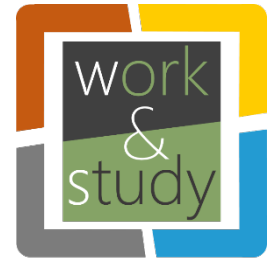


VERBUNDPROJEKT  
„work&study“  
Offene Hochschulen Rhein-Saar



## Modul "Software Engineering"

Erstellung des Moduls "Software Engineering" und erste Probendurchläufe in der Wirtschaftsinformatik

Ergebnisbericht WiSe 2015/16 und 2016/17



Das diesem Bericht zugrundeliegende Vorhaben wird mit Mitteln des Bundesministeriums für Bildung und Forschung unter den Förderkennzeichen 16OH21054, 16OH21055, 16OH21056 & 16OH21057 gefördert.

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei folgendem Autor:

Prof. Dr. Manfred Kaul

## Inhaltsverzeichnis

1	Aufgabenstellung und Herleitung des Ansatzes .....	3
2	Lösungsansatz .....	3
3	Umsetzung.....	3
4	PBL-Studie .....	11
5	Erfahrungen.....	11
6	Reflexion und Planung der Überarbeitung .....	12
7	Literatur.....	13

## Abkürzungsverzeichnis

ACM	Association for Computing Machinery.....	5
<i>ccm</i>	<i>Client-side Component Model</i> .....	6, 12
DigiBib	digitale Bibliothek der Hochschule Bonn-Rhein-Sieg .....	5, 11
H-BRS	Hochschule Bonn-Rhein-Sieg .....	4
IB	Internet Business .....	13
IEEE	Institute of Electrical and Electronics Engineers .....	5
NTS	Nicht-traditionell Studierender .....	4, 6
PBL	Problem Based Learning .....	2, 12
SE	Software Engineering .....	2, 4, 5, 8, 13
SE1	Software Engineering I.....	2, 7
VCRP	Virtuellen Campus Rheinland-Pfalz.....	4
w&s	work&study .....	4
WE	Web Engineering .....	2, 4, 13
WI 2	Wirtschaftsinformatik II.....	13

## Abbildungsverzeichnis

Abb. 1	Drei Studierenden-Sichten der SE1-Course App auf Smartphone .....	6
Abb. 2	Eingebettete Übung mit UML-Modellen .....	7
Abb. 3	PlantUML, siehe (PlantUML 2017) .....	8
Abb. 4	Dozenten-Dashboard: 2 Screenshots .....	9
Abb. 5	ILIAS-Ansicht der 14. Lehrinheit zu Software Engineering .....	10
Abb. 6	Neuer Aufbau der Vertiefung der Hochschule Bonn-Rhein-Sieg mit vertauschter Reihenfolge von SE und WE.....	12

## Tabellenverzeichnis

Tab. 1	Unterschiede SE-Vorlesungen im WiSe 2015/16 und im WiSe 2016/17.....	4
--------	--	---

# 1 Aufgabenstellung und Herleitung des Ansatzes

Im Verbund-Studiengang des Verbund-Projektes „work&study“ (w&s) sollte vertragsgemäß für die Vertiefung "Wirtschaftsinformatik" der Hochschule Bonn-Rhein-Sieg (H-BRS) ein Modul "Software Engineering" (SE) im Umfang von 6 Credits erstellt und erprobt werden. Es dient als Grundlage für das darauf aufsetzende Modul "Web Engineering" (WE), das konsekutiv im Anschluss daran gelehrt werden sollte. Dabei sollte gemäß des Bedarfs der Zielgruppe "Nicht-traditionell Studierender" (NTS) der Online-Anteil wesentlich erhöht werden. Als grundlegendes Modell für den gesamten Studiengang hat der Verbund "Blended Learning" gewählt, also eine Mischung aus Online- und Präsenz-Anteilen. Die Art der Mischung lässt viel Spielraum übrig, so dass es sinnvoll erschien, frühzeitig mit Experimenten in dem gleichnamigen Modul des Autors in dem seit Jahren laufenden Studiengang "Wirtschaftsinformatik" der H-BRS zu beginnen. Das Verbundprojekt „work&study“ hatte sich auf das HTML5-Format für den Content und OpenOLAT des Virtuellen Campus Rheinland-Pfalz (VCRP) als Plattform verständigt. Für den Betrieb vor Ort stand jedoch die ILIAS-Plattform der H-BRS - genannt LEA - zur Verfügung, keine OLAT-Plattform. Als gemeinsamer Nenner konnte auf HTML5-Basis jedoch portabler Content entwickelt werden, der sowohl vor Ort als auch im Verbund zum Einsatz gebracht werden konnte. Der zusätzliche Vorteil der Fokussierung auf HTML5-Content-Generierung lag außerdem in der Übertragbarkeit auf weitere Plattformen, insbesondere moderne mobile Plattformen wie Smartphones und Tablets. Dies eröffnet einen leichteren Übergang zu neuen Formaten wie Apps, Web Apps und Course Apps, die sich als neuer Hoffnungsträger am E-Learning-Himmel zeigen, siehe die strategische Studie des Horizon Projektes, Course Apps (2015). Die Aufgabenstellung war also die Produktion von HTML5-Content für SE, der universell einbettbar sein sollte für OLAT, ILIAS und Course Apps.

## 2 Lösungsansatz

Da die Hochschule Bonn-Rhein-Sieg bereits seit vielen Jahren einen Studiengang "Wirtschaftsinformatik" betreibt und der Autor darin die Vorlesung "Software Engineering" liest, lag es nahe, aus dieser Präsenz-Vorlesung Online-Material zu generieren, das die Grundlage für ein gleichnamiges Blended-Learning-Modul bilden könnte. Außerdem konnten die vorhandenen Studierenden der Hochschule Bonn-Rhein-Sieg als Probanden die ersten Durchläufe testen und dazu Feedback geben. Zudem sieht der Projektplan das Thema "Übertragbarkeit" vor, also die Untersuchung der Frage, ob man das Online-Material auch in anderen Studiengängen mit anderen Studierenden einsetzen könne. So konnten mehrere Ziele gleichzeitig erreicht werden.

## 3 Umsetzung

Das um Online-Material angereicherte Modul "Software Engineering" wurde im Blended Learning-Modus bisher zweimal gelehrt, und zwar im WiSe 2015/16 und im WiSe 2016/17. Es wurden zwei verschiedene didaktische und zwei verschiedenen technische Formate bei gleichem Lehrinhalt erprobt. Gemeinsam war beiden Vorlesungen die wochenweise Einteilung des Lehrstoffes in 14 bzw. 15 Lehreinheiten und der umfangreiche Einsatz von Online-Material in Form von wiss. Texten zur eigenständigen Lektüre, Folien und Probeklausur als PDF-Datei, interaktiver Klausurtrainer, Videos, Quiz mit sofortigem Feedback, Schock-Memory (interaktive Frage-Antwort-Datenbank, aus der Quizfragen generiert werden), Übungen mit Upload-Funktion sowohl formular-basiert als auch Datei-Upload, Ansicht der Lösungen aller Mitstudierenden nach der Deadline und universeller Feedback-Funktion. Auf einzelne Features wird weiter unten eingegangen.

In beiden Vorlesungen wurde außerdem die Realtime-Gruppenbildung eingesetzt, (vgl. Kless 2017), allerdings zu unterschiedlichen Zwecken: Im WiSe 2015/16 wurden damit die semesterbegleitenden Projektteams gebildet. Im WiSe 2016/17 ging es lediglich um kurzfristige Ad Hoc Teams für die gemeinsame Bearbeitung einer einzelnen Übung in Lehrinheit 14.

Schock-Memory ist ein beliebtes didaktisches Format, bei dem Studierende zu einem vorgegebenen Themengebiet oder zu einer vorgegebenen Lektüre eine Frage formulieren müssen, zu der sie auch die Antwort wissen. Beides, sowohl Frage als auch Antwort müssen in das System formular-basiert eingegeben werden. Daraus entsteht eine Datenbank von korrekten Frage-Antwort-Paaren. In einem Master-Projekt, das vom Autor betreut wurde, (vgl. Malcher 2015), konnte diese Idee dahingehend erweitert werden, dass nicht nur die korrekte Antwort eingegeben wird, sondern eine größere Anzahl von Antworten, die jeweils als wahr oder falsch zu markieren sind. Daraus entsteht eine umfangreichere Frage-Antwort-Datenbank, aus der Quiz generiert werden können, die dann eine Woche später zur Wiederholung oder am Ende des Semesters zur Klausurvorbereitung eingesetzt werden können. Durch ein studentisches Rating können die beliebtesten Frage-Antwort-Paare ermittelt werden, so dass eine Art "Wunsch-Klausur" daraus generierbar ist.

Für die wiss. Lektüre wurde auf die hauseigene digitale Bibliothek der Hochschule Bonn-Rhein-Sieg (DigiBib) zugegriffen. Die Bibliothek der Hochschule Bonn-Rhein-Sieg hat für viel Geld prominente Anbieter wissenschaftlicher Literatur unter Vertrag wie ACM, IEEE und Springer-Verlag. Dieses Angebot wird im Präsenzstudium viel zu selten genutzt, wenn es von den Dozenten nicht in die Übungen geeignet als Pflichtbestandteil eingebaut wird. Daher wurden Übungen, Quiz und Schock-Memory gezielt mit der Lektüre weniger Kapitel ausgewählter Bücher des SE verknüpft und Quizfragen konnten erst beantwortet werden, wenn die Lektüre gelesen war.

Unterschiedlich gestaltet waren die beiden SE-Vorlesungen im WiSe 2015/16 und im WiSe 2016/17 in folgenden Punkten (siehe Tabelle 1):

	WiSe 2015/16	WiSe 2016/17
semesterbegleitendes Projekt	Ja.	Nein.
Course App	Ja.	Nein.
LEA-Nutzung	weniger	stärker

Tab. 1 | Unterschiede SE-Vorlesungen im WiSe 2015/16 und im WiSe 2016/17

Im WiSe 2015/16 wurde ein semesterbegleitendes Projekt in Teamarbeit parallel zu den Übungen und der Vorlesung von den Studierenden verpflichtend verlangt. Es stellte sich jedoch heraus, dass beides zusammen zu einer zeitlichen Überlastung führte. Daher wurde das Projekt aus dem SE-Konzept gestrichen, zumal es im Studiengang Wirtschaftsinformatik im Folgesemester explizit angeboten wird. Für den Erstkontakt mit SE-Themen ist ein behutsamer, kleinschrittiger Übungsbetrieb angemessener als direkt mit anspruchsvoller Projektarbeit zu starten.

Um die volle Portierbarkeit von HTML5-Content zu testen, wurde im WiSe 2015/16 ganz ohne Plattform getestet, d.h. der HTML5-Content direkt in eine einfache HTML-Seite eingebettet. Um die Interaktivität voll nutzen zu können, wurde bereits eine Course App in einfachster Form programmiert, siehe Course Apps (2015). Im WiSe 2016/17 stand jedoch die Einbettbarkeit des HTML5-Contents in LEA im Vordergrund. Daher wurde auf eine Web App verzichtet, obwohl diese recht leicht aus dem gegebenen HTML5-Code für die einzelnen Lehrinheiten zu bewerkstelligen wäre.

Die LEA-Nutzung beschränkte sich im WiSe 2015/16 auf Einbettung der Web App in einer LEA-Seite. Dort war die Web App voll funktionsfähig, konnte jedoch auch ohne LEA aufgerufen werden. Die meisten Studierenden bevorzugten die Web App ohne LEA, weil diese wesentlich schneller startete und den gleichen Funktionsumfang bot. Im WiSe 2016/17 lag der Fokus auf der LEA-Einbettung. Dazu wurden eigene HTML-Custom Tags definiert, mit denen interaktiver HTML5-Content direkt in LEA zur Verfügung deklarativ gestellt werden konnte. Die Konfiguration ist dadurch mittels Tag-Attribute deklarativ möglich und erfordert keinerlei Programmierkenntnisse.

Insgesamt wurde der HTML5-Content so produziert, dass ein Übergang zu Course Apps problemlos möglich wird. In dem strategischen Bericht des Horizon Projects, Course Apps (2015), wird festgestellt, dass die üblichen E-Learning-Plattformen wie Moodle, ILIAS und OLAT dem stetig steigenden Bedarf an (A) Mobilität, (B) Interaktivität, (C) ansprechendem Design und (D) Learning Analytics nicht mehr gerecht werden können. Daraus wurde der Bedarf abgeleitet, universell einbettbaren Content zu produzieren.

Nach dem strategischen Bericht des Horizon Projects, „Course Apps“ (2015), sind diese Ziele mit speziellen für Kurse generierten Course Apps wesentlich besser erreichbar. Durch sie werden vier aktuelle Trends zusammengeführt:

- a. Der Trend von gedrucktem zu digitalem Lehrmaterial,
- b. der wachsende Bedarf an Mobilität der Lernplattform und der Trend zum Smartphone als universellem Lebensbegleiter
- c. Innovationen im WWW und
- d. die Konvergenz von Web und Learning Analytics.

Um den leichten Übergang zu Course Apps zu fördern, wurde das Client-side Component Model (*ccm*) eingesetzt. *ccm* ist das Ergebnis der Masterarbeit von André Kless, der vom Autor bei seiner Master Thesis betreut wurde (vgl. Kless 2015). Mit dieser Technologie wird die Einbettbarkeit des produzierten HTML5-Contents universell möglich. Als Dozent und Autor produziert man nur einmal und kann seine Kurse vielfach nutzen in allen E-Learning-Plattformen wie ILIAS, Moodle, OLAT und in allen Varianten von Apps wie Web Apps, Native Apps oder Hybrid Apps, ohne auf Interaktivität und Kollaboration verzichten zu müssen. Mittels Responsive Design erstreckt sich der Einsatz über alle Hardware-Plattformen, angefangen beim Smartphone über Tablet und Notebook bis hin zum Desktop. Gerade für die angestrebte Zielgruppe der NTS sind diese Anforderungen unabdingbar (vgl. Kless 2017).

E-Books sind neben E-Learning-Plattformen ein beliebtes Medium. Allerdings werden E-Books häufig nur als digitaler Abdruck der Print-Version eines Buches produziert und verfehlen die genannten vier Ziele (A) bis (D) ebenso. Mit *ccm* wird auch diese Plattform bedient, so dass man seinen E-Book-Content um Interaktivität und Kollaboration anreichern kann: Quiz, Trainings, Übungen, Simulationen und Social Reading werden dadurch nahtlos ermöglicht.

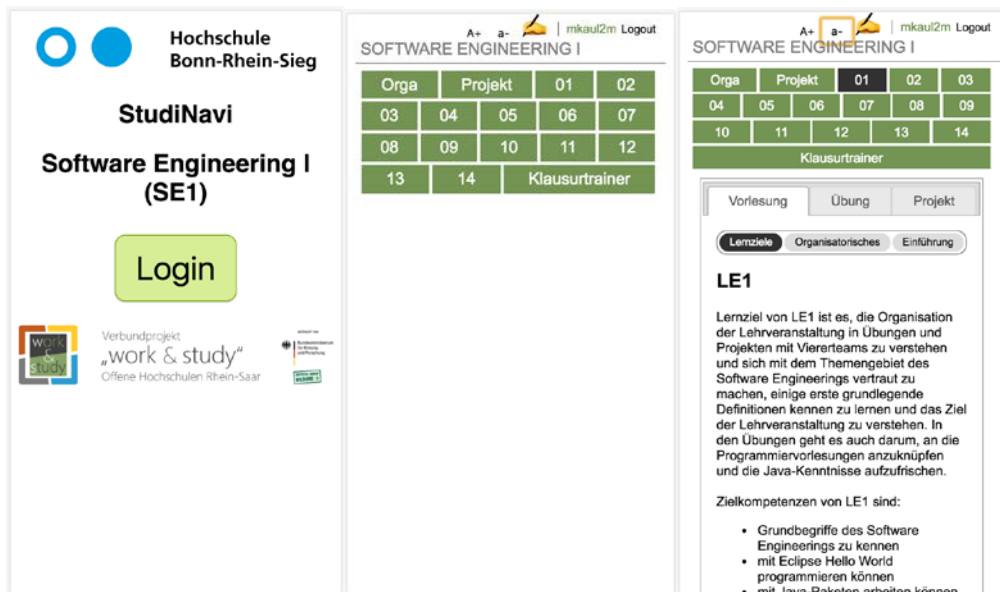


Abb. 1 | Drei Studierenden-Sichten der SE1-Course App auf Smartphone

In Abb. 1 sieht man die Sicht der Studierenden auf die SE1-Course App auf einem iPhone 5. Darin stehen in der Kopfzeile: A+ zum Vergrößern des Fonts, a- zum Verkleinern. Die schreibende Hand dient dem Feedback, das kontext-sensitiv jederzeit möglich ist. Das dunkelgrüne Menü zeigt die Aufteilung des Lehrstoffes in 14 Wochen. Nach Auswahl einer Woche erscheinen darunter geschachtelte Menüs zur Vorlesung, zu den Übungen und zum Projekt. Eine Auswahl im geschachtelten Menü führt schließlich zur Aufgabenstellung, wie in Abb. 2 zu sehen ist.



Orga	Projekt	01	02	03	04	05	06	07	08
09	10	11	12	13	14	Klausurtrainer			

Vorlesung
Übung
Projekt

Lernziele
4.5 Abbott-Methode
4.6 Analysemodell
4.7 Robustheitsdiagramm

Feedback zur LE

## 04.06: Analysemodell

Deadline war 1.11.2015 (1 Punkt)

Erstellen Sie in Ihrem Team für Ihr Zielsystem einer Parkhaus-Software, das Ihr Team in diesem Semester im Rahmen der Lehrveranstaltung SE1 planen und realisieren will, ein Klassendiagramm als Analysemodell auf der Grundlage der Ergebnisse der Abbott-Methode aus der letzten Aufgabe.

- Laden Sie das Analysemodell als Klassendiagramm als PNG-, GIF- oder JPG-Datei hoch! Wählen Sie als Name für die Datei 'project\_04\_06\_mkaul2m'.  
*Upload (max. 4 MB, max. 60 sec)*
- Beschreiben Sie den Inhalt Ihres Klassen-Diagramms

Mein Klassendiagramm-Diagramm umfasst folgende Klassen: ...

- PlantUML-Spezifikation als Alternative zum Hochladen:

```
@startuml
class MyClass {
    id: int
+id(): int
+add(a:int, b:int):int
}
@enduml
```

```
classDiagram
    class MyClass {
        id: int
        +id(): int
        +add(a:int, b:int):int
    }
```

Redraw UML
Help UML

[PlantUML](#) is a component that allows to quickly write :

- [Sequence diagram.](#)
- [Usecase diagram.](#)

Abb. 2 | Eingebettete Übung mit UML-Modellen

Ein wichtiger Lehrgegenstand von SE ist die Modellierung, vorzugsweise mit der Unified Modeling Language (UML). Die Einarbeitung in ein UML-Modellierungswerkzeug stellt jedoch eine zusätzliche Zeitbelastung dar. Daher wurde in die SE-Course App PlantUML integriert (vgl. PlantUML 2017): "PlantUML ist ein Open Source Projekt, welches das Erstellen von UML-Diagrammen ermöglicht. Es werden die folgenden Typen von UML-Diagrammen unterstützt:

- | Sequenzdiagramm,
- | Anwendungsfalldiagramm,
- | Klassendiagramm,
- | Aktivitätsdiagramm,
- | Komponentendiagramm,
- | Zustandsdiagramm,
- | Deployment diagram,
- | Objektdiagramm
- | wireframe graphical interface

Diagramme werden in einer einfachen und intuitiven Sprache in textueller Notation beschrieben.

Mit PlantUML gibt man eine textuelle Spezifikation in ein Textfeld ein und das entsprechende Diagramm wird sofort gerendert, siehe Abb. 3. Die Software PlantUML wird frei als Cloud Service betrieben, so dass keinerlei Installation erfolgen muss. Im Hintergrund wird der eingegebene Text mittels Ajax-Request an den PlantUML-Server gesendet und in weniger als einer halben Sekunde das gerenderte UML-Diagramm als Bild zurück geliefert und im Browser dargestellt.

- PlantUML-Spezifikation als Alternative zum Hochladen:

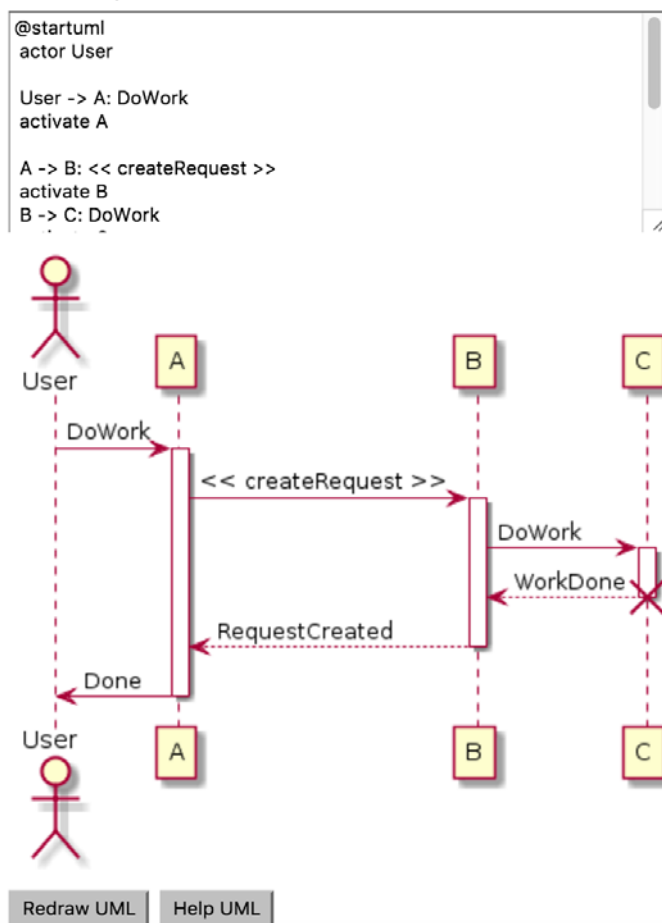


Abb. 3 | PlantUML, siehe (PlantUML 2017)



Für die Sicht der Dozierenden wird ein Dashboard als Web App angeboten, siehe Abb. 4. Darin kann sich der Dozent einen Überblick über den derzeitigen Stand der Übungsleistungen der Studierenden verschaffen.

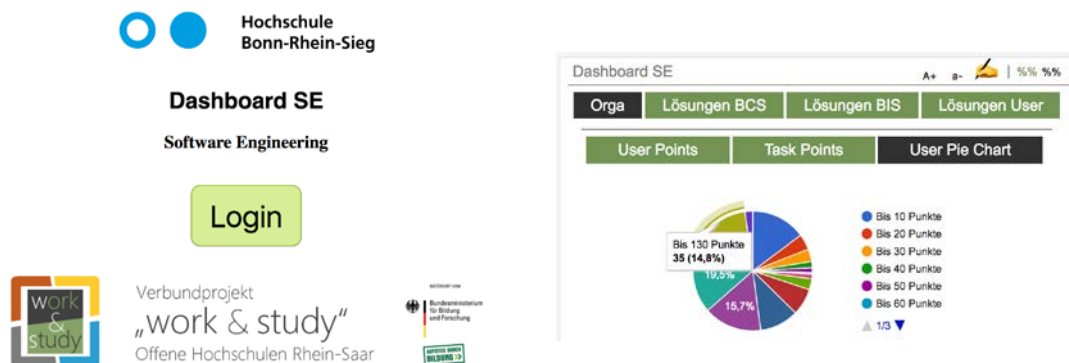


Abb. 4 | Dozenten-Dashboard: 2 Screenshots

Im WiSe 2016/17 wurde die Integration des interaktiven Lehrangebotes in LEA (die seit vielen Jahren bestehende zentrale E-Learning-Plattform der Hochschule Bonn-Rhein-Sieg) weiter vorangetrieben. Folgende von LEA angebotene Funktionalität wurde unverändert genutzt:

- | ILIAS-Mail (E-Mail-freies Benachrichtigungssystem der Plattform). (Da Studenten immer schlechter über Standard-E-Mail (Outlook, SMTP-Mail) zu erreichen sind, gewinnt dieses Medium an Bedeutung.)
- | Forum
  - o Nachrichtenforum für den Dozenten zur Benachrichtigung aller Studierender (6 Beiträge und viele Hundert Besuche)
  - o Kommunikationsforum für studentischen Austausch (56 Beiträge und viele Hundert Besuche)
- | ILIAS-Chat für Online-Betreuung
- | Gruppeneinteilung
- | Links auf Lehrmaterialien anderer Kurse (z.B. Programmierung)
- | Wiki

Jede der wöchentlichen Lehreinheiten wurde als eigenständige HTML5-Seiten umgesetzt. Mittels Custom Tags wurde die Interaktivität dieser Seiten erweitert um

- | Videos
- | Quizzes
- | Formulare für Text- und Code-Eingabe studentischer Lösungen mit Punktevergabe
- | Upload von Dateien studentischer Lösungen mit Punktevergabe
- | Anzeige der Liste aller studentischer Lösungen nach der Deadline



Abb. 5 | ILIAS-Ansicht der 14. Lehreinheit zu Software Engineering

Alle 15 Ordner in ILIAS wurden mit HTML-Inhalten, Videos und PDF-Anhängen gefüllt. Jeder Inhalt war nach dem gleichen Schema strukturiert:

- e. Lernziele
- f. Vorbedingungen
- g. Vorlesung
- h. Übung
- i. Literatur

Zusätzlich wurden zu jeder Woche wiss. Lektüre aus der DigiBib, ein Video-Mitschnitt der Vorlesung (die in einem klassischen Hörsaal gehalten wurde) sowie die Präsentationsfolien als PDF-Datei mitgeliefert. Mehr als die Hälfte der Informatik-Studierenden war von diesem umfangreichen Angebot begeistert und hat den Kurs mit der Bestnote evaluiert. Als größtes Problem stellte sich hingegen die Stofffülle und der Zeitbedarf für die Veranstaltung heraus.

## 4 PBL-Studie

Da der Kurs "Software Engineering" für beide Bachelor-Studiengänge des Fachbereichs Informatik gehalten wurde und sich über 450 Studierende insgesamt angemeldet hatten, wurde die Gelegenheit für eine Vergleichsstudie zum Thema "Problem Based Learning" (PBL) genutzt. Dazu wurde die 14. Lerneinheit mit dem Thema "Patterns" in der verschiedenen Studiengängen unterschiedlich gelehrt: Die Informatik-Studierenden sollten die Aufgaben in Präsenz-Gruppenarbeit in der Übung vor Ort lösen, während die Wirtschaftsinformatik-Studierenden die gleichen Aufgaben online bewältigen sollten (in Online Teams mittels der üblichen Online-Werkzeuge wie TeamViewer, Skype, WhatsApp, Facebook-Gruppen). Vorher und nachher wurden ausführliche Befragungen in beiden Studierendengruppen für Vergleichszwecke durchgeführt. Über 130 Studierende haben sich an den Befragungen beteiligt. Die Ergebnisse der Befragungen werden gesondert publiziert.

## 5 Erfahrungen

E-Learning-Plattformen bieten heute viele Möglichkeiten, Hochschul-Lehre durch Online-Zugaben interessant zu ergänzen. Davon wurde in LEA ausgiebig Gebrauch gemacht. Vor allem das Kommunikationsforum und Plattform-interne Benachrichtigungssystem haben sich bewährt. Gleichzeitig konnte der Wunsch, über die Möglichkeiten der E-Learning-Plattform hinaus zu gehen und eigene didaktische Anforderungen und Interaktions- sowie Kollaborationsszenarien umzusetzen, durch HTML-Custom Tags mittels *ccm* erfüllt werden. Dazu zählen sowohl Quizzes, formularbasierte Lösungsabgabe von Quellcode mit Punkten und Deadline als auch die Anzeige der Lösungen aller Mitstudierender. Gerade im Software Engineering gibt es oft keine "Musterlösung", sondern der didaktische Wert liegt in der Anerkennung vieler Sichtweisen und Aspekte, dem Erkennen von Zielkonflikten, dem Vergleich der unterschiedlichen Lösungen, der Abwägung und Balancierung und der Diskussion darüber. Auch studentische Lösungen mit Fehlern haben sich für die Diskussion in den Übungen als sehr wertvoll herausgestellt. Man lernt nicht nur von Musterlösungen, sondern auch von Fehlersuche und Fehler-Diskussionen. Dieses Präsenz-Format lässt sich durchaus in ein Online-Format übersetzen, z.B., indem man eine fehlerhafte Lösung laut denkend im Screencast korrigiert und das so aufgezeichnete Video allen zur Verfügung stellt. Dies ist auch bereits im WiSe 2016/17 durchgeführt worden.

Auch wenn ILIAS formularbasierte Lösungsabgabe bereits anbietet, so ist es doch oft eine Frage der Integration und Usability, ob man von dieser Möglichkeit Gebrauch macht. Im vorliegenden Fall wurde oft eine eigene Lösung eingesetzt, weil sie weniger Klicks erfordert und in die Lerneinheit einfacher zu integrieren war. Außerdem konnten die eingereichten Lösungen so einem gesonderten Post-Processing (Punkte-Vergabe, Plagiatssuche, Leereingaben-Detektion, usw.) einfacher zugeführt werden, was bei Daten, die in ILIAS-Datenbanken liegen, nur nach manueller Kopie und Extraktion möglich ist: ILIAS-Datenbanken sind geschützt und für den Dozenten nicht zugreifbar.

Videos wurden von Studierenden intensiv genutzt. Viele Studierende haben sich lieber die Videos zu Hause angeschaut, statt in die Vorlesung zu kommen. Studierende, für die das Sprechertempo zu langsam war, konnten das Video beschleunigt abspielen. Umgekehrt war auch eine Verlangsamung möglich. So kann auch ein Beitrag zur Bewältigung der Heterogenität der Sprachkompetenzen der Studierendenschaft geleistet werden. Bei 450 Anmeldungen wurde nebenbei auch das Problem der überfüllten Hörsäle gelöst.

Will man Video-Mitschnitte von Vorlesungen in Hörsälen einsetzen, so muss auf gute Qualität geachtet werden. Mindestbedingungen sind heute HD-Kameras und HD-Beamer. Bei einem schlechten Beamer sind die Folien kaum zu erkennen. Noch wichtiger als Video ist guter Ton. Dazu ist unbedingt ein erstklassiges Headset einzusetzen. Da Video-Technik sehr schnell veraltet, sind immer wieder Reinvestitionen erforderlich.

## 6 Reflexion und Planung der Überarbeitung

Im Unterschied zum Präsenz-Studiengang „Wirtschaftsinformatik“ an der Hochschule Bonn-Rhein-Sieg kann man im Verbundstudiengang im Modul "Software Engineering" keine Programmierkenntnisse voraussetzen. Außerdem sind alle essentiellen Themen der Informatik wie formale Sprachen, Programmiersprachen und Codierungsthemen für Nicht-Informatiker ungewohnt. Auf der anderen Seite fällt es Informatik-Laien häufig sehr leicht, sich in das WWW mit seinen formalen Sprachen HTML, CSS und JavaScript einzuarbeiten. Viele Nicht-Informatiker sind als Web-Entwickler tätig ohne ein Informatik-Studium, viele als Autodidakten. Die Einarbeitung in den Umgang mit formalen Sprachen und formalen Systemen fällt also über die formalen Sprachen des WWW leichter als über die klassischen Programmiersprachen.

Daher wurde für das Verbundprojekt „w&s“ entschieden, die Reihenfolge von Software Engineering (SE) und Web Engineering (WE) in der Vertiefung „Wirtschaftsinformatik“ umzukehren: Zunächst sollen sich Studierende in die leichteren Themen des WWW einarbeiten, bevor sie zu den schwereren Themen des Software Engineerings vorstoßen. Zusammen mit Wirtschaftsinformatik II (WI 2) und Internet Business (IB) hat die Vertiefung Wirtschaftsinformatik das Ziel, beide Seiten, Wirtschaft und Informatik zu beleuchten und die Kompetenz des Wirtschaft-Technik-Ko-Designs exemplarisch anhand eines App-Geschäftsmodells und eines App-Prototypen zu vermitteln (siehe Abb. 6). Die Einführung in den Umgang mit formalen Systeme und formalen Sprachen soll dann exemplarisch anhand von WWW, HTML und CSS erfolgen. Die Einführung in Programmierung und Codierungsthemen soll exemplarisch anhand von JavaScript erfolgen. Das wird Inhalt von WE sein. In der Folge ist SE darauf abzustimmen und alle Codierungsbeispiele aus dem Bereich des WWW zu nehmen.

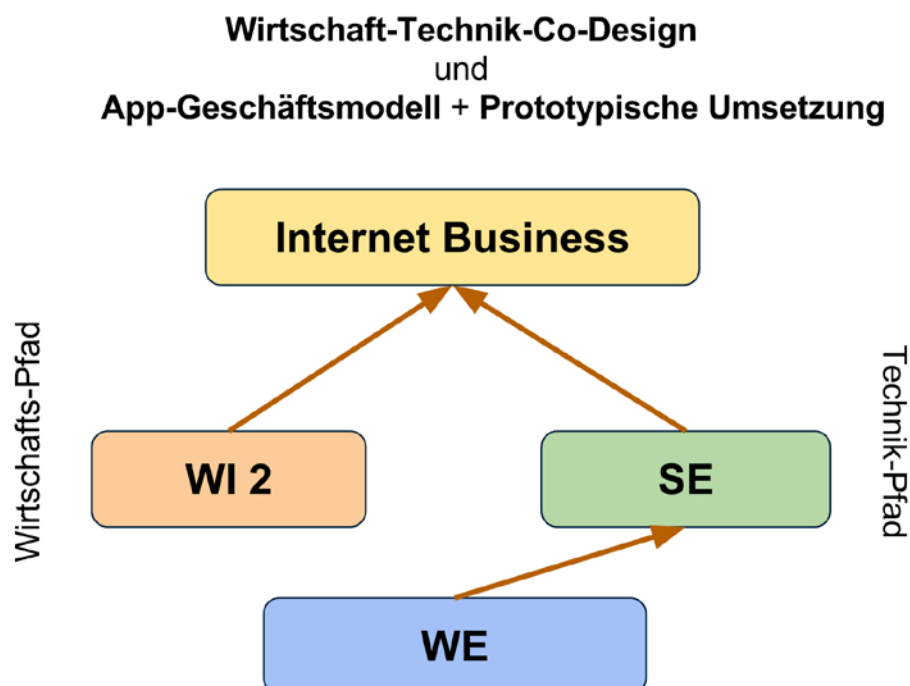


Abb. 6 | Neuer Aufbau der Vertiefung der Hochschule Bonn-Rhein-Sieg mit vertauschter Reihenfolge von SE und WE

## 7 Literatur

Course Apps (2015), A NMC Horizon Project Strategic Brief, Volume 2.4, October 2015, Horizon Project, <http://horizon.nmc.org>. abgerufen am 20.1.2017

Kless, André (2015): Eingebettetes kollaboratives E-Learning, Master-Thesis, Hochschule Bonn-Rhein-Sieg.

Kless, André (2017): Lösungen zielgruppengerechter Lehre, Zusammenfassung der bisherigen Forschungsaktivitäten des Verbundpartners Hochschule Bonn-Rhein-Sieg, BMBF-Verbundprojekt Work and Study.

Malcher, Roman (2015): Projektbericht im Studiengang Master of Science in Computer Science, Konzeption und Implementierung einer plattformunabhängigen und einbettbaren HTML5 E-Learning Webanwendung, von Roman Malcher, Erstbetreuer: Prof. Dr. Manfred Kaul, Zweitbetreuer: Prof. Dr. Thorsen Bonne, Eingereicht am: 18. September 2015, Hochschule Bonn-Rhein-Sieg.

PlantUML (2017): <http://plantuml.com>, abgerufen am 20.1.2017.