

Software Engineering im Wettbewerb

Industrienah Softwareentwicklung mit Studierenden

Andreas Hartmann

Hochschule für Telekommunikation - Leipzig
Leipzig, Deutschland
hartmann@hft-leipzig.de

Sabine Radomski

Hochschule für Telekommunikation Leipzig
Leipzig, Deutschland
radomski@hft-leipzig.de

Abstract—Software developer must provide methodical competences as well as social skills. However, standard teaching concepts are often not suitable to impart that knowledge. We introduce an agile concept based on a challenge of industrial style software projects, where students autonomously train desirable competences and skills. Written exams are replaced with a quality assurance of the projects and their prototype solutions.

I. EINLEITUNG

Voruntersuchung, Analyse, Design, Implementierung und Test – oft werden Studierende in den Kompetenzen des Software Engineering (SWE) nach einem klassischen Modell ausgebildet. Dabei geht es um theoretische Inhalte oder einfach nur die Phasen des SWE nach Balzert [1]. In der Industrie haben dagegen agile Methoden das Ruder übernommen. Mehrere Iterationen und eine starke Rückkopplung mit den Nutzern sind die Merkmale von z.B. SCRUM [2]. Umso wichtiger wird es für die Lehrenden, die Kompetenzen des SWE am realistischen Beispiel zu trainieren. Das hier gezeigte Modul stellt einen Wettbewerb mit industrienahen Aufträgen in den Vordergrund. Es orientiert sich am Flipped Classroom [3] und richtet sich an einer selbstbeschreibenden Dokumentation aus.

II. GRUNDLAGEN

SWE im Bachelor Studiengang hat die Aufgabe, den Studierenden die grundlegende Fachkompetenz und Methoden zu vermitteln, die für die erfolgreiche Durchführung von Projekten notwendig sind [4, 5]. Zu den erlernten Fachkompetenzen zählen Qualitätsmerkmale von Software, Vorgehensmodelle, die Unified Modeling Language, Phasen, Architekturen sowie Auswirkungen der agilen Softwareentwicklung auf die Qualität der Software und die Projektdauer. Die Studierenden erarbeiten sich zudem Kompetenzen bei der Anwendung von entsprechenden Entwicklungswerkzeugen.

III. DAS MODUL SOFTWARE ENGINEERING

A. Wettbewerb mit realen Kunden

Anstelle von fiktiven Projektaufträgen treten reale Industriekunden [6]. Die HfTL hat hierzu Partnerprojekte gegründet, insbesondere mit der Deutschen Telekom. Mit den Partnern werden Innovationen und Ideen [7, 8] erprobt. Im Ergebnis entstehen Prototypen, die bei Bedarf zu Produkten finalisiert werden. Studierende erhalten einen Einblick in reale Projekte und Herausforderungen in der Industrie. Aufgrund interdisziplinärer Aufgabenstellungen profitiert das Modul von einer

enormen fachlichen Vielfalt. Aktuelle Themen sind autonomes Fahren und 5G Mobilfunk. Als Motivation und in Kooperation mit den Partnern wird jedes Semester ein Wettbewerb ausgetobt. Die Studierenden – organisiert in Teams – treten mit ihren Lösungen gegeneinander an und die besten 3-5 Teams gewinnen Preise. Diese können mit der beruflichen Perspektive verbunden sein, wie z.B. kostenlose Zertifizierungen.

B. User Centered Design (UCD)

Tatsächlich lassen sich die gewünschten Kompetenzen durch eine Adaption von agilen Modellen besser trainieren. Studierende haben zu Beginn die Wahl zwischen SCRUM, Kanban oder dem Spiralmodell (auch V-Modell und Wasserfall sind möglich). Nach einer Einführung auf *Foundation Level* beginnen die Studierenden ihre Projekte mit Fokus auf Agilität und Nutzerzentrierung. Dabei konnte bereits eine enorme Kreativität und Motivation der Studierenden beobachtet werden. Unterschiedliche Rollen im Team sorgen für eine faire Verteilung der Fähigkeiten. Der Wettbewerb wird hierbei ebenfalls unterstützt, da frühzeitig Lösungen gezeigt werden können. So kreieren die Teams in den ersten 6 Wochen bereits Wireframes und Low-Fidelity-Prototypen. Später folgen High-Fidelity-Prototyp und finale Lösung. Mit jedem Meilenstein werden die Lösungen vorgestellt und in der gesamten Semestergemeinschaft verglichen. Das beste Team qualifiziert sich für eine bessere Position im Wettbewerb, während andere Teams aufholen oder sich verbessern können.

C. Dokumentation und Qualität

Grundlage der Dokumentation bilden nummerierte Templates und Checklisten – sie führen das Team durch das Projekt. Im Idealfall werden die Templates nacheinander bearbeitet und mit den Checklisten überprüft. Einige Dokumente werden iterativ in jeder Phase aktualisiert. Die Studierenden lernen anhand der Templates professionell mit Dokumenten umzugehen (z.B. Änderungsverzeichnisse). Die Verwaltung der Dokumente erfolgt parallel zu den Sourcen im GIT. Mit den Templates wird eine vergleichbare Dokumentation des Projektfortschritts [9] und der Qualität ermöglicht. Letzteres ist besonders wichtig, da sich hier die Kompetenzen der Studierenden spiegeln. Je nach Aufgabe und Rolle im Projekt kann die Qualität bewertet und somit die Kompetenz eingeschätzt werden, wobei auf Industriestandards (vgl. ISO 9126, ISO 25010) zugegriffen wird.

D. Bewertung nach Rollen und Kompetenzen

Eine Herausforderung bei Projekten stellt die Bewertung dar. Zu leicht können sich Einzelne in der Gruppe verstecken

oder herausragende Leistungen unsichtbar bleiben [10]. Begegnung wird der Situation zunächst mit einem Rollenkonzept. Für jedes Vorgehensmodell existiert eine Liste mit Rollen, die die Studierenden zuordnen müssen. Über Übungen wird mit den Studierenden zu Beginn des Semesters die optimale Rolle identifiziert. Mehrfachrollen sind ebenso möglich wie geteilte Aufgaben. Die Lehrenden achten auf eine möglichst gleiche Verteilung. Teams mit 10-15 Personen zeigen sich effektiv und somit sind auch größere Semestergruppen erfolgreich abbildbar. Im Semester wird mit den Studierenden über ihre Rolle und die zu verantwortenden Aufgaben diskutiert. In der Dokumentation, im Projektablauf und im Produkt wird den Studierenden genau gezeigt, für welche Qualität sie verantwortlich sind. Dabei müssen die Teams ihre Projekte in den Präsenzveranstaltungen regelmäßig vorstellen und Studierende in ihrer Rolle die Fragen der Lehrenden beantworten. Die Lehrenden können hier optimal eine Rückkopplung zum Selbststudium herstellen. Entsprechend der kommunizierten Schwerpunkte werden am Ende des Semesters die Projekte selbst, die Dokumentation und das Produkt ausgewertet. Während der Bewertung entsteht sowohl ein Eindruck der Teamleistung als auch der einzelnen Rollen. Entsprechend werden die Teilnoten vergeben und miteinander kombiniert. Die Notenvergabe ist transparent und wird zu Beginn des Semesters kommuniziert.

E. Anlehnung an Flipped Classroom

Durch eine Fokussierung auf Projekte und die Arbeit im Team muss im Modul SWE für die theoretischen Grundlagen eine alternative Lösung angewendet werden. Daher wurde ein Ansatz gewählt der am Flipped Classroom angelehnt ist. Die Vertiefung der Themen erfolgt dabei dreistufig und kann sich über das gesamte Semester erstrecken. Als ersten Einstieg erhalten die Studierenden über eine eLearning Plattform (ILIAS) Zugang zu Tutorials. Dabei handelt es sich um 5-12 min Videos. Mit den Tutorials sind die Studierenden in der Lage, in den ersten Präsenzveranstaltungen mit den Lehrenden über die Projekte und Organisation des Semesters zu diskutieren. Wenn die Projekte starten, erhalten die Studierenden Zugang zu einem interaktiven Lernmodul. Die Grundlagen werden vertieft und mit Fragen und Übungen das Wissen verfestigt. In den Präsenzstunden finden dann die Diskussionen mit den Lehrenden statt, wo die Studierenden weiter die Anwendung der Theorie im praktischen Projekt trainieren. An spezifischen Stellen werden die Studierenden aufgefordert, konkrete Fragestellungen zu bearbeiten und die Ergebnisse in Form von neuen Tutorials bereitzustellen. Hierfür müssen sie zusätzliche Literatur studieren.

F. Der Wettbewerb

Zur zusätzlichen Motivation der Studierenden treten die Teams in einem Wettbewerb gegeneinander an. Über 200 Studierende in 18 Teams haben 2017 an dem Wettbewerb teilgenommen. Die Jury wird durch die Mitarbeiter des Bereiches SWE der HfTL gebildet, da innerhalb von 2 Tagen 18 Präsentationen zu bewerten sind. Im Gegensatz zur Bewertung der

Modulleistung werden beim Wettbewerb die Kriterien Schwierigkeitsgrad, Innovation, Nutzen und Kundenzufriedenheit zur Bestimmung der besten Teams genutzt. Die Bewertung erfolgt nach einem Punktesystem. Die Bewertung für den Wettbewerb berücksichtigt dabei hauptsächlich die Präsentationen der Projekte, so kann auch der Teamzusammenhalt eingeschätzt werden. Als Preise werden gesponserte Sachpreise und Workshops für Penetration Test oder Design Thinking vergeben.

IV. DISKUSSION UND AUSBLICK

Das Lehrmodul SWE an der HfTL stellt auf einem Wettbewerb basierende Projekte in den Vordergrund der Lehre. Die zugehörigen Projektaufträge stammen aus Kooperationen mit Industriepartnern. Die Studierenden lernen ebenso mit agilen Methoden umzugehen, wie die Kompetenzen im Team trainiert werden. Die Projekte werden durch eine ablauforientierte Dokumentation unterstützt, welche die Studierenden durch das Semester führt. Checklisten sorgen für eine kontinuierliche Prüfung der Qualität, welche gleichzeitig die Basis für die spätere Bewertung der Kompetenzen darstellt. Individuelle Leistungen werden an Rollen in den Projekten gekoppelt. Die theoretischen Grundlagen werden im Selbststudium vermittelt, wobei nach einem mehrstufigen Modell auf Video-Tutorials und eLearning Lernmodule zurückgegriffen wird.

Die Annahme seitens der Studierenden wird sehr positiv bewertet. Über Fragebögen und Lehrevaluationen melden die Studierenden eine hohe Motivation und Abwechslung im Studium zurück. Zudem befürworten sie das professionelle Umfeld mit den Partnern. Die Qualität der Bewertung hängt sehr stark von einem stabilen und transparenten Rollenmodell ab.

REFERENCES

- [1] H. Balzert, Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, 3. Aufl., Spektrum Akademischer Verlag, 2011.
- [2] B. Gloger, Scrum – Produkte zuverlässig und schnell entwickeln, 4. Aufl., Carl Hanser Verlag München, 2013.
- [3] A.J. Boevé, R.R. Meijer, R.J. Bosker et al., High Educ (2017) 74: 1015. <https://doi.org/10.1007/s10734-016-0104-y>.
- [4] F. Fagerholm, M. Kuhrmann, J. Münch, Guidelines for using empirical studies in software engineering education, PeerJ Computer Science, vol.3, p.e131, 2017.
- [5] N. Pratheesh, T. Devi, Necessity of learning analytics in software engineering education, Journal of Engineering Science and Technology, vol.10(3), pp.269-281, 2015.
- [6] Y. Liguó, Overcoming challenges in software engineering education: delivering non-technical knowledge and skills, IGI Global, 2014.
- [7] Ch.-Y. Chen, P.P. Chong, Software engineering education: A study on conducting collaborative senior project development, The Journal of Systems & Software, vol.84(3), pp.479-491, 2011.
- [8] P. Henning, eLearning 2015. Stand der Technik und neueste Trends, HMD Praxis der Wirtschaftsinformatik, vol.52, pp.132-143, 2015.
- [9] E. Bjarnason, K. Smolander, E. Engström, P. Runeson, A theory of distances in software engineering, Information and Software Technology, vol.70, pp.204-219, 2016.
- [10] H.-W. Schönell, Übersicht Lehr- und Lernmethoden, <http://www.hwschoenell.de/xdocs/Lehr-%20und%20Lernmethoden>, 12.12.2017.