
ArchiLab

Lehr- und Lernplattform für die gemeinschaftliche Erarbeitung von Software-Architektur in großen Gruppen

Prof. Dr.-Ing. Stefan Bente

Inhalt

1	Problemstellung.....	1
	1.1 Persönliche Motivation	1
	1.2 Problemstellung in der Hochschullehre.....	1
	1.3 Lösungsansatz der Lehrinnovation „ArchiLab“.....	2
	1.4 Geplantes Einsatzgebiet von ArchiLab	3
2	Management von Learning Outcomes	3
	2.1 Geplante Umsetzung in ArchiLab.....	4
3	Fallstudien-Bearbeitung	5
	3.1 Geplante Umsetzung in ArchiLab.....	6
	3.2 Vernetzung mit dem Learning Outcome der Veranstaltung (Kapitel 2)	7
4	Onlinetool zur Lernstandskontrolle.....	7
	4.1 Geplante Umsetzung in ArchiLab.....	8
	4.2 Vernetzung mit dem Learning Outcome der Veranstaltung (Kapitel 2)	8
5	Langfristige Planung für ArchiLab	9
	5.1 Verstetigung der Lehrinnovation	9
	5.2 Übertragung auf andere Fächer und Disziplinen	9
	5.3 Erfolgs- und Risikobewertung in der Erprobung	10
6	Referenzen	10

1 Problemstellung

1.1 Persönliche Motivation

Die meiste Zeit meines beruflichen Lebens habe ich als Software-Architekt verbracht, als Planer und Gestalter großer IT-Systeme oder -Landschaften. Der Software-Architekt entwirft die Blaupausen dieser Systeme. Software-Applikationen müssen nach den Regeln der Technik umgesetzt sein und den Wünschen des Auftraggebers entsprechen. Der Architekt muss aber auch zukünftige Anforderungen antizipieren, um das IT-System nachhaltig gestalten zu können – nur so kann es leicht gewartet und erweitert werden.

Hierzu hält die Softwaretechnik einen großen und sehr komplexen Werkzeugkasten bereit – Modellierungssprachen wie UML, Design-Patterns und Architekturstile, Entwurfsmethoden, etc. Diese beherrscht ein erfahrener Architekt so flüssig und intuitiv wie ein Konzertmusiker sein Instrument. Andernfalls könnte er sich nicht genügend auf seinen Entwurf konzentrieren.

Darüber hinaus muss der Architekt aber auch noch an der Umsetzung seines Entwurfs mitwirken. Er muss also in der Lage sein, seinen hochkomplexen Entwurf so zu kommunizieren, dass sowohl Ausführende (Softwareentwickler, Betriebsexperten, etc.) wie auch nicht IT-affine Auftraggeber ihn verstehen. Häufig verlaufen solche fachliche Diskussionen kontrovers. Kommunikationsvermögen ist also neben analytischen Fähigkeiten eine Schlüsselkompetenz.

Ich habe viele Jahre Teams von Softwarearchitekten geleitet und damit auch an der Ausbildung von Architekten mitgewirkt. Die Frage, was genau einen guten Software-Architekten ausmacht, und wie man die nötigen Fähigkeiten am besten vermittelt, beschäftigt mich also schon lange.

1.2 Problemstellung in der Hochschullehre

In der Berufswirklichkeit wird man erst nach einigen Jahren praktischer Erfahrung als Software-Entwickler zum Architekten. In der Hochschullehre müssen die Grundlagen der oben beschriebenen Kernfähigkeiten aber schon ab dem 3. Semester vermittelt werden, beispielsweise in der Veranstaltung Softwaretechnik.

Die klassische Softwaretechnik tendiert dabei dazu, sich auf die Lehre des Werkzeugkastens (UML-Modellierung, Liste der Entwurfsmuster, ...) zu fokussieren. Drängende Fragen der Studierenden nach dem Kontext der Anwendung – „Wie setze ich das alles ein? Was nutze ich wann? Womit fange ich an, wenn ich einen neuen Entwurf erstelle?“ – werden in den klassischen Lehrwerken oft nur unzureichend beantwortet. Stattdessen behandeln die Autoren die einzelnen Werkzeuge (wie etwa UML-Diagrammtypen) eins nach dem anderen in großer Detailtiefe.

Diese Art der Lehre bewegt sich hauptsächlich auf Stufe 2-3 (Understanding, Applying) der *Revised Bloom's Taxonomy* [1]. In der praktischen Anwendung im Projekt sind aber eher die Stufen 4-6 gefordert (Analysing, Evaluating, Creating). Wenn also ein in dieser Form ausgebildeter Student nach Ende des Studiums an die eigenverantwortliche Umsetzung geht, hat er vermutlich 80% der Details bereits vergessen. Bei dem Rest an verbliebenem Wissen fehlt ihm dann oft der Kontext der praktischen Anwendung. Dies führt in der Praxis häufig dazu, dass die Entwurfsphase ausgelassen oder das Softwaresystem mit Hilfe einer unvollkommenen Ad-hoc-Modellnotationen spezifiziert wird. Das Risiko von unvollständigen und inkonsistenten Softwareentwürfen steigt dadurch deutlich.

1.3 Lösungsansatz der Lehrinnovation „ArchiLab“

Das oben genannte zentrale Problem der Softwarearchitektur-Ausbildung lässt sich nach dem Ansatz von Middendorf und Pace [4] lösen. In ihrem Artikel *Decoding the Disciplines* schlagen sie einen siebenstufigen Zyklus für die Vermittlung komplexen Expertenwissens vor (siehe Abbildung 1).

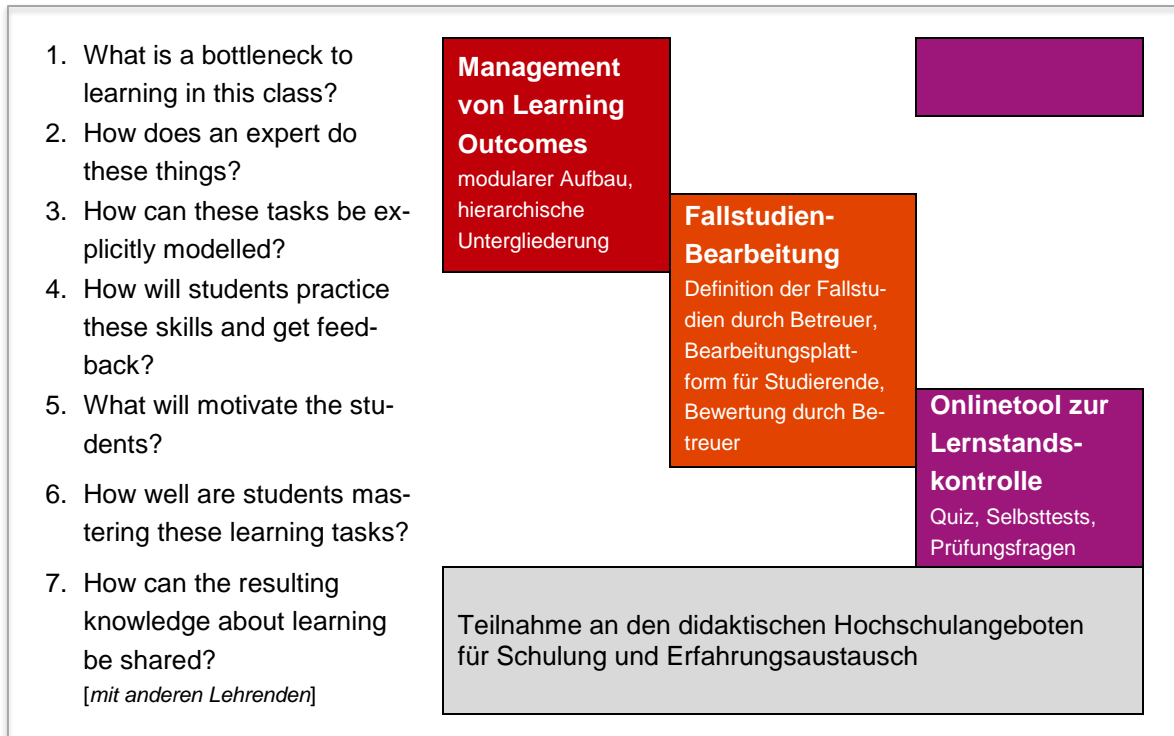


Abbildung 1: Lernschritte nach Middendorf und Pace [4] und deren Abdeckung durch die vorgeschlagenen Lehrinnovationen

Die Informatik-Bachelor-Veranstaltung *Softwaretechnik*, die vornehmlich Architekturwissen vermittelt, nutzt drei didaktische Kernelemente, um diesen Ansatz umzusetzen:

- *Management von Learning Outcomes*¹: modularer Aufbau und hierarchische Untergliederung der Veranstaltung in Form von (Teil-)Fähigkeiten
- *Fallstudien-Bearbeitung*: Kollaborative, projektorientierte Erarbeitung von Softwarearchitekturen für eine Fallstudie, zur praktischen Erprobung der erlernten Kompetenzen
- *Onlinetool zur Lernstandskontrolle*: Quiz, Selbsttests, Verwaltung von Prüfungsfragen

In der Veranstaltung *Softwaretechnik* werden diese Mittel schon jetzt erfolgreich eingesetzt, allerdings jeweils nur in einer stark vereinfachten, manuellen Form. Dadurch wird ihr Potenzial bei weitem nicht ausgeschöpft. Darüber hinaus ist die Veranstaltung durch den hohen manuellen Aufwand sehr betreuungsintensiv.

ArchiLab ist die Vision einer digitalen und vernetzten Plattform zur Lehr- und Lernunterstützung gemäß des Ansatzes von *Decoding the Disciplines*, fokussierend auf die drei genannten Kernfunktionalitäten. Mittels einer Projektförderung könnte ArchiLab umgesetzt und in die praktische Lehre eingeführt werden.

¹ Aus dem *Constructive Alignment* nach [2].

In den nachfolgenden Kapiteln werden die drei Teilmodule von ArchiLab sowie ihr jeweiliger Bezug zu den ersten sechs der „*Decoding the Disciplines*“-Stufen näher erläutert.

1.4 Geplantes Einsatzgebiet von ArchiLab

Zunächst soll ArchiLab in der zweisemestrigen Veranstaltung *Softwaretechnik* eingesetzt werden, einer Pflichtveranstaltung im 3. und 4. Semester des Informatik-Bachelor-Studiums an der TH Köln.

Eine Ausweitung von ArchiLab auf die Pflichtmodule *Mensch-Computer-Interaktion* und *Datenbanken* der Bachelor-Studiengänge Informatik und Wirtschaftsinformatik (ebenfalls zweisemestrig im 3. und 4. Semester) liegt nahe, da die Praktika dieser Veranstaltungen schon jetzt miteinander koordiniert sind.

Zukünftig wird innerhalb des Instituts für Informatik an der TH Köln angestrebt, die Praktika dieser drei Pflichtmodule noch stärker zu verzahnen. Einerseits senkt dies angesichts der hohen Studierendenzahlen den Betreuungsaufwand, zum anderen ergibt sich ein realistischerer Projekteindruck für die Studierenden, wenn Architektur des Gesamtsystems, Nutzerschnittstelle und Datenhaltung gleichzeitig und mit gemeinsamer Aufgabenstellung spezifiziert werden.

In einem solchen Szenario können dann Learning Outcomes, Praktikums-Fallstudien und Online-Fragenpool dieser drei Informatik-Kernfächer gemeinsam in ArchiLab verwaltet werden.

2 Management von Learning Outcomes

Gemäß Middendorf und Pace beginnt der Lehr- und Lernprozess bei der Frage nach Lernhindernissen (*Bottlenecks*) für Studierende (Schritt (1) in Abbildung 1). Beim Erlernen von Softwarearchitektur-Fähigkeiten kann das beispielsweise die Frage sein, mit welchen konkreten Schritten man beim Entwurf denn beginnt. Eine Unsicherheit an dieser Stelle wirkt typischerweise als Blockade.

In den Schritten (2) und (3) werden Hemmnisse durch den Lehrenden mittels einer dekompositionischen Analyse der Kompetenzen aufgelöst: Wie macht der Experte das? Welche expliziten Schritte unternimmt er? Wie kann man diese weiter in Teilschritte zerlegen, damit auch ein Novize geführt diesen Weg gehen kann?

Das Learning Outcome aus dem *Constructive Alignment* [2] ist das ideale Beschreibungswerkzeug für diese hierarchisch untergliederten Kompetenzen. Statt nur ein einziges, globales Learning Outcome zu schreiben, kann die Veranstaltung auch durch eine Menge von rekursiv verschachtelten Learning Outcomes definiert werden, welche die zu erwerbenden Kompetenzen sehr feingranular beschreiben.

In der jetzigen Veranstaltung wurde dieser Ansatz in statischer Form (Excel) punktuell durchgespielt. Das Learning Outcome folgt dabei einer festen formalen Form:

Als <Rolle> kann ich <Fähigkeit>, vorausgesetzt <Vorbedingungen>, indem ich <Teilfähigkeiten> und abschließend <abschließende Teilfähigkeit>, so dass <Nutzen der Fähigkeit>.

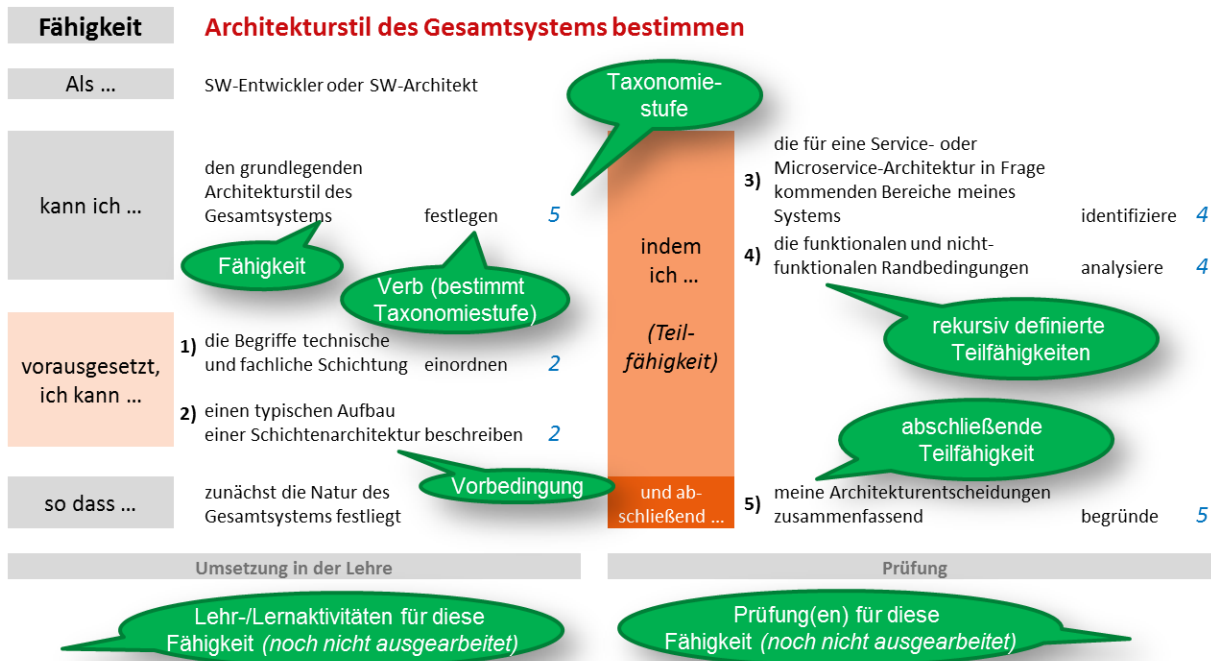


Abbildung 2: Beispiel für eine (Teil-)Fähigkeit im Architekturentwurfsprozess, in Form einer rekursiv aufgebauten und untergliederten Learning Outcomes

Abbildung 2 zeigt als Beispiel eine typische Teilfähigkeit zu Anfang des Architekturentwurfs, die Bestimmung des sogenannten Architekturstils. Die Teilfähigkeiten (wie etwa 4) *die funktionalen und nichtfunktionalen Randbedingungen bestimmen*) sind ihrerseits wieder (Sub-) Learning Outcomes, die dann in der gleichen Form spezifiziert werden.

Auf diese Weise kann der Lehrende, den Schritten (2) und (3) von *Decoding the Disciplines* folgend, die Zielkompetenzen seiner Veranstaltung spezifizieren. Es entsteht auf diese Weise ein feingranulares, inhaltlich-didaktisches Gerüst.

2.1 Geplante Umsetzung in ArchiLab

In ArchiLab ist geplant, einen Editor für diese rekursiv untergliederten Learning Outcomes zu implementieren. Neben einer flexibleren Erstellung gewinnt man so die Möglichkeit, Teilfähigkeiten mit praktischen Übungen (siehe Kapitel 3) sowie mit Prüfungs- und Selbsttestfragen (Kapitel 4) zu verknüpfen.

Auf diese Weise kann die Frage (1) nach Lernhindernissen (*Bottlenecks*) gezielt beantwortet werden. Übungen und Prüfungs- / Selbsttestfragen, die von Studierenden mit unterdurchschnittlichem Erfolg bearbeitet werden, weisen auf schlecht vermittelte Teilfähigkeiten hin. Deren Lehr- / Lernformat kann dann für die nächsten Iteration der Veranstaltung überarbeitet werden.

Als Plattform für das Management von Learning Outcomes bietet sich das Konzept des semantischen Wikis [5] an. Diese verbindet die freie Eingabe von Inhalten mit der Möglichkeit, diese Inhalte durch ein dynamisch erweiterbares Datenmodell vorzustrukturieren. Zahllose Editor-Funktionalitäten sind bereits vorhanden und müssen nicht mehr implementiert, sondern nur durch das passende Datenmodell konfiguriert werden.

Ein Kandidat für die ArchiLab-Plattform ist das als Open Source veröffentlichte, webbasierte semantische Wiki SocioCortex [7] der TU München. Dessen Eignung wird vorab noch durch eine Technologieevaluierung überprüft werden (siehe Arbeitsplan).

3 Fallstudien-Bearbeitung

Die Stufen (3) – (5) des „*Decoding the Disciplines*“-Ansatzes (Abbildung 1) sehen vor, dass auf die explizite Definition der Arbeitsschritte (Stufe 3) das praktische Einüben dieser Tätigkeiten (4) folgt. Das hat wiederum eine Motivation (5) durch das Erleben eigenen Könnens und der Relevanz des Gelernten zur Folge.

Es ist also unerlässlich, den Studierenden die Anwendung der erlernten Methoden und Werkzeuge im Rahmen ihres Studiums zu ermöglichen. Daher besteht die Veranstaltung Softwaretechnik neben der Vorlesung auch aus einem projektbasierten Praktikum. Hier wird ein Projektgegenstand – ein zu erstellendes fiktives Softwaresystem – mit möglichst hohem Realitätsbezug definiert. Dieses wird in Teilbereiche zerlegt und von den Studierenden in Gruppen bearbeitet. Das Ergebnis des Praktikums ist dann pro Gruppe ein Softwarearchitekturentwurf für das zugeordnete Teilsystem.

Wie schon in Kapitel 1.1 ausgeführt, beinhaltet der Architekturentwurf immer ein hohes Maß an Kommunikation. Dies gilt nicht nur für die Vermittlung des fertigen Entwurfs an die ausführenden Entwickler, sondern auch für die Entwurfsphase selbst. Gerade in sehr großen Softwaresystemen sind unzählige Abhängigkeiten und Schnittstellen zu modellieren, die jeweils mit den zuständigen Architekten oder Teams abgestimmt werden müssen.

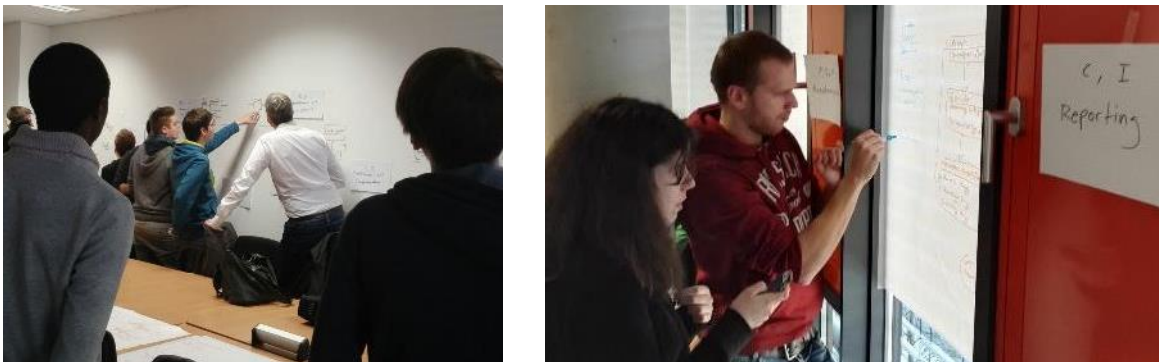


Abbildung 3: Moderierter Schnittstellen-Abstimmungstermin mehrerer Praktikumsgruppen in Rahmen des Softwaretechnik-Praktikums

Ein Großteil der Entwurfsarbeit im Praktikum besteht daher für die Studierenden darin, die relevanten Aspekte ihres jeweiligen Teilbereichs in Form von UML-Diagrammen zu modellieren. Hierbei ist ein wichtiges Kriterium, dass die Schnittstellen zwischen den einzelnen Subsystemen klar definiert und die Zuständigkeiten für einzelne Geschäftsobjekte und -prozesse eindeutig geklärt werden. Während die Modellierung *innerhalb* der Praktikumsgruppe stattfindet, finden Bewertung und Abstimmung der Schnittstellen im kommunikativen Austausch *zwischen* den Praktikumsgruppen statt – wie im realen Projekt im späteren Berufsleben.

Die Veranstaltung Softwaretechnik bildet dies unter anderem dadurch ab, dass zu ein bis zwei Praktikumsterminen im Semester sich alle Gruppen gemeinsam treffen (siehe Abbildung 3). Sämtliche potenziellen Schnittstellen zwischen Teilsystemen werden durch beschriftbare elektrostatische Wandfolien repräsentiert, an denen die Studierenden ein gemeinsames Verständnis der jeweiligen Schnittstellenmodellierung erarbeiten können.

Dies simuliert im Praktikumskontext, dass ein Softwarearchitekt seine Modellierarbeit nicht nur allein am Rechner erledigen sollte. Die Architekturqualität ist deutlich höher, wenn der erste Entwurf gemeinsam am Whiteboard erarbeitet wird. Dort können die Teammitglieder lebhaft und kontrovers diskutieren und neue Ideen rasch ausprobieren und wieder verwerfen.

Um hier eine vollständige Abstimmung unter den einzelnen Gruppen und eine Bewertung des erreichten Abstimmungsgrades zu erreichen, sind im aktuellen Veranstaltungsformat mehrere zusätzliche Termine mit den Praktikumsgruppen notwendig. Die Abstimmungsqualität wird in einem aufwändigen Verfahren ermittelt, bei dem die Studierenden sich selbst einschätzen. Dem wird eine aus mehreren Einzelkriterien errechnete Einschätzung der Praktikumsbetreuer entgegengesetzt.

	H - Preisengine und Speisekarte	I - Reporting	J - Bestellungen (Gerichte) und Gastperspektive	K - Bestellungsbearbeitung und Kochperspektive	L - Rezeptmanagement	M - Rezeptmanagement
G - Bestands und Lieferantenverwaltung	1.0	1.0		2.0	1.0	1.0
H - Preisengine und Speisekarte		2.0	2.0		1.0	1.0
I - Reporting			2.0	2.0	2.0	1.0
J - Bestellungen (Gerichte) und Gastperspektive				1.0	2.0	2.0
K - Bestellungsbearbeitung und Kochperspektive					1.0	1.0

	H - Preisengine und Speisekarte	I - Reporting	J - Bestellungen (Gerichte) und Gastperspektive	K - Bestellungsbearbeitung und Kochperspektive	L - Rezeptmanagement	M - Rezeptmanagement
G - Bestands und Lieferantenverwaltung	2.0	1.7		2.3	2.3	2.3
H - Preisengine und Speisekarte		2.3	2.3		1.3	1.3
I - Reporting			2.1	2.0	2.0	2.0
J - Bestellungen (Gerichte) und Gastperspektive				2.3	2.0	2.3
K - Bestellungsbearbeitung und Kochperspektive					1.7	2.0

Abbildung 4: Bewertung der Abstimmungsqualität der Schnittstellenspezifikationen zwischen den Praktikumsgruppen.
Oben: Selbsteinschätzung der Studierenden (gemittelt aus den Einzeleinschätzungen der Mitglieder beider beteiligten Teams)
Unten: Einschätzung durch Betreuer (gemittelt aus mehreren Einzelkriterien)

Abbildung 4 zeigt einen Ausschnitt der so ermittelten Bewertung (der Praktikumsgegenstand war ein Softwaresystem für eine Restaurantkette). Die Studierenden erhalten eine Rückmeldung, ob ihre Schnittstellenmodellierung und -abstimmung genügt, um die gewünschten Geschäftsprozesse und deren Datenaustausch zwischen verschiedenen Modulen der Software abzubilden.

3.1 Geplante Umsetzung in ArchiLab

In der jetzigen Veranstaltung findet dieser Prozess komplett manuell statt, was aufgrund der stetig wachsenden Anzahl an Studierenden mehr und mehr eine organisatorische Herausforderung darstellt. Auch können nicht annähernd alle relevanten Abstimmungsaspekte betrachtet werden.

In ArchiLab ist es geplant, dass die Fallstudie mit ihren Teilbereichen durch die Betreuer im System selbst definiert werden kann. Hier bietet sich als Plattform wieder das semantische Wiki an.

Damit können alle Modelldaten, die eine Praktikumsgruppe für den Architektorentwurf eines Teilmoduls zu spezifizieren haben, als Attribute im Datenmodell konfiguriert werden².

Die Praktikumsgruppen nutzen dann ArchiLab als webbasiertes Managementsystem für ihren Entwurf. Damit können dann sowohl Abstimmungsprozesse zwischen den Gruppen wie auch Bewertung der Entwurfs- und Abstimmungsqualität komplett innerhalb des Systems vorgenommen werden. Das gilt sowohl für Selbsteinschätzung wie auch für Bewertung durch Betreuer.

Die kollaborative Erstellung von Architekturentwürfen großer Systeme wird mit einem solchen System transparent gemacht, ohne zusätzlichen „technischen“ Betreuungsaufwand wie das manuelle Erstellen von Einschätzungs-Umfragen nötig zu machen. Die Betreuer können sich also ganz auf die fachliche Auseinandersetzung mit den Studierenden konzentrieren.

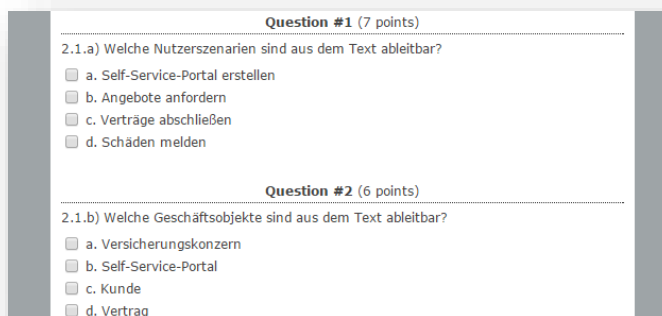
3.2 Vernetzung mit dem Learning Outcome der Veranstaltung (Kapitel 2)

Da das Management von Learning Outcomes (Kapitel 2) und Praktikums-Fallstudien zwei Teilmodule von ArchiLab auf der gleichen Plattform eines semantischen Wiki sind, ist eine effektive Vernetzung dieser Module leicht möglich.

Die Attribute des Entwurfs der Praktikumsgruppen lassen sich unmittelbar mit den Fähigkeiten verlinken, die zu ihrer Entstehung benötigt werden. Nehmen wir die Teilfähigkeit „Festlegung des Architekturstils“ aus Abbildung 2 auf Seite 4 als Beispiel: Hier gäbe es dann Attribute „Architekturstil“ und „Begründung des Stils“, die von den Praktikumsgruppen mit Inhalt gefüllt werden. Die Eigen- und Betreuerbewertungen geben dann Aufschluss darüber, wie gut die Teilfähigkeit „Festlegung des Architekturstils“ bereits beherrscht wird, und ermöglicht ein gezieltes Erkennen von Lehr- und Lerndefiziten.

4 Onlinetool zur Lernstandskontrolle

Der „*Decoding the Disciplines*“-Ansatz enthält als Stufe (6) die Frage „How well are students mastering these learning tasks?“. Feedback-Erhebungen unter Studierenden ergeben regelmäßig, dass sie die Überprüfung des eigenen Kenntnisstandes als wichtig, aber auch schwierig empfinden. Eine effektive Möglichkeit zur Selbsteinschätzung des Lernstandes wirkt auch gleichzeitig als Motivator für Studierende (Stufe 5). Zusätzlich helfen die aggregierten und ausgewerteten Ergebnisse natürlich auch den Lehrenden, Lernhindernisse zu erkennen (Stufe 1).



The screenshot shows a quiz interface with two questions. The first question is titled "Question #1 (7 points)" and asks "2.1.a) Welche Nutzerszenarien sind aus dem Text ableitbar?". It has four options: a. Self-Service-Portal erstellen, b. Angebote anfordern, c. Verträge abschließen, and d. Schäden melden. The second question is titled "Question #2 (6 points)" and asks "2.1.b) Welche Geschäftsobjekte sind aus dem Text ableitbar?". It has four options: a. Versicherungskonzern, b. Self-Service-Portal, c. Kunde, and d. Vertrag.

² Die eigentlichen Modelldiagramme sollen weiterhin in einem externen UML-Modellierungstool erstellt werden und nur als fertige Grafiken eingebunden werden. Die Erstellung eines kollaborativen UML-Editors wäre ein vielfach komplexeres Vorhaben mit einer ganz anderen Zielstellung als ArchiLab.

Abbildung 5: Online-Quiz zum Abschluss der Veranstaltung Softwaretechnik (Ausschnitt)

In der jetzigen Form der Veranstaltung Softwaretechnik wird das Tool Pingo der Universität Paderborn [2] für interaktive Wiederholungsfragen eingesetzt. Zusätzlich werden am Ende der Veranstaltung die Pingo-Fragen gesammelt, nochmals variiert sowie ergänzt und dann in der letzten Vorlesung als Online-Quiz mit direkter Auswertung durchgespielt (siehe

Abbildung 5). Die drei bestplatzierten Studierenden erhalten einen symbolischen Preis.

Das Feedback zur Veranstaltung zeigt eine hohe Wertschätzung der Studierenden für die beiden Einschätzungsmittel. Leider bedeutet insbesondere das Online-Quiz einen relativ hohen manuellen Vorbereitungsaufwand, da parallel zur Quizerstellung auch eine automatisch durchlaufende Powerpoint-Präsentation mit den teilweise komplexen Fragestellungen (die nicht in dem einfachen Quiztool allein abzubilden sind) erstellt werden muss.

Darüber hinaus ist es zurzeit kaum möglich, den Grad der korrekten Beantwortung der Fragen systematisch nachzuhalten. Die Anteile der richtigen Antworten müssten gesammelt und systematisch mit Bezug auf die jeweiligen Inhalte nachgehalten werden, was aus Zeitgründen unterbleibt.

4.1 Geplante Umsetzung in ArchiLab

Ein Online-Quiztool soll vor diesem Hintergrund fester Bestandteil von ArchiLab werden. Den Studierenden werden hier verschiedene Modi angeboten, von zufällig ausgewählten Fragen bis hin zu Fragen bestimmter Themengebiete oder Schwierigkeitsgrade. Die Anpassbarkeit des Quiz sorgt dafür, dass jeder Studierende sich individuell anhand seiner Stärken und Schwächen gezielt auf die Prüfung vorbereiten kann.

Die Studierenden sollen die Möglichkeit erhalten, den Fragenkatalog des Quiztools selbstständig zu erweitern. Die durch die Studenten vorgeschlagenen Fragen und Antworten werden durch die Betreuer der Veranstaltung überprüft werden und freigegeben. Dadurch wird die aktive Mitwirkung der Studierenden an der Veranstaltung Softwaretechnik gefördert und Elemente eines Flipped Classroom eingeführt.

Teil des Quiztools ist damit ein kontinuierlich wachsender Pool von Quizfragen zu den Architekturf Entwurfsmethoden. Fragen können mittels Rollen-/Rechtekonzept nur für Betreuer sichtbar gemacht werden. Das erlaubt es, auch Fragenpools für mündliche und schriftliche Prüfungen zu verwalten, die die Studierenden natürlich nicht vorher zu Gesicht bekommen sollen. Nach Verwendung in der Prüfung gehen sie dann in den allgemein zugänglichen Pool über.

4.2 Vernetzung mit dem Learning Outcome der Veranstaltung (Kapitel 2)

Die Vernetzung der Quiz- / Prüfungsfragen mit dem Learning Outcome liegt an der Stelle nahe. Fragen lassen sich direkt den Teilfähigkeiten eines Learning Outcome zuordnen (siehe auch Abbildung 2 auf Seite 4, unten rechts). Sofern die entsprechenden passende Auswertungssichten von der Software zur Verfügung gestellt werden, wird es mit ArchiLab unmittelbar transparent, bei welchen Themengebieten die Studierenden noch Verständnisprobleme haben.

Für das Online-Quiztool ergibt – anders als beim Management von Learning Outcomes und der Fallstudienverwaltung – ein semantisches Wiki als Plattform keine Vorteile, da das Datenmodell hier eher einfach ist und sich selten ändert. Eine klassische Webarchitektur ist an dieser Stelle

der einfachere und bessere Weg. Dies bedeutet, dass die Integration mit dem Management der Learning Outcome etwas aufwändiger ist als bei der Fallstudienverwaltung. Im Arbeitsplan ist die Integration daher auch als explizites Teilprojekt vorgesehen.

5 Langfristige Planung für ArchiLab

5.1 Verstetigung der Lehrinnovation

Das ArchiLab-Konzept ist auf eine Laufzeit von mehreren Jahren ausgelegt. Die grundsätzliche Idee hierbei ist, die starke Projektorientierung der TH Köln als Fachhochschule zu nutzen. Die Studierenden im Informatik-Bachelor und -Master müssen verschiedene Pflichtpraktika absolvieren, in denen konkrete Programmierarbeiten zu absolvieren sind.

Es ist vorgesehen, einen großen Teil der Spezifikation und Implementierung von ArchiLab als Studienleistung in studentischen Projektteams oder als Teil von Abschlussarbeiten durchzuführen (siehe hierzu auch den Arbeitsplan). Dadurch ist sichergestellt, dass das Projekt durch das Team der Fachgruppe Systemgestaltung am Institut für Informatik der TH Köln und durch meine eigene Person nachhaltig vorangetrieben wird.

Eine Projektförderung ist allerdings nötig für die Produktifizierung der Software – und damit auch erst für den praktischen Einsatz in der Lehre. Robustheit, Tests, Dokumentation und Wartung bleiben bei reinen Studentenprojekten erfahrungsgemäß mangelhaft. Eine Betreuung durch fest angestellte Kräfte ist nötig, um ArchiLab auch tatsächlich effektiv im Lehralltag zu verankern.

Bei diesem Lehreinsatz ist es auch hilfreich, dass ich als Lehrender die Randbedingungen der Lehre so gestalten kann, dass ein schlüssiges Gesamtkonzept entsteht. So wurden beispielsweise beim jetzt anstehenden Umzug der Fachgruppe Systemgestaltung in ein anderes Gebäude der TH Köln die Projekträume mit beschriftbaren und abwaschbaren Wänden ausgestattet wurden. Gemeinschaftliche Modellierungs-Workshops (wie in Abbildung 3 auf Seite 5) werden dadurch einfacher: Die Studierenden können einfach komplett alle Wände mit Whiteboardstiften beschriften.

5.2 Übertragung auf andere Fächer und Disziplinen

In Kapitel 1.4 (Seite 3) wurde schon dargestellt, welche Synergien innerhalb der Informatik-Studiengänge der TH Köln möglich sind.

Darüber hinaus ist – vorausgesetzt das Konzept wird erfolgreich umgesetzt und produktifiziert – ein Einsatz von ArchiLab auch in der Informatik-Ausbildung außerhalb der Hochschule sinnvoll. Es ist geplant, ArchiLab als Open Source zu veröffentlichen.

Eine Übertragung auf andere komplexe Fachdisziplinen (z.B. Ingenieurwissenschaften) scheint ebenfalls grundsätzlich möglich. Durch die Orientierung am „*Decoding the Disciplines*“-Ansatz trägt das Konzept keine inhärent Informatik-spezifischen Züge. Ein Einsatz von ArchiLab ist grundsätzlich auch in der Lehre von komplexen Ingenieursprojekten sinnvoll und möglich, wenn das Datenmodell entsprechend angepasst wird.

5.3 Erfolgs- und Risikobewertung in der Erprobung

Durch die Bindung an das Learning Outcome ist Feedback und Erfolgskontrolle ein wesentliches Element von ArchiLab, wie in den obigen Kapiteln dargestellt. Darüber hinaus werden wir – wie auch schon jetzt – Evaluationsbögen und andere Feedbackformen wie etwa den Teaching Analysis Poll (TAP) [5] zurückgreifen, um Erfolg und Risiken des Einsatzes von ArchiLab in der Lehre zu bewerten.

6 Referenzen

- [1] Anderson, Lorin W. und Krathwohl, David R. (2001). A Taxonomy for Learning, Teaching and Assessing: a Revision of Bloom's Taxonomy. New York. Longman
- [2] Beutner, J. et al. (2013). PINGO in der Lehre - didaktische Handreichung zu Einsatzmöglichkeiten. https://groups.uni-paderborn.de/wininfo2/pingo/misc/PINGO_Didaktische%20Handreichung%20Einsatzm%C3%B6glichkeiten_final.pdf, abgerufen 18.07.2016
- [3] Biggs, John und Tang, Catherine (2007): Teaching for quality learning at university. What the student does, Maidenhead: McGraw-Hill.
- [4] Middendorf, Joan, und David Pace (2004). "Decoding the Disciplines: A Model for Helping Students Learn Disciplinary Ways of Thinking." New Directions for Teaching and Learning 2004, no. 98 (2004): 1–12
- [5] Rüdell, Michael et al. (2016): Teaching Analysis Poll. https://www.th-koeln.de/mam/downloads/deutsch/hochschule/profil/lehre/planung_und_durchfu_hrung_tap.pdf, abgerufen 12.07.2016
- [6] Schaffert, S. et al. (2007). Semantic Wiki. In: Informatik Spektrum Bd.30, Nr.6 (Dezember 2007). ISSN 0170-6012, S. 434–439.
- [7] Sebis-Lehrstuhl an der TU München (2016). Social Information Modelling Platform for Collaborative, Evolutionary Data and Process Management (SocioCortex). www.socio-cortex.com, abgerufen 16.07.2016