen auf die Klausur und das anschließende rasche Vergessen soll vermieden und stattdessen nachhaltiges Wissen vermittelt werden.

Damit die Studierenden sich mit der Spielwelt identifizieren können, wurde als Hauptperson der Student Ben ausge-

wählt, der am Semesterende noch mit der Prüfung in seinem Studienfach Software Engineering zu tun hat. Um gleichzeitig aber eine alternative Lernsituation zu erhalten, spielt die Geschichte in der Zukunft. Ben möchte nach Abschluss seiner Prüfung schnellstmöglich zusammen mit seiner Freundin Jacqueline zu einem Festival in einem anderen Teil des Weltalls reisen. Auf seiner Reise wird er mit unterschiedlichen Problemen konfrontiert, deren Lösungen Kenntnisse und Fertigkeiten des Software Engineering erfordern.

Die Abbildungen 4.3-2a bis 4.3-2c zeigen exemplarisch die Umsetzung der beschriebenen Überlegungen. In Abbildung 4.3-2a) ist Ben in dem pink gefärbten Helm dargestellt. Abbildung 4.3-2b) zeigt ein Detail seines Fahrzeugs, ein Stück der Armaturen des sogenannten Spacebikes. Weil Ben Student ist, kann er sich kein wirklich gutes Fahrzeug leisten und repariert auftretende Schäden notdürftig und „zukunftsgerichtet“ – mit Klebeband. Abbildung 4.3-2c) zeigt eine futuristische Szene, in

deren Vordergrund der zentrale Platz der Hochschule Offenburg mit einem tatsächlich vorhandenen Kunstwerk stilisiert ist. Jeder Studierende der Hochschule wird das Szenario wiedererkennen.

Obwohl die Story einige Kontraste zur studentischen Realität aufweist, ist es für das Lernen unabdingbar, dass die Lerninhalte und somit die einzelnen Aufgaben sehr eng an den Inhalten der Vorlesung Software Engineering orientiert sind. Die verwendeten Begriffe, Notationen, Diagramme und Konzepte zeigen große Pa-



Abb. 4.3-2a): Ben in dem pink gefärbten Helm



Abb. 4.3-2b): Fahrzeugteil von Ben



Abb. 4.3-3:  
Ausschnitt eines  
ER-Diagramms



rallelen zu den Unterlagen der Vorlesung. Ein Beispiel für den Ausschnitt eines ER-Diagramms zeigt Abbildung 4.3-3.

Laut [5] fördern digitale Spiele die Motivation der Lernenden, indem sie

- die Fantasie anregen,
- neugierig machen,
- Herausforderungen bieten,
- audio-visuell gestaltet sind und
- den Lernenden über definierte Regeln die Kontrolle überlassen.

Alle diese Aspekte wurden in „Software Engineering in the Future“ bewusst integriert, um die Motivation zum Spielen hochzuhalten.

Die bereits vorn erwähnten Moodle-Statistiken der vergangenen zwei Semester zeigen Erfolge dieser Strategie. Die Studierenden spielen das Spiel oft, gehen somit automatisch mehr mit den Begriffen und Methoden des Software Engineerings um und erwerben auf diese Weise Wissen und Können. Eine Umfrage ergab sehr positive Bewertungen des Spiels.

### 3. Bewertung

Wegen der bereitgestellten E-Learning Applikationen arbeiten die Studierenden ergänzend zum Frontalunterricht in der

Vorlesung selbstständig mit dem Stoff der Lehrveranstaltung. Laut eigenen Aussagen macht diese selbstbestimmte Arbeit Spaß, und dies erhöht die Erfolgchancen. Wir werden verfolgen, ob die Klausurergebnisse sich in den nächsten Jahren signifikant verbessern. Ebenso werden wir untersuchen, ob Zusammenhänge zwischen der Intensität der Arbeit mit den E-Learning-Modulen und Klausurergebnissen erkennbar sind. Schon heute stellen wir fest, dass der Spaß, die Mitarbeit und die Begeisterung für die Informatik-Veranstaltungen aufgrund der E-Learning-Angebote gestiegen sind.

### Referenzen

- [1] Kerres, M. Multimediale und telemediale Lernumgebungen. Konzeption und Entwicklung (2. Aufl.), München, R. Oldenbourg, 2001.
- [2] Sänger, V; Schmidt, C. MI-Learning: ein Rahmenwerk für webbasiertes E-Learning. In „Die Energie der Didaktik – Beiträge zum 7. Tag der Lehre“, pp. 64-67, Biberach, 2007
- [3] Schulmeister, R., Interaktivität in Multimedia-Anwendungen. e-teaching.org,2005.<http://www.e-teaching.org/didaktik/gestaltung/interaktiv/InteraktivitaetSchulmeister.pdf>
- [4] Oracle. SQLDeveloper. In Oracle Technology Network, [http://www.oracle.com/technology/products/database/sql\\_developer/index.html](http://www.oracle.com/technology/products/database/sql_developer/index.html), 13.11.08
- [5] Garris, R; Ahlers, R; Driskell, J.E. Games – Motivation and Learning: Research and Practice Model. In “Simulation & Gaming”. Newbury Park, Sage Publ., 2002

Abb. 4.3-2c): Futuristische Szene, in deren Vordergrund der zentrale Platz der Hochschule Offenburg mit einem tatsächlich vorhandenen Kunstwerk stilisiert ist