

Software-Engineering-Unterricht online

Ilse Schmiedecke, Debora Weber-Wulff

TFH Berlin, Labor Online-Learning, Luxemburger Str. 10, 13353 Berlin, schmiedecke@tfh-berlin.de
FHTW Berlin, Internationale Medieninformatik, Treskowallee 8, 10318 Berlin weberwu@fhtw-berlin.de

1 Ist Online-Unterricht für Software Engineering geeignet?

Der Bedarf an E-Learning-Modulen wächst ständig. So wurde auch im Rahmen des Bachelor-Studienganges Medieninformatik an der Virtuellen Fachhochschule (VFH)¹ ein Online-Modul Software Engineering benötigt. Für das Online-Lehrmaterial galten die bekannten Anforderungen: an Semesterwochen orientierte Lerneinheiten, Tests zur Selbstkontrolle und vor allem die Nutzung der Hyper- und Multimediamöglichkeiten und des Internets zur didaktisch optimalen Präsentation des Lernstoffs [GöM 02].

Die größere Herausforderung war die angemessene Gestaltung des Online-Unterrichts. Denn Software-Engineering-Unterricht muss über reines Wissen hinaus auch fachspezifische „Soft Skills“ trainieren: Fachkommunikation, Team-Arbeit und -Organisation und gemeinschaftliche Problemlösung. Das Studium an der VFH unterscheidet sich vom klassischen Fernstudium vor allem dadurch, dass die Studierenden ihren Mentor und die Lerngruppe kennen und online miteinander kommunizieren, etwa im Chat, in dem der Mentor oder die Mentorin mit der gesamten Lerngruppe bestimmte Themen aufgreift und vertieft. Die Frage ist, ob die eingeschränkten Kommunikationsmöglichkeiten in einem virtuellen Lernraum ausreichen, um die erforderlichen Soft Skills zu trainieren. Da wir hier selbst skeptisch waren, möchten wir über unsere Erfahrungen mit dem Modul berichten.

2 Asynchrone Kommunikation

Voraussetzung für echte Teamarbeit ist ein synchronisierter Lernfortschritt, was gut durch bewertete terminierte Einsendeaufgaben zu erreichen ist. Wir sind an dieser Stelle noch eine Schritt weiter gegangen, indem wir anstelle der Einsendung die Veröffentlichung im Forum des virtuellen Lernraums gefordert haben. Bewertet wurde nur die fristgerechte Veröffentlichung – die Inhalte wurden zur Diskussion gestellt, im Forum und im wöchentlichen Chat. Dabei lernten die Studierenden einerseits, ihre guten Ideen

¹ BMBF Leitprojekt Virtuelle Fachhochschule: <http://www.oncampus.de>

preiszugeben und von denen anderer zu profitieren, und andererseits entwickelte sich teilweise im Forum eine sehr fruchtbare fachliche Diskussion, auf die der Mentor oder die Mentorin dann im Chat zurückgreifen konnte.

3 Fortgeschrittene synchrone Kommunikation

Was die synchrone Kommunikation betrifft, mussten wir feststellen, dass ein reiner Text-Chat für unsere Zwecke sehr unbefriedigend war. Die meisten Diskussionen im Rahmen des Software-Engineering-Unterrichts drehen sich um Fragen der Modellierung. Die Studierenden lernen schnell, dass es das „richtige“ Modell für ein Problem nicht gibt, sondern nur je nach Intention mehr oder weniger gute Modelle. Verschiedene Alternativen asynchron zu diskutieren, ist ermüdend und schließt selten die ganze Lerngruppe ein.

Daher wird ein synchrones Kommunikationswerkzeug benötigt, das Gruppendiskussionen am Flipchart simuliert. Ein Whiteboard parallel zum Text-Chat, wie es die meisten Lernumgebungen anbieten, ist eine Hilfe, aber zumeist demotivierend schwerfällig und unflexibel. Die Autorinnen haben deshalb oft Zusatzmaterial auf ihre Homepage gestellt – teils auch während des Chats –, auf das sich der Chat beziehen konnte. Aber allein die textuelle Bezugnahme ist unangemessen mühsam.

Im Netucate-Lernraum² haben wir eine empfehlenswerte Alternative gefunden: Netucate bietet einen Audio-Chat (Video ist zuschaltbar) mit einem zusätzlichen Text-Chat für Einwürfe und Zwischenfragen. Zusätzlich zum Whiteboard gibt es einen „Shared Pointer“, mit dem die Teilnehmer zeigen können, worauf sie sich beziehen, und vor allem das sog. „Application Sharing“: Jeder Teilnehmer kann der Gruppe eine beliebige Applikation online zur Verfügung stellen. Modelldiskussionen verlaufen dann fast wie am Flipchart: Jemand präsentiert einen Modellvorschlag, der gemeinsam modifiziert wird. Darüber hinaus lassen sich Sitzungen aufzeichnen.

4 Persönliche Begegnungen

Technisch und didaktisch lässt sich also auch der „Soft Skill“-Aspekt des Software-Engineering-Unterrichts online mindestens so gut abdecken wie im Präsenzstudium. Dennoch empfehlen wir, wenigstens zwei Präsenztermine vorzusehen, da die persönliche Begegnung die Online-Kommunikation nachhaltig verbessert³.

Literatur

[GöM 02] G. Görlitz und S. Müller: Didaktisches Design für eine Online-Programmierausbildung. GI Jahrestagung, Dortmund. 2002

² <http://www.netucate.com>

³ Siehe auch <http://www.f4.fhtw-berlin.de/~weberwu/papers/SoftEngUnterrOnline.doc>.