

Kompetenzförderung im Software Engineering durch ein mehrstufiges Lehrkonzept im Studiengang Mechatronik

Jörg Abke, Vincent Schwirtlich

Yvonne Sedelmaier

Fakultät Ingenieurwissenschaften
Hochschule Aschaffenburg
Würzburger Strasse 45
63743 Aschaffenburg
joerg.abke@h-ab.de
vincent.schwirtlich@h-ab.de

Fakultät Elektrotechnik und Informatik
Hochschule Coburg
Friedrich-Streib-Str. 2
96450 Coburg
sedelmaier@hs-coburg.de

Abstract: Dieser Beitrag stellt das Lehr-Lern-Konzept zur Kompetenzförderung im Software Engineering im Studiengang Mechatronik der Hochschule Aschaffenburg dar. Dieses Konzept ist mehrstufig mit Vorlesungs-, Seminar- und Projektsequenzen. Dabei werden Herausforderungen und Verbesserungspotentiale identifiziert und dargestellt. Abschließend wird ein Überblick gegeben, wie im Rahmen eines gerade gestarteten Forschungsprojektes Lehr-Lernkonzepte weiterentwickelt werden können.

1 Einleitung

Software ist heute verantwortlich für die Funktion nahezu jedes technischen Geräts. Entsprechend ist Kompetenzförderung zur Entwicklung von Software, dem Software Engineering, in Informatikstudiengängen wie in ingenieurwissenschaftlichen Studiengängen notwendig. Speziell im Software Engineering stehen Hochschullehrer hierbei vor einer Vielzahl von Herausforderungen. Diese reichen von heterogenen Vorkenntnissen und Vorwissen bei den Studierenden z. B. im Bereich der Programmierung bis hin zu unterschiedlichen Erfahrungen im Software Engineering. Gleichzeitig ist die Affinität der Studierenden von Querschnittsstudiengängen wie Mechatronik zu Informatikthemen eher gering.

Dieser Beitrag stellt die angestrebten Kompetenzen und das Lehr-Lern-Konzept zur Kompetenzförderung im Software Engineering im Studiengang Mechatronik der Hochschule Aschaffenburg vor und diskutiert Erfahrungen mit diesem Lehr-Lern-Konzept aus inzwischen zwei Durchläufen.

2 Lehr-Lern-Konzept

2.1 Ziele und angestrebte Kompetenzen des Lehr-Lern-Konzepts

Die Veranstaltung Software Engineering (SWE) findet im Studiengang Mechatronik im vierten Semester mit 4 Semesterwochenstunden (SWS) statt und besteht aus 2 SWS Vorlesung und 2 SWS Gruppenübungen. Vorherige Semester vermitteln Grundlagen der strukturierten Programmierung mit der Sprache C, Grundlagen der Informatik sowie Grundlagen der Mikrocontroller und deren Programmierung. Die Lehrveranstaltung SWE will den Studierenden einen Überblick über Software-Entwicklungsprozessmodelle und -prozesse geben und in Anforderungsanalyse, Spezifikationen und das Testen von Software einführen. Die Studierenden sollen erste praktische Erfahrungen im Projekt- und Konfigurationsmanagement sammeln und gleichzeitig Soft Skills wie Abstraktions- und Problemlösungsvermögen oder Teamkompetenzen und Kommunikationsvermögen trainieren. Schlüsselqualifikationen erfahren besonders seit [Wi91] gerade auch im Bereich des Software Engineering zunehmende Bedeutung [Vi11], um später im sich immer schneller wandelnden Berufsleben bestehen zu können.

2.2 Erste Umsetzungsphase des Lehr-Lern-Konzepts im Sommersemester 2010

Im Vorlesungsteil der Veranstaltung vermittelte der Dozent theoretisches Grundlagenwissen zur Anwendung des Software Engineerings im Allgemeinen.

Die Gruppenübung erfolgte in zwei Phasen. Im ersten Drittel des Semesters wurde die Veranstaltung seminaristisch aufgebaut und im Rahmen der Handlungsorientierung zunächst die Methode Lernen-durch-Lehren angewandt [Ma02]. Hier müssen die Studierenden sich selbst aktiv Lerninhalte aneignen. Der Dozent vergibt Lernthemen wie etwa Projektmanagement mit MS-Project, Einführung in UML oder Software-Testtechnologien an Kleingruppen von ca. sieben bis acht Studierenden. Gleichzeitig fand eine Lehrsequenz zu Recherchetechniken in der Vorlesung statt. Die Kleingruppen werden vom Dozenten eingeteilt und müssen nicht nur die vorgegebenen Lerninhalte selbstorganisiert erarbeiten, sondern die Ergebnisse wiederum strukturieren und zusammenfassen, um sie dann den Mitstudierenden zu präsentieren.

In der zweiten Phase der Übung bearbeiten die gleichen Kleingruppen ein Software-Projekt. Die Aufgaben waren beispielsweise die Programmierung eines Morse-Trainers, eines Reaktionstesters oder eine verteilte Ampelsteuerung mit Lichtzeichendiagnose. Der zu verwendende Entwicklungsprozess, die Zielumgebung sowie ein klar umrissener Zeitrahmen mit Zielterminen zur Übergabe von Dokumenten wie Anforderungsliste, Systemspezifikation, Testfälle, und dem Finalprodukt wurde vorgegeben. Auf Basis dieser Vorgaben hatten die Gruppen ihre Zeit und Ressourcen selbstverantwortlich zu organisieren. Dazu waren sowohl das theoretische Überblickwissen der Vorlesung als auch die durch die handlungsorientierte Methode erworbenen Kenntnisse nötig.

Die Studierenden konnten so im Software Engineering selbst aktiv werden und Kenntnisse durch praktische Anwendung weiter vertiefen. Ferner konnten sie Team- und Kommunikationskompetenzen stärken [We08]. Problem-basiertes Lernen adressiert zudem Abstraktions- und Problemlösungskompetenzen [Zu06].

2.3 Zweite Umsetzungsphase des Lehr-Lern-Konzepts im Sommersemester 2012

Aufgrund der Evaluationen der Lehrveranstaltung aus dem Sommersemester 2010 wurde das Lehr-Lern-Konzept im Sommersemester 2012 leicht modifiziert: Die Zweiteilung der Veranstaltung in Vorlesung und Praktikum/Übung wie auch die Teilung des praktischen Teils in Seminar und Projektarbeit blieb bestehen. Allerdings werden jetzt nach dem seminaristischen Übungsteil die Gruppen durch den Dozenten neu zusammengesetzt, so dass die neuen Gruppen jeweils ein Mitglied der „alten“ fachdisziplinarisch gebildeten Kleingruppen enthalten. Bei Beibehaltung der alten Gruppeneinteilung wären pro Projektgruppe lediglich Kompetenzen für ein und dieselbe Thematik vertreten, mit entsprechender Ratlosigkeit während der Projektarbeit. Nicht ausgewogene Verteilung von Kompetenzen in den Teams minderte das Verständnis des Gesamtzusammenhanges und ließ das im Vorfeld vermittelte Fachwissen als nicht ausreichend erscheinen. Die neue Gruppeneinteilung stellt sicher, dass jede Projektgruppe einen „Fachmann“ aus jedem Teilgebiet besitzt. Auch für die gezielte Förderung von Soft Skills bietet diese Neuordnung Vorteile, da der „Fachmann“ in der neuen Kleingruppe gewissermaßen gezwungen ist, sein Fachwissen in die Gruppe einzubringen. So leistet dieses Lehr-Lern-Arrangement einen aktiven Beitrag zur Entwicklung des Selbstbewusstseins der Studierenden und bietet zahlreiche Anknüpfungspunkte für jeden einzelnen Studierenden, sich individuell fachlich wie auch persönlich weiterzuentwickeln.

Das beschriebene mehrstufige Lehrkonzept besteht also aus drei zentralen Teilen: Die Vorlesung zu Beginn versucht heterogenes Vorwissen und sehr unterschiedliche Kenntnisse auf ein einheitliches Wissensniveau zu bringen. Dieses Grundwissen stellt erste inhaltliche Zusammenhänge her, strukturiert den Rahmen für den weiteren Wissenszuwachs und für die selbstgesteuerte Arbeit in Kleingruppen. In der zweiten Phase werden die Studierenden über die Beschäftigung mit rein fachlichen Themen langsam an selbständiges und eigenverantwortliches Arbeiten herangeführt. Den Studierenden wächst so mehr Verantwortung zu, ohne sie bereits mit der Bearbeitung eines komplexen Projektes zu überfordern. Diese Herausforderung folgt erst in der dritten Phase: Studierende müssen dann selbständig im Team Lösungen finden, bewerten und gemeinsam getroffene Entscheidungen umsetzen. Hier steht projektorientiertes Lehren und Lernen im Vordergrund, auf das die Merkmale nach [Ju10] zutreffen. Der Fokus wechselt also im Lauf des Semesters von eher passiver Aufnahme fachspezifischer Kenntnisse im Rahmen der Vorlesung hin zu aktiver Anwendung vieler fachlicher und über-fachlicher Kompetenzen auf komplexe Problemstellungen in der Projektarbeit. Darstellendes Lehren tritt zugunsten entdeckenden Lernens zurück. Ist anfänglich noch der Erwerb neuen Wissens zentrales Ziel, so gewinnen zum Ende des Semesters Strukturierung, Anwendung und Transfer von Wissen an Bedeutung.

3 Fazit und Evaluation

Die Modifikation des Lehr-Lern-Konzepts erzeugt mit der Neueinteilung der Teams eine Situation, die dem tatsächlichen Arbeitsalltag sehr nahe kommt: Die Fachkompetenzen beschränken sich meist auf ein Gruppenmitglied, so dass das Team gewissermaßen zur Zusammenarbeit und gegenseitigen Kommunikation gezwungen wird [Fi08].

Mittels Lernen-durch-Lehren trainieren die Studierenden Kommunikationskompetenz und Abstraktionsvermögen durch die Notwendigkeit, Inhalte zu verstehen, zu strukturieren und für die Präsentation aufzubereiten [Gr08]. [We92] nennt diese Fähigkeiten didaktische Reduktion und Rekonstruktion.

[Ju10] sieht auch die Projektmethode in der letzten Phase als besonders geeignet zur Kompetenzförderung an. So nennt er als Vorteile neben der Prozess- und Problemorientierung auch die Förderung personaler und sozialer Kompetenzen wie z. B. die Befähigung zur Selbständigkeit, Selbststeuerung und -beurteilung.

Die größten Herausforderungen des aktuellen Konzepts sind die heterogenen Eingangskompetenzen und die geringe Affinität zur Software-Entwicklung. Die Auswirkung von Eingangskompetenzen auf den Lernerfolg wurde bereits in [We08] thematisiert. Im ersten Teil der Veranstaltung ist der Lehrende besonders gefordert, da er die heterogene Studierendengruppe auf ein gemeinsames Wissensniveau bringen muss. Teilnehmerorientierung [Si97] ermöglicht den Studierenden, neues Wissen mit vorhandenen Kenntnissen und Erfahrungen in Verbindung zu bringen.

Auch die Notengebung ist eine Herausforderung. Aufgrund der Vorgabe, Leistungen einzelner Studierender bewerten zu müssen, wird ein Portfoliokonzept [Pr10] mit einer mündlichen Prüfung kombiniert. Das Portfolio besteht aus Artefakten aus Seminar- und Softwareprojektphase, die jedem Studierenden zuzuordnen sind. Das Einzel-Kolloquium ermöglicht die Bewertung individueller Kenntnisse und wird zudem als Reflexionsmöglichkeit der Softwaregruppen genutzt. Eine Bewertung sozialer und personaler Kompetenzen scheint in diesem Kontext nur schwer möglich und auch nicht sinnvoll zu sein. Dieses Problem sieht auch [Ju10] und diskutiert Möglichkeiten zur Überprüfung des Kompetenzerwerbs ausführlich.

Eine systematische Evaluation dieser Lehrveranstaltungen erfolgte in beiden Durchgängen, wobei zuletzt ein überarbeiteter Evaluationsbogen verwendet wurde, der die Ergebnisse im Forschungsprojekt EVELIN (Experimentelle Verbesserung des Lernens von Software Engineering) sechs bayerischer Hochschulen vergleichbar macht. Die Evaluation wurde online durchgeführt und beinhaltete Fragen zum allgemeinen organisatorischen Ablauf, zur Didaktik sowie eine Selbsteinschätzung der Studierenden bezüglich der eigenen Kompetenzentwicklung infolge der Lehrveranstaltung. Es zeigte sich eine sehr geringe Affinität der Studierenden zu Software Engineering. Dennoch geben die meisten Studierenden an, in dieser Lehrveranstaltung viel gelernt zu haben. Als positiv wurde die Praxisnähe der Projektarbeit bewertet, was aber gleichzeitig aus studentischer Sicht den schwierigsten Teil darstellte. Insbesondere die Zusammenarbeit in der Gruppe lässt auf weiteres Entwicklungspotential bezüglich der Teamkompetenzen

schließen. So wird im nächsten Durchgang dieser Lehrveranstaltung die Teamarbeit noch intensiver begleitet, moderiert und unterstützt, so dass der Trainingseffekt bei den Studierenden noch wächst.

4 Zusammenfassung und Ausblick

Das dreistufige Lehr-Lern-Konzept aus Vorlesung, Kleingruppen- und Projektarbeit eignet sich gut, um sowohl fachliche Kompetenzen zu vermitteln als auch Soft Skills zu trainieren. Zur systematischen Weiterentwicklung reichen jedoch Lehrveranstaltungsevaluationen alleine nicht aus. Im Anfang 2012 gestarteten interdisziplinären Forschungsprojekt EVELIN (Experimentelle Verbesserung des Lernens von Software Engineering) arbeiten sechs bayerische Hochschulen an der systematischen kompetenzorientierten Weiterentwicklung der Software-Engineering-Ausbildung an Hochschulen. Ein wesentlicher Aspekt ist die Frage, wie bereits im Studium ein breites Spektrum an fachlichen und Schlüsselkompetenzen möglichst realitätsnah trainiert werden kann. Dazu sind innovative didaktische Ansätze notwendig, die gezielt und auf breiter Basis evaluiert, verglichen und optimiert werden. Das Forschungsdesign ist in [Se12] beschrieben. Für das vorgestellte Lehr-Lern-Konzept bedeutet dies, dass zunächst eine systematische Erhebung der Soll-Kompetenzen für Software Engineering im Studiengang Mechatronik stattfindet. In Verbindung mit den Lehrzielen der Dozenten wird daraus eine Definition gelungenen Lernens [Be09], die diejenigen fachlichen und überfachlichen Kompetenzen beschreibt, die während des Studiums erreicht werden sollen. Auf dieser Grundlage werden dann die Einflussvariablen auf den Lernprozess systematisch analysiert und so Hypothesen und Theorien über ein besseres Verständnis der Lernprozesse im Software Engineering gebildet. Im Rahmen von EVELIN wird dann das beschriebene Lehr-Lern-Konzept auf seine Zielerreichung hinsichtlich der Soll-Kompetenzen noch passgenauer evaluiert und weiterentwickelt.

Literatur

- [Be09] Bender, W.: Wie kann Qualitätsentwicklung aus dem Pädagogischen heraus entwickelt werden? – Pädagogische Reflexivität. In (Dehn, C., Hrsg.): *Pädagogische Qualität*. Expressum-Verlag, Hannover, 2009; S. 69–77.
- [Fi08] Figl, K., Motschnik, R.: Researching the Development of Team Competencies in Computer Science Courses. In: *Proceedings of the 38th ASEE/IEEE Frontiers in Education Conference (FIE 2008)*, 2008; S. S3F-1–S3F-6. – DOI: 10.1109/FIE.2008.4720296
- [Gr08] Grzega, J. et al.: The didactic model LdL (Lernen durch Lehren) as a way of preparing students for communication in a knowledge society. In: *Journal of Education for Teaching* Heft 38, 2012; S. 167–175.
- [Ju10] Jung, E.: *Kompetenzerwerb – Grundlagen, Didaktik, Überprüfbarkeit*. Oldenbourg Wissenschaftsverlag, München, 2010.
- [Ma02] Martin, J.-P.: Lernen durch Lehren. In: *Die Schulleitung – Zeitschrift für pädagogische Führung und Fortbildung in Bayern* 29(4), Dezember 2002; S. 3–9.
- [Pr10] Predoiu, L.: Didaktik und Bewertung in längerfristigen Teamprojekten in der Hochschullehre. In (Diethelm, I., Dörge, C., Hildebrandt, C., Schulte, C., Hrsg.): *Didaktik der In-*

formatik – Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik; 6. Workshop der GI-Fachgruppe DDI. LNI P-168. Bonn, 2010; S. 113–118.

- [Si97] Siebert, H.: *Didaktisches Handeln in der Erwachsenenbildung – Didaktik aus konstruktivistischer Sicht*. 2. Auflage. Luchterhand, Neuwied, 1997; S. 103–110.
- [Se12] Sedelmaier, Y.; Landes, D.: A Research Agenda for Identifying and Developing Required Competencies in Software Engineering. In: *Proceedings of the 15th International Conference on Interactive Collaborative Learning (ICL 2012)*. IEEE, 2012.
- [Vi11] Vigerschow, U., Schneider, B., Meyrose, I.: *Soft Skills für Softwareentwickler – Fragetechniken, Konfliktmanagement, Kommunikationstypen und -modelle*. 2. Auflage. dpunkt-Verlag, Heidelberg, 2011.
- [We08] Weicker, N.; Weicker, K.: Analyse des Kompetenzerwerbs im Softwarepraktikum: In (Schwill, A., Hrsg.): *Hochschuldidaktik der Informatik, HDI2008 – 3. Workshop des GI-Fachbereichs Ausbildung und Beruf / Didaktik der Informatik*. Universitätsverlag Potsdam, 2009; S. 93–104.
- [We92] Weinberg, J.: Didaktische Reduktion und Rekonstruktion. In: *Studienbibliothek für Erwachsenenbildung, Band 2: Didaktische Dimensionen der Erwachsenenbildung*. Päd. Arbeitsstelle des DVV, Frankfurt, 1992; S. 130–150.
- [Wi91] Wilsdorf, D.: *Schlüsselqualifikationen – die Entwicklung selbständigen Lernens und Handelns in der industriellen gewerblichen Berufsausbildung*. Lexika-Verlag, München, 1991.
- [Zu06] Zumbach, J.: Problemorientiertes Lernen im Hochschulunterricht – Selbstgesteuertes Lernen anhand authentischer Probleme. – URL https://www.sbg.ac.at/mediaresearch/zumbach/download/1999_2006/book_chapters/Zumbach_Beitrag.pdf (06/2012)