

Weiterentwicklung des Teilzeitstudiengangs Informatik an der Westsächsischen Hochschule Zwickau



Prorektorat
für Bildung

September 2017

Förderhinweis

Das dieser Dokumentation zugrundeliegende Vorhaben „Offene Hochschule Zwickau“ wird mit Mitteln des Bundesministeriums für Bildung und Forschung und des Europäischen Sozialfonds der Europäischen Union unter dem Förderkennzeichen 16OH12018 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autorinnen und Autoren.

Impressum

Herausgegeben durch: das Projekt „**Offene Hochschule Zwickau**“ an der Westsächsischen Hochschule Zwickau, vertreten durch den Prorektor für Bildung Prof. Dr. Gundolf Baier (Leiter des Projektes „Offene Hochschule Zwickau“)

Copyright: Vervielfältigung oder Nachdruck auch auszugsweise zur Veröffentlichung durch Dritte nur mit ausdrücklicher Zustimmung der Verfasser/-innen.

Datum: September 2017



Inhalt

1	Weiterentwicklung des Teilzeitstudiengangs „Informatik“	4
1.1	Ausgangssituation im Studiengang „Informatik“	4
1.2	Grundlagen der Programmierung I.....	6
	Rahmendaten des Moduls.....	6
	Konzeptionelle Überlegungen zur Umgestaltung des Moduls	6
	Herangehensweise	7
	Erprobung im Wintersemester 2015/2016 (Oktober bis Januar)	9
	Wissenschaftliche Begleitung.....	10
	Fazit.....	12
1.3	Grundlagen der Programmierung II.....	14
	Rahmendaten des Moduls.....	14
	Konzeptionelle Überlegungen	14
	Herangehensweise	15
	Erprobung im Sommersemester 2016 (April bis Juni)	15
	Wissenschaftliche Begleitung.....	16
	Fazit.....	19
1.4	Entwicklung verteilter Anwendungen.....	20
	Rahmendaten des Moduls.....	20
	Inhalt und Methodik	20
	Kursaufbau	21
	Verlauf der Erprobung.....	23
	Wissenschaftliche Begleitung:.....	23
	Fazit.....	47
1.5	Netzwerke	48
	Rahmendaten des Moduls.....	48
	Inhalt und Methodik	48
	Kursaufbau	49
	Wissenschaftliche Begleitung.....	56
	Fazit.....	63
2	Abbildungsverzeichnis.....	64

1 Weiterentwicklung des Teilzeitstudiengangs „Informatik“

1.1 Ausgangssituation im Studiengang „Informatik“

Innerhalb der ersten Förderphase des Projekts „Offene Hochschule Zwickau“ wurde für den Bachelor-Studiengang Informatik die Möglichkeit geschaffen, das Studium auch in Teilzeit zu absolvieren. Es wurde eine Teilzeitordnung konzipiert und 2014 von den Hochschulgremien verabschiedet. Die Studierenden haben nach dieser Ordnung die Möglichkeit, zwischen Vollzeit- und Teilzeitstudium zu wechseln, um das Studium an ihre Lebensplanung anzupassen. Zwei Semester in Teilzeit entsprechen dabei dem Studiumumfang eines Vollzeitstudiensemesters.

Die Möglichkeit zum Teilzeitstudium wird als Einstieg in die Flexibilisierung von Studiengängen verstanden und ist mittlerweile auch für andere Bachelor- und Masterstudiengänge der WHZ in den Studien- und Prüfungsordnungen festgeschrieben.

Der Bachelorstudiengang Informatik, der zum Abschluss Bachelor of Science (mit 210 ECTS) führt, kann in Vollzeit über 7 Semester und in Teilzeit über (bis zu) 14 Semester studiert werden. An das Studium kann ein konsekutiver Master of Science in Voll- (3 Semester) oder Teilzeit (bis zu 6 Semester) in Informatik an der WHZ angeschlossen werden.

Gemeinsam mit dem Fachbereich Informatik wurden vier Module des Bachelor-Studiengangs Informatik für die Umgestaltung auf online-basiert ausgewählt, die aber auch teilweise von Studierenden des Studiengangs Gesundheitsinformatik belegt werden müssen. Mit der Umgestaltung soll ein Beitrag zu mehr Flexibilität in der Studiengestaltung geleistet werden, unabhängig davon, ob das Studium in Voll- oder Teilzeit absolviert wird. Innerhalb des Projektverlaufs hat es Veränderungen bei der Auswahl der Module und der Erprobungszeiträume ergeben. Diese sind auf Veränderungen in der Studienorganisation und auf Neubesetzungen bzw. Nichtbesetzungen von Professuren zurückzuführen. Da gleich zu Projektbeginn für die Module zu „Grundlagen der Programmierung“ neue Lehrende bestimmt wurden, bestand die Möglichkeit diese Module neu zu entwickeln und schon früher in die Erprobung zu gehen. 2016 wurde in den Studienablaufplan des Fachs Informatik ein neues Modul aufgenommen. Das Modul „Entwicklung verteilter Anwendungen“ greift die neuen technologischen Entwicklungen innerhalb des Fachgebiets auf und wurde bisher in Ansätzen nur innerhalb des Masters gelehrt:

Laut Projektplanung		Erprobung	
Module	Erprobungszeitraum	Module	Erprobungszeitraum
Grundlagen der Programmierung I	Wintersemester 2016/17	Grundlagen der Programmierung I	Wintersemester 2015/16
Grundlagen der Programmierung II	Wintersemester 2016/17	Grundlagen der Programmierung II	Sommersemester 2016
Betriebssysteme	Sommersemester 2017	Entwicklung verteilter Anwendungen	Wintersemester 2016/17
Netzwerke	Sommersemester 2017	Netzwerke	Sommersemester 2017

Zu den Modulen werden im Folgenden die konzeptionellen Überlegungen und Ergebnisse der Erprobung und wissenschaftlichen Begleitung dargestellt.

1.2 Grundlagen der Programmierung I

Rahmendaten des Moduls

Das Modul „Grundlagen der Programmierung I“ ist für alle Studierenden der Informatik und Gesundheitsinformatik verpflichtend im ersten Semester zu besuchen.

Für das Modul ist ein Workload in Höhe von 180 Stunden angesetzt. Es setzt sich aus zwei Semesterwochenstunden (SWS) Vorlesung und 2 SWS Praktikum zusammen, die in Präsenz (60 Stunden) durchgeführt werden. 120 Stunden und damit ein sehr großer Anteil sind für das Selbststudium der Studierenden vorgesehen. Die Prüfungsleistung erfolgt in Form einer schriftlichen Prüfungsleistung/Klausur von 90 Minuten. Für die Zulassung zur Prüfung muss das Praktikum erfolgreich absolviert werden. Es werden sechs ECTS für die erfolgreiche Teilnahme an dem Modul vergeben.

Als Lernziel ist in der Modulbeschreibung festgehalten:

„Die Studierenden haben Fähigkeiten und grundlegende Erfahrungen mit der Programmierung in einer ersten Programmiersprache wie Java und Python. Dabei liegt der Fokus auf allgemeinen objektorientierten Programmierkonzepten aus der Sicht der Softwaretechnik.

Die Studierenden beherrschen das Entwickeln und Implementieren von Python- und Java-Programmen unter Verwendung einer Entwicklungsumgebung.“¹

Die Durchführung als online-unterstütztes Modul wurde im Wintersemester 2015/16 erprobt und wissenschaftlich begleitet.

Konzeptionelle Überlegungen zur Umgestaltung des Moduls

Das Modul „Grundlagen der Programmierung I“ bildet den Ausgangspunkt für das Studium der Informatik. Es ist für alle Studierenden der Informatik verbindlich im ersten Semester zu besuchen. Frühere Erfahrungen haben gezeigt, dass die Studierenden sehr unterschiedliche Vorkenntnisse in der Programmierung haben. Aufgrund von personellen Umstrukturierungen innerhalb der Fachgruppe Informatik wurden 2015 neue Lehrende für das Modul bestimmt. Diese Chance wurde genutzt, um das Modul neu zu konzipieren und um damit auch auf die Heterogenität bei den Programmierkenntnissen zu reagieren. Das Modul wurde als online-gestütztes Modul entwickelt und ein Kurs auf OPAL erstellt, um das Modul zu strukturieren.

Ein weiteres Ziel der Neugestaltung des Moduls war, einen Beitrag zu mehr Flexibilität in der Studiengestaltung für die Studierenden zu schaffen, um z. B. einer Berufstätigkeit nachzugehen. So sollte es möglich sein, das Modul auch ohne Teilnahme an den Präsenzveranstaltungen erfolgreich zu absolvieren.

¹ WHZ: Modulbeschreibung zu PTI 605: Grundlagen der Programmierung I im Modulux

Für das Modul „Grundlagen der Programmierung I“ wurde daher ein Flipped Classroom konzipiert. Dieser setzte sich aus einem hohen Selbstlernanteil und Präsenzveranstaltungen zusammen, wo das selbst Erarbeitete diskutiert und angewendet wurde.

Für die Strukturierung des Selbstlernanteils wurde auf das Buch „Einführung in Java mit Greenfoot“ zurückgegriffen, da es anhand von Beispielen die grundlegenden Konzepte der objektorientierten Programmierung in Java beschreibt und in mehrere in sich geschlossene Themenkomplexe aufgeteilt ist.

Eine wichtige Ergänzung des Buches stellen die frei verfügbare Greenfoot-Entwicklungsumgebung und die im Quellcode verfügbaren Beispielprojekte und Übungen zur Vertiefung dar. Die Studierenden sollten sich jede Woche ein Kapitel des Buches in ihrem eigenen Lerntempo erarbeiten und waren verpflichtet Fragen an das Gelesene zu formulieren und auf die Lernplattform hochzuladen. In der Präsenzveranstaltung wurden die Fragen durch die Lehrenden beantwortet. Die Präsentationen wurden im Nachgang auch auf OPAL zur Verfügung gestellt, so dass sich die Studierenden die Antworten auch im Selbststudium erarbeiten konnten, ohne an der Vorlesung teilzunehmen.

Ergänzt wurde die Vorlesung durch Programmierübungen, die live vorgeführt wurden und einen praktischen Teil, in dem die Studierenden selbst programmieren konnten. Auch diese Übungen wurden auf OPAL zur Verfügung gestellt. Das Online-Angebot wurde weiterhin um Lehrvideos zu grundlegenden Fragen der Programmierung ergänzt.

Herangehensweise

Innerhalb dieses Moduls fand auch die Auftaktveranstaltung zum Studium der Informatik statt. Daher wurde der erste Präsenztermin dazu genutzt, den Studierenden Informationen zu ihrem Studium zu geben. Dabei wurden auch die Erwartungen der Studierenden an ihr Informatikstudium abgefragt. Aufgrund der hohen Selbstlernanteile in Modulen der Informatik gab es weiterhin eine Einführung zum Thema „Wie lese ich wissenschaftliche Texte“ durch den Mitarbeiter für Hochschuldidaktik.

Innerhalb des ersten Präsenztermins wurden auch das Modell des Flipped Classrooms für das Modul „Grundlagen der Programmierung I“ und der dazugehörige Online-Kurs auf OPAL vorgestellt und erklärt.

Innerhalb des Selbststudienteils des Flipped Classroom mussten sich die Studierenden jede Woche ein Kapitel des Buchs „Einführung in Java mit Greenfoot“ erarbeiten, die Lösung der dazugehörigen Übungsaufgaben war freiwillig. Sie waren verpflichtet, drei Fragen zu dem Kapitel in OPAL hochzuladen. Diese Fragen konnten Punkte umfassen, die sie nicht verstanden haben, zu Funktionsweisen und zum Nutzen verschiedener Programmierkonzepte. Hatten die Studierenden keine Fragen zum Kapitel, konnten

Vorschläge für Prüfungsfragen eingereicht werden. Dabei sahen die Studierenden nicht die Fragen der anderen:

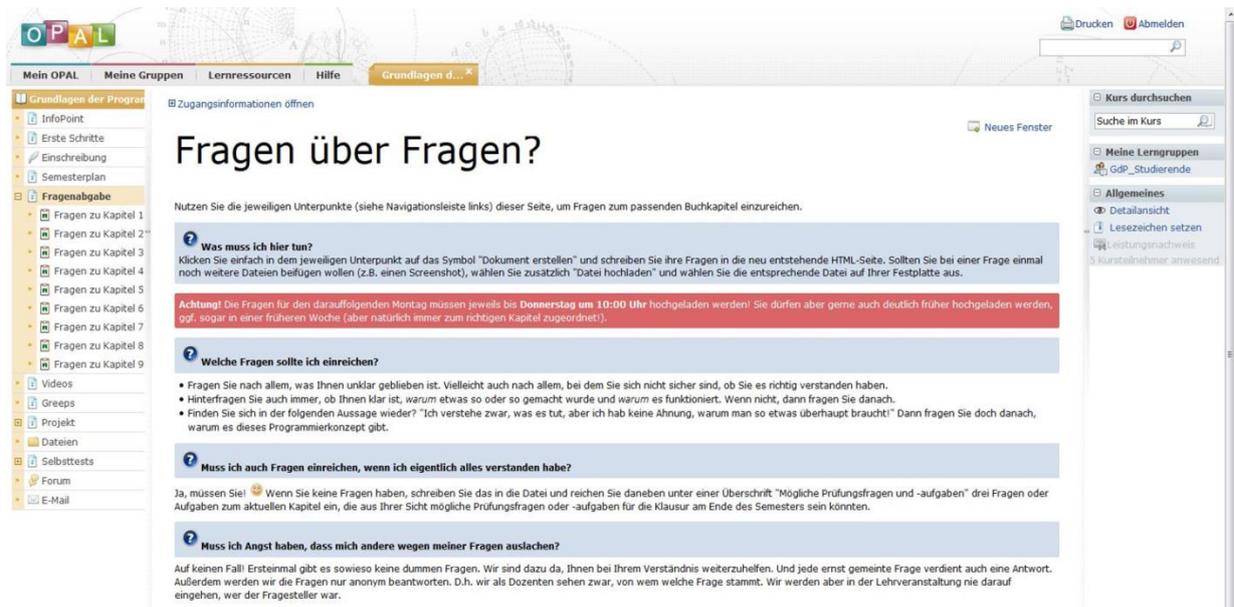


Abbildung 1: Darstellung der Erklärung zur Fragenabgabe im OPAL

Die eingegangenen Fragen (zu jedem Kapitel) wurden durch den Mitarbeiter für lehrbezogene Aufgaben des Projekts vorsortiert. Auf dieser Basis erarbeiteten die Lehrenden die Antworten auf die Fragen. Diese wurden in der Präsenzveranstaltung präsentiert. Im Anschluss an diese Präsentationen gab es Übungsaufgaben in Form eines Live-Hackings. Bei diesem führte der Lehrende selbst eine Programmierung durch, die die Studierenden live an ihren eigenen Rechnern mitverfolgten. Diesem folgte ein Praktikumsteil bestehend aus Programmieraufgaben für die Studierenden. Die Präsentationen der Antworten, die Übungen zum Live-Hacking und die Programmieraufgaben wurden im Anschluss auf der Lernplattform zur Verfügung gestellt. So konnten auch die Studierenden die Inhalte der Präsenzveranstaltung nachvollziehen, die nicht daran teilnehmen konnten.

Der Online-Kurs wurde noch ergänzt durch Videos, die Tipps zu Java und Greenfoot sowie zu Konzepten der Java-Programmierung enthielten. Diese Videos wurden mit dem E-Learning-Werkzeug „Camtasia“ erstellt. Außerdem wurden im OPAL-Kurs noch drei Selbsttests für die Studierenden zur Verfügung gestellt. Damit konnten sie testen, welche Themen sie noch einmal vertiefen sollten oder an welchen Stellen sie vielleicht auch etwas falsch verstanden haben. Nach Beenden des Tests war es möglich sich die korrekten Lösungen anzeigen zu lassen und bei offenen Fragen nochmals die Dozenten zu kontaktieren.

Nachdem das Buch „Einführung in Java mit Greenfoot“ mit dem Flipped Classroom durchgearbeitet war, gab es noch zwei Projektarbeiten für die Studierenden. In einem ersten Projekt, an dem die Teilnahme freiwillig war, handelte es sich um einen kleinen Programmierwettbewerb. Bei diesem sollten die Studierenden in Teams von zwei

Personen so genannte „Greeps“ programmieren. Dabei handelte es sich um kleine insektenartige Kreaturen, die durch Programmierung im Raum bewegt werden sollten.

Das zweite Projekt zum Ende des Semesters war für alle Studierenden des Moduls verpflichtend als Prüfungsvorleistung definiert. Hier sollten die Studierenden in Teams von zwei Personen ein eigenes Greenfoot-Projekt entwickeln. Dabei konnte es sich um ein Spiel oder eine Simulation handeln. Jedes Team entwickelte dazu eine eigene Projektidee und setzte diese nach Abstimmung mit den Dozenten über einen Zeitraum von vier Wochen selbstständig um.

Die Projektergebnisse mussten kurz in der letzten Präsenzveranstaltung präsentiert werden. Die Gruppenfindung und Koordination der Projektarbeit sowie die Abgabe der Projekte wurden ebenfalls über den OPAL-Kurs gesteuert:

Projekt als Teil der Prüfungsvorleistung in „Grundlagen der Programmierung 1“, Wintersemester 2015/16

Aufgabenstellung

Entwickeln Sie ein eigenes Greenfoot-Projekt. Dabei kann es sich um ein Spiel oder eine Simulation handeln. Jedes Team (bestehend aus zwei Studierenden) entwickelt dazu eine eigene Projektidee und setzt diese nach Abstimmung mit den Dozenten über einen Zeitraum von vier Wochen selbstständig um. Das Projekt muss auch über eine grafische Oberfläche verfügen.

Randbedingungen

- Das Projekt darf nicht auf einem bereits vorhandenen Projekt aufbauen oder dieses lediglich erweitern.
- Das Projekt muss einen hinreichenden Umfang und eine hinreichende Komplexität aufweisen. Die endgültige Einschätzung dazu im Einzelfall nehmen die Dozenten vor. Als Anhaltspunkt für eine geeignete Komplexität können Sie die Szenarien `marbles` und `lifs` aus den Buchszenarien zu Kapitel 10 nutzen.
- Die Klassen sollen sauber programmiert sein, d.h. mit aussagefähigen Variablennamen, korrekter Einrückung und vollständigen JavaDoc-Kommentaren.
- Die Abgabe erfolgt termingerecht über den OPAL-Kurs.
- Nach der Abgabe stellt jedes Team sein Projekt vor der Gruppe vor.

Zur Verwendung von Quellen

Die Studierenden sind verpflichtet, alle Quellen, die zur Umsetzung des Projekts genutzt wurden (Quellcode, Pseudocode, Algorithmen, Spielregeln, Bilder, Audio-Dateien etc.), in nachvollziehbarer Form anzugeben. Dazu wird zusammen mit dem Projekt eine PDF-Datei eingereicht, in der in tabellarischer Form sowohl die Quelle angegeben wird als auch wozu diese Quelle genutzt wurde.

Mit der Abgabe im OPAL-Kurs versichern die Studierenden, dass sie keine weiteren Quellen zur Umsetzung des Projekts genutzt haben. Bei Übernahme von Quellcode ist dies zusätzlich direkt im Projekt an der entsprechenden Stelle durch einen Quellcodekommentar deutlich zu machen.

Wird gegen diese Vorgaben verstoßen, wird dies als Plagiat und Betrugsversuch gewertet, was zur Nichtanerkennung des Projekts als Prüfungsvorleistung führt.

Das Projekt muss im Wesentlichen aus selbst entwickeltem Quellcode bestehen.

Ablauf und Termine sowie damit verbundene Teilaufgaben

Hinweis: Achten Sie sorgfältig darauf, regelmäßig eine geeignete Sicherungskopie Ihrer Projektarbeit anzulegen! Auch ein Festplatten-crash entschuldigt nicht für eine nicht erfolgte termingerechte Abgabe.

Abbildung 2: Darstellung und Steuerung der Projektarbeit im OPAL

Erprobung im Wintersemester 2015/2016 (Oktober bis Januar)

In der ersten Präsenzvorlesung des Moduls im Oktober 2015 wurde neben der Einführung in das Studium, dem Infoblock zum „Lesen wissenschaftlicher Texte“ und zur Erklärung der Methode des Flipped Classrooms auch der Online-Kurs und der Umgang mit OPAL durch den Mitarbeiter für lehrbezogene Aufgaben vorgestellt.

Für das Modul hatten sich 48 Studierende angemeldet. Da dieses Modul das erste Modul im Studiengang Informatik darstellt, hatten die Studierenden noch keine Erfahrungen mit der Arbeit in OPAL. Für die Teilnahme an dem Online-Kurs mussten sich die Studierenden im OPAL einschreiben. Um die Teilnahme an dem Selbststudium verpflichtend zu regeln, wurde die Fragenabgabe für sieben von neun Kapiteln des Buches als Prüfungsvoraussetzung definiert. Einige Studierende brachen im Laufe des Semesters das Modul ab. Fünf Studierende machten von der Möglichkeit Gebrauch sich die Inhalte ohne Besuch der Präsenzveranstaltungen online selbst zu erarbeiten. Sie nahmen dann erst zu der Ergebnispräsentation der Projektarbeiten am Ende des Semesters teil und traten auch die Prüfung an. Teilweise nutzten sie die Möglichkeit

außerhalb der Präsenzveranstaltungen Konsultationen bei den Lehrenden wahrzunehmen. An dem als Prüfungsvorleistung definierten Softwareentwicklungsprojekt haben 38 Studierende teilgenommen, wovon 29 Personen bestanden haben. Zu der schriftlichen Klausur sind dann 28 Studierende angetreten.

Da die Teilnahme an dem Online-Kurs verpflichtend geregelt war, konnte über das gesamte Semester eine aktive Beteiligung der Studierenden gewährleistet werden.

Wissenschaftliche Begleitung

Da das Modul das Einstiegsmodul in das Informatikstudium darstellt, wurden in der Auftaktveranstaltung (Ende September) die Studierenden befragt, was sie für Erwartungen an das Studium der Informatik haben. Es wurden sowohl ihre Erwartungen an die Lehrenden, als auch an ihre Kommilitonen als auch die an sich selbst abgefragt. Die Ergebnisse wurden an der Tafel gesammelt und als Fotoprotokoll dokumentiert. Die Studierenden äußerten u. a. den Wunsch, dass sie sich in ihrem Studium Inhalte selbst erarbeiten können.

Fünf Wochen nach Semesterbeginn (Anfang November) wurden die Frage der Erwartungen an das Studium nochmals aufgegriffen, ergänzt um Fragen, wie es derzeit im Studium läuft und auch zur Durchführung des Flipped Classroom. Dafür wurde eine Live-Umfrage ein sogenanntes Tweedback² durchgeführt. Damit war eine anonyme Antwortabgabe durch die Studierenden möglich und die Ergebnisse konnten gleich vor Ort ausgewertet und diskutiert werden. Außerdem konnte die Umfrage um weitere Fragen ergänzt werden, die sich aus der Diskussion ergaben. Eine der Fragen umfasste den Umgang der Studierenden mit dem Greenfoot-Buch. Dabei zeigte sich, dass nur 7 % das Kapitel eher nur schnell überfliegen, um schnell ihre Fragen hochladen zu können. Alle anderen lasen es genau und lösten größtenteils auch die Übungsaufgaben am Ende eines Kapitels:

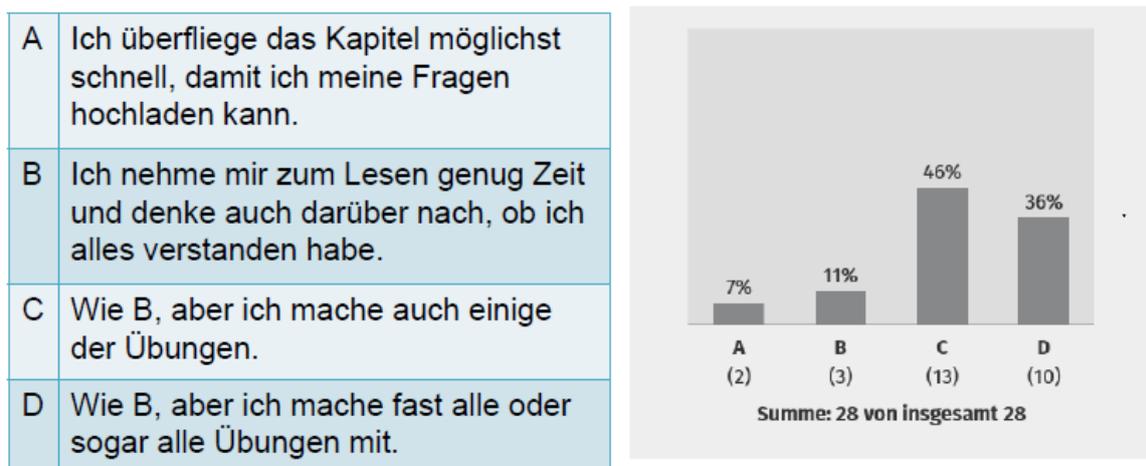


Abbildung 3: Ergebnis des Tweedbacks zum Umgang mit dem Greenfoot-Buch

² www.tweedback.de

Durch den Mitarbeiter für Hochschuldidaktik wurde in der Mitte des Semesters (Ende November) ein qualitatives Evaluationsverfahren durchgeführt. Dieses Verfahren findet nicht erst zum Semesterende statt, um den Lehrenden die Möglichkeit zu geben, schon innerhalb des Semesters Veränderungen vorzunehmen um auf die Lernbedürfnisse der Studierenden zu reagieren. Bei diesem „Teaching Analysis Poll“ (TAP) sammelten die Studierenden in lernförderliche und -hinderliche Faktoren, die sie beobachtet haben und erarbeiten Verbesserungsvorschläge. Diese Punkte wurden gesammelt, vor der Gruppe vorgestellt und die anderen Studierenden konnten abstimmen, ob sie die Aspekte teilen. In der Ergebnisdarstellung wurde dann das Abstimmungsverhalten in den Kategorien vereinzelt, häufig und durchgehend dargestellt. Die Ergebnisse des TAP wurden durch das Projektteam nochmals ausgewertet in Bezug auf die Frage, inwieweit die Studierenden Lerneffekte, Lernerschwernisse und Verbesserungsvorschläge für die Durchführung des Flipped Classrooms und die Unterstützung durch den Online-Kurs herausgearbeitet haben:

Nennung/ Zustimmung	vereinzelt	häufig	durchgängig/ alle
Lerneffekte	Arbeitsform Flipped Classroom	Greenfoot-Buch	Livehacking Übungsaufgaben
Lern- erschwernisse		hoher Zeitaufwand durch das Lesen eines Kapitels, Druck der Fragenabgabe	Komplexität der Praktikums- aufgaben
Verbesserungs- vorschläge	Rückkehr zu klassischer Vorlesung mit Übungsaufgaben für Zuhause		Mehr Mitmach- aufgaben Vorlage der Lösungen

Abbildung 4: Ergebnisse des TAPs zur Durchführung des Moduls als Flipped Classroom

Die Ergebnisse zeigen, dass zu diesem Zeitpunkt des Semesters einige Studierende sich wieder eine klassische Form der Durchführung der Lehrveranstaltung wünschten. Häufiger wurden das regelmäßige Lesen des Kapitels und die Fragenabgabe als Lernerschwernis empfunden, was aber auch darauf zurückzuführen ist, dass einigen Studierenden das Selbststudium zu aufwändig war. Am meisten lernten die Studierenden nach eigener Einschätzung durch das Live-Hacking in der Präsenzveranstaltung.

Am Ende des Semesters wurden für das Modul die Prüfungsergebnisse ausgewertet, auch im Vergleich zu einem der vorherigen Jahrgänge, der die gleiche Programmiersprache gelernt hatte.³ Die Prüfungsergebnisse haben sich im Durchschnitt geringfügig von 3,4 auf 3,0 verbessert. Allerdings ist die Durchfallquote von 30 % auf 18 % gesunken. Es wurde weiterhin geschaut, ob die Prüfungsergebnisse in Zusammenhang mit der Teilnahme an der Präsenzveranstaltung stehen. Auch, wenn diejenigen Studierenden, die regelmäßig bei der Präsenzveranstaltung waren, bessere Ergebnisse erzielt haben, haben auch drei der fünf Studierenden, die die Veranstaltung nicht regelmäßig besucht haben, das Modul erfolgreich absolviert.

Zum Ende des Semesters wurde weiterhin ein Feedbackgespräch mit den Lehrenden durchgeführt. Dieses Feedback fiel grundsätzlich positiv aus und es wurde sich dafür ausgesprochen „Grundlagen der Programmierung I“ auch in den nächsten Jahren in dieser Form durchzuführen. Aufgrund früherer Erfahrungen, hatten sie den Eindruck, dass diese Form der Durchführung einen positiven Einfluss auf die Motivation der Studierenden genommen hat und die Studierenden somit bereits im ersten Semester lernten sich selbst Wissen anzueignen und in Teams zusammenzuarbeiten. Die Lehrenden wollen beobachten, ob sich dadurch auch langfristig positive Effekte im weiteren Studienverlauf zeigen. Nach der Einschätzung der Lehrenden hat sich im Laufe des Semesters die Qualität der eingereichten Fragen erhöht. Mit den Ergebnissen der Prüfungen waren sie zufrieden. Negativ bewertet wurde der hohe Arbeitsaufwand für die Lehrenden bei der Zusammenstellung und Beantwortung der Fragen. Hier erhoffen sie sich, dass sich dieser Aufwand mit dem nächsten Durchgang reduziert.

Fazit

Mit der Durchführung des Moduls als Flipped Classroom ist es gelungen, die Studierenden zur Nutzung des Online-Kurses zu animieren. Vor allem die Verbindung zwischen Selbststudium, der Präsenzvorlesung und den Live-Übungen scheint mit Blick auf die in der wissenschaftlichen Begleitung erhobenen Ergebnisse erfolgreich gewesen zu sein. Daher ist dieser Blended-Learning-Ansatz weiter zu verfolgen.

Die Durchführung des Moduls hat zu mehr Flexibilität des Studiums beigetragen. Die Studierenden konnten nicht nur ihr eigenes Lerntempo bestimmen, sondern hatten auch die Möglichkeit auf die Teilnahme an der Präsenzveranstaltung zu verzichten und sich alles selbst zu erarbeiten, wovon fünf Studierende Gebrauch machten.

Die Gestaltung des Moduls als Flipped Classroom kann als ein Erfolgsmodell bezeichnet werden, auch wenn es einen hohen Zeitaufwand für die Lehrenden bedeutet. Das Modell ist auch auf andere Module der Studiengänge der Informatik übertragen worden. Durch die Bereitschaft der Lehrenden auch unabhängig vom Projekt bei dem

³ Hierbei ist zu beachten, dass jeder Studienjahrgang unterschiedliche Voraussetzungen, aber auch Vorkenntnisse in das Studium mitbringt. Mit dem Vergleich der Prüfungsnoten, kann daher nur bedingt eine Aussage darüber getroffen werden, ob ein Modul erfolgreich gelaufen ist.

Modell zu bleiben, ist die Nachhaltigkeit gesichert und das Modul „Grundlagen der Programmierung I“ wurde auch im Wintersemester 2016/17 wieder in dieser Form durchgeführt. Erfahrungen der Erprobung flossen in die erneute Durchführung mit ein und es wurde an einzelnen Aspekten nachjustiert. So wurden weitere Videos zu den Grundfragen der Programmierung erstellt und die Fragenabgabe angepasst. Außerdem wurden die Videos Untertitelt, damit auch Studierende mit einer Hörbeeinträchtigung die Videos als Lerngrundlage nutzen können. Diese Videos werden auch eingesetzt, um Studienwerbung für die Informatik an der WHZ bei Gehörlosen zu machen.

1.3 Grundlagen der Programmierung II

Rahmendaten des Moduls

Das Modul „Grundlagen der Programmierung II“ ist für alle Studierenden der Informatik verpflichtend im zweiten Semester zu besuchen.

Für das Modul ist ein Workload in Höhe von 180 Stunden angesetzt. Es setzt sich aus zwei Semesterwochenstunden (SWS) Vorlesung und 2 SWS Praktikum zusammen, die in Präsenz (60 Stunden) durchgeführt werden. 120 Stunden und damit ein sehr großer Anteil sind für das Selbststudium der Studierenden und die Vor- und Nachbereitung vorgesehen. Das Modul schließt mit einer alternativen Prüfungsleistung ab, die in Form eines Softwareprojekts erbracht wird. Dies wird ergänzt durch ein mündliches Prüfungsgespräch von 20 Minuten. Es werden sechs ECTS für die erfolgreiche Teilnahme an dem Modul vergeben.

Als Lernziel ist in der Modulbeschreibung festgehalten:

„Die Studierenden haben erweiterte Fähigkeiten und fortschreitende Erfahrungen mit der Programmierung in Java. Dabei liegt der Fokus auf allgemeinen objektorientierten Softwareentwicklungstechniken.

Die Studierenden beherrschen das Entwickeln und Implementieren von anspruchsvolleren Java-Programmen unter Verwendung der Entwicklungsumgebung Eclipse.“⁴

Die Durchführung als online-unterstütztes Modul wurde im Sommersemester 2016 erprobt und wissenschaftlich begleitet.

Konzeptionelle Überlegungen

Die Teilnehmenden des Moduls „Grundlagen der Programmierung II“ waren die gleichen wie bei „Grundlagen der Programmierung I“. Ihnen war die Durchführung als Flipped Classroom mit Unterstützung der Online-Lernplattform bereits von ihrem ersten Semester vertraut.

Aufgrund der positiven Erfahrungen mit dem Flipped Classroom in „Grundlagen der Programmierung I“ haben sich die Lehrenden entschlossen, auch „Grundlagen der Programmierung II“ als Flipped Classroom durchzuführen. In diesem Modul lernen die Studierenden komplexere Softwaretechniken und Entwicklungsumgebungen kennen.

Ziel der Neugestaltung des Moduls war es auch hier, einen Beitrag zu mehr Flexibilität in der Studiengestaltung für die Studierenden zu schaffen. Durch die online-unterstützte Durchführung konnten Präsenzveranstaltungen reduziert werden. Eine Teilnahme an

⁴ WHZ: Modulbeschreibung zu PTI 601: Grundlagen der Programmierung II im Modulux

den Vorlesungen war jedoch notwendig, um in Teams parallel zur Vorlesung verschiedene Softwareprojekte durchführen zu können, die sowohl die Prüfungsvorleistung als auch die alternative Prüfungsleistung darstellten. Weiterhin wurden Live-Hackings und ein Video zur Darstellung eines bestimmten Programmierungsproblems eingesetzt.

Herangehensweise

Der für das Modul „Grundlagen der Programmierung II“ konzipierte Flipped Classroom konnte nicht auf ein einzelnes Buch zurückgreifen, sondern es mussten verschiedene Quellen einbezogen werden. Für die Strukturierung des Selbstlernanteils wurden daher verschieden Texte und Internetdokumente zusammengestellt, auch englischsprachige Quellen wurden mit einbezogen. Basierend auf den vorliegenden Quellen wurde der Ablauf für das Modul festgelegt, wobei die Komplexität der Texte im Laufe des Semesters zugenommen hat. Die Texte und Quellen wurden auf OPAL zur Verfügung gestellt. Auch in diesem Modul waren die Studierenden verpflichtet sich jede Woche ein bis zwei Texte im Selbststudium zu erarbeiten und wieder drei Fragen zu diesen Texten zu formulieren und über OPAL hochzuladen. In OPAL wurden auch ergänzende Quellen zur Verfügung gestellt, die von den Studierenden zusätzlich bearbeitet werden konnten. Die Lehrenden haben die eingereichten Fragen sortiert und die Antworten vorbereitet. Diese wurden wieder innerhalb der Vorlesung präsentiert. Im Anschluss an die Vorlesung erarbeiteten die Studierenden in Zweier-Teams und in Kleingruppen zwei größere Softwareprojekte, die zum einen die Prüfungsvorleistung und zum anderen die alternative Prüfungsleistung darstellten. Für die Koordination der Softwareprojekte wurde ebenfalls OPAL eingesetzt.

Live-Hacking wurde nur gelegentlich zur Illustration eines Beispiels durch die Lehrenden eingesetzt. Für ein besonders komplexes Thema⁵ wurde wieder ein Video zur Erklärung erstellt. Den Abschluss bildete ein mündliches Prüfungsgespräch, in welchem auch bestimmte Aspekte des Softwareprojekte der Studierenden hinterfragt wurden.

Erprobung im Sommersemester 2016 (April bis Juni)

In der ersten Präsenzvorlesung wurde diesmal nur kurz das Modell des Flipped Classrooms vorgestellt, da dies den Studierenden bereits aus dem vorherigen Semester bekannt war. Es wurde bereits in der ersten Woche mit Anwendungsübungen gestartet. Für die Teilnahme an dem Online-Kurs mussten sich die Studierenden im OPAL einschreiben. Um die Teilnahme an dem Selbststudium verpflichtend zu regeln, wurde die Fragenabgabe zu mindestens 80 % der zu den zur Verfügung gestellten Quellen als Prüfungsvoraussetzung definiert. Für das Modul haben sich 46 Studierende angemeldet. Für die Prüfungszulassung war die erfolgreiche Absolvierung eines

⁵ Installation des Spock-Testframeworks in Eclipse

Softwareprojekts zu „Shapes“⁶ notwendig, welches in Zweierteams durchgeführt wurde. 27 Studierende sind zu der alternativen Prüfungsleistung und dem Prüfungsgespräch angetreten. Die alternative Prüfungsleistung war ein in Kleingruppen erstelltes Softwareprojekt zum Thema „SmsHandy“.⁷

Wissenschaftliche Begleitung

Durch den Mitarbeiter für Hochschuldidaktik wurde in der Mitte des Semesters (Ende April) wieder das qualitative Evaluationsverfahren TAP durchgeführt. Die Studierenden sammelten in kleinen Gruppen lernförderliche und -hinderliche Faktoren, die sie beobachtet haben und erarbeiten Verbesserungsvorschläge. Diese Punkte wurden gesammelt, vor der Gruppe vorgestellt und die anderen Studierenden konnten abstimmen, ob sie die Aspekte teilen. In der Ergebnisdarstellung wurde dann das Abstimmungsverhalten in den Kategorien vereinzelt, häufig und durchgehend dargestellt. Die Ergebnisse dieses TAPs wurden wieder durch das Projektteam ausgewertet in Bezug auf die Frage, inwieweit die Studierenden Lerneffekte, Lernerschwernisse und Verbesserungsvorschläge für die Durchführung des Flipped Classrooms und die Unterstützung durch den Online-Kurs herausgearbeitet haben:

Nennung/ Zustimmung	vereinzelt	häufig
Lerneffekte	Texte lesen	Selbststudium Gruppenarbeit in der Vorlesung
Lern- erschwernisse	Druck/Stress des Selbststudiums und der Fragenabgabe	Lehrmethode des Flipped Classrooms Schwierigkeiten bei Fragenfindung
Verbesserungs- vorschläge		Rückkehr zur klassischer Vorlesung, kein Flipped Classroom mehr Freiwillige Fragenabgabe

Abbildung 5: Ergebnisse des TAPs zur Durchführung des Moduls als Flipped Classroom

⁶ Shapes sind grafische Objekte, die auf einem Computerbildschirm dargestellt werden.

⁷ Dafür sollte ein Projekt mit dem Namen SmsHandy programmiert werden, in dem ein Handynetz simuliert wird, dessen Handys ausschließlich SMS senden können.

Die Ergebnisse zeigten, dass ein Großteil der Studierenden gerne wieder zu einer klassischen Vorlesung zurückkehren will. Das Selbststudium und die Fragenabgabe waren ihnen zu aufwändig.

Auch in dieser Veranstaltung wurde vor Beginn der Prüfungszeit wieder eine Live-Umfrage mit Tweedback durchgeführt. Die Fragen waren diesmal umfangreicher und thematisierten den Umgang mit der Lernplattform und dem Flipped Classroom nochmals konkreter:

Mein Umgang mit den online zur Verfügung gestellten Texten zur Programmierung (1)

A	Ich habe den Text möglichst schnell überflogen, damit ich meine Fragen hochladen kann.
B	Ich habe mir zum Lesen genug Zeit genommen und auch darüber nachgedacht, ob ich alles verstanden habe.

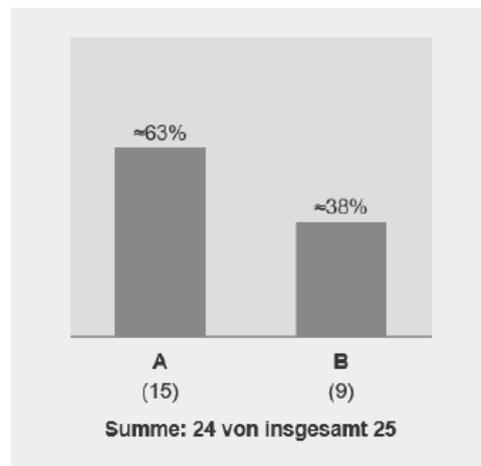


Abbildung 6: Ergebnis des Tweedbacks zu den Texten

Hier zeigt sich, dass die Studierenden doch eher die Intention hatten, schnell die Texte durchzulesen, als sich die Zeit zu nehmen, sie richtig zu verstehen.

Mein Umgang mit der Fragenabgabe (2)

A	Ich habe meistens Alibi-Fragen hochgeladen.
B	Ich habe meistens Fragen hochgeladen, an deren Beantwortung ich wirklich interessiert war.
C	Ich habe meistens mögliche Prüfungsfragen hochgeladen.

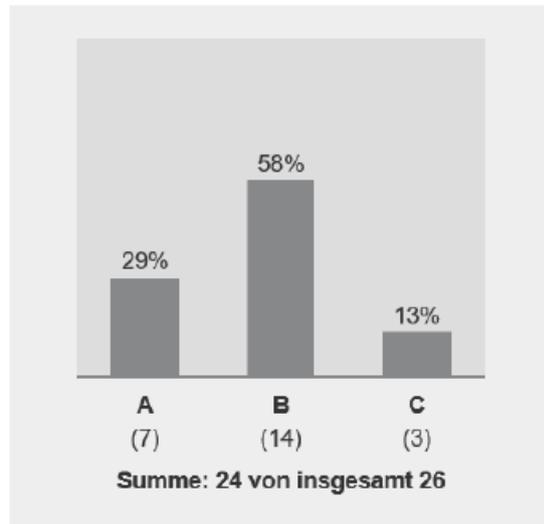


Abbildung 7: Ergebnis des Tweedbacks zur Fragenabgabe

Die Fragenabgabe wurde im Gegensatz zu dem Lesen der Texte ernster genommen. Der Arbeitsaufwand für das Modul wurde von 56 % der Studierenden als angemessen bewertet.

Die Struktur des OPAL-Kurses wurde mit 96 % sehr positiv bewertet. Der Kurs scheint das Lernen unterstützt zu haben. Nur 24 % der Studierenden haben angegeben, dass sie den Kurs nur genutzt haben, weil er verpflichtend für die Erbringung der Prüfungsvorleistung war.

Durch die Durchführung der Lehrveranstaltung als Flipped-Classroom habe ich mehr gelernt als in einer vergleichbaren klassischen Lehrveranstaltung. (9)

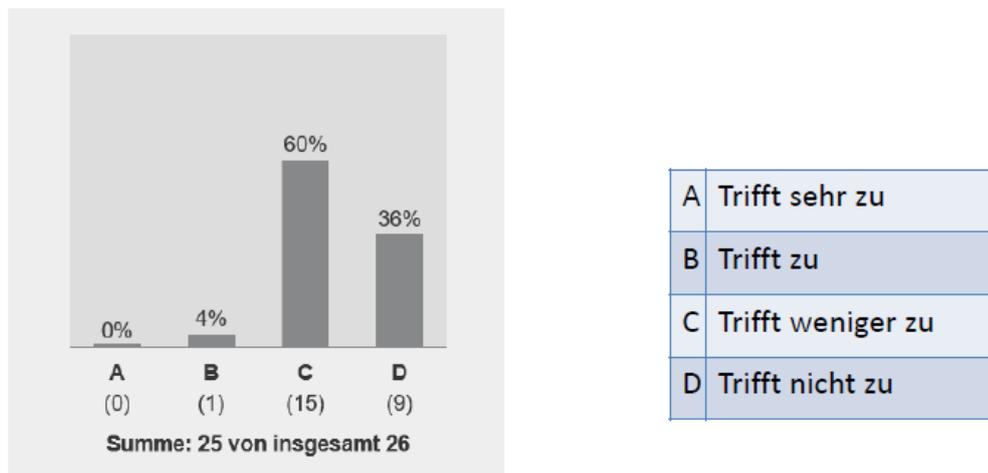


Abbildung 8: Ergebnis des Tweedbacks zum Lerneffekt des Flipped Classrooms

Der Lerneffekt wurde mit der Durchführung als Flipped Classroom im Vergleich zu einer klassischen Vorlesung von 96 % als nicht größer eingeschätzt. Die online-basierte Begleitung hat bei 58 % der Befragten dazu geführt, dass sie sich dadurch besser auf das Prüfungsgespräch vorbereitet fühlten.

Die Durchführung des Moduls als Flipped Classroom wurde aufgrund des hohen Selbstlernanteils von den Studierenden eher kritisch bewertet. Sie wünschten sich mehrheitlich eine Rückkehr zur klassischen Vorlesung. Vergleicht man jedoch die Ergebnisse der mündlichen Prüfungsgespräche fallen diese deutlich besser aus als in den Vorjahren. Im Durchschnitt wurde ein Ergebnis von 1,9 erreicht. 2015 hat der Durchschnitt noch bei 3,1 gelegen, 2014 bei 2,3. Dies könnte ein Indiz dafür sein, dass durch die Anwendung des Flipped Classrooms der Lerneffekt höher ist, als in einer klassischen Vorlesung.

Fazit

Die Lehrenden schätzten die Durchführungsform als Flipped Classroom, trotz der Kritik der Studierenden, als positiv ein. Dies zeigten auch die Prüfungsergebnisse. Das Modul soll in dieser Form weitergeführt werden. Es wird im Sommersemester 2017 erneut in dieser Form durch die Lehrenden durchgeführt. Erfahrungen der Erprobung fließen in eine Anpassung des Konzepts mit ein.

Das Konzept des Flipped Classroom wird in Ansätzen in immer mehr Veranstaltungen in der Informatik übertragen.

1.4 Entwicklung verteilter Anwendungen

Rahmendaten des Moduls

Modulname	Entwicklung verteilter Anwendungen
Modulnummer	PTI 793
Lehrende des Moduls	Prof. Wolfgang Golubski, Oliver Arnold
Zuordnung zum Curriculum: Studierende welcher Studiengänge nehmen an den Modulen teil und in welchem Semester?	Informatik (B.Sc. im 3. Semester)
Workload	120 h
ECTS-Credits	4
Veranschlagte Präsenzzeiten	45 h
Veranschlagte Selbststudienzeiten	75 h
Prüfungsvorleistung? In welcher Form?	„Testat“ (Softwareentwicklungsprojekt bestehend aus mehreren Teilaufgaben, die jeweils zu festen Terminen („Meilensteine“) erledigt sein müssen)
Prüfungsleistung, in welcher Form?	mündlich
Beschreibung der Lernziele	Die Studierenden haben grundlegende Kenntnisse zu Architekturkonzepten. Sie können diese im Rahmen von Neuentwicklungen von verteilten Systemen und Anwendungen einsetzen. Anhand von Spring wird ein detaillierteres Verständnis von Technologien zur Web-Anwendungsentwicklung erworben. Durch die Bearbeitung eines umfangreicheren Projektes können die Studierenden Software entwickeln, designen und mit Komponenten-Techniken realisieren.
Zeitpunkt der Erprobung	WS 16/17

Inhalt und Methodik

Das Modul „Entwicklung verteilter Anwendungen“ wurde für den Bachelor-Studiengang Informatik neu entwickelt. Es wurden Inhalte aus einer umfangreicheren Veranstaltung aus dem Masterstudiengang in das dritte Bachelorsemester verlegt. Diese Verschiebung wurde notwendig, da viele Bachelor-Studierende diese Inhalte im Praxissemester benötigen. Allerdings macht die Verlagerung aus dem Master natürlich eine inhaltliche und didaktische Anpassung auf das Bachelorniveau notwendig.

Dafür wurden die vorhandenen Master- Unterlagen gesichtet und weiterführende und sehr komplexe Themen wegestrichen. Im Anschluss daran wurden neue Online-Quellen gesucht und ein Opal-Kurs angelegt.

Der gewählte methodische Ansatz orientierte sich sehr stark am Flipped-Classroom aus Grundlagen der Programmierung 1 und 2, d. h. es mussten wöchentlich Quellen gelesen und Fragen dazu hochgeladen werden. Die Fragenabgabe war jedoch in diesem Semester freiwillig und nicht als verpflichtende Prüfungsvorleistung definiert. Dies sollte den Studierenden die Möglichkeit geben, sich auch gegenseitig Fragen zu beantworten. Da im 14-täglichen Rhythmus ein Praktikum geplant war, sollte dieses Forum der Fragenabgabe auch eine Kollaborationsmöglichkeit darstellen.

Die Vermittlung der Konzepte geschah klassisch durch Präsentationen in der Präsenzzeit. Dies wurde durch die Vermittlung konkreter Vorgehensweisen bei der Softwareentwicklung und dem Umgang mit den verwendeten Werkzeugen durch Tutorial-Videos ergänzt. Diese Videos wurden durch den Dozent aufgenommen und durch den OHZ-Mitarbeiter nachbearbeitet. Zu Beginn des Semesters gab es mehrere Tutorial-Videos, um für das Projekt handlungsfähig zu sein. In der zweiten Semesterhälfte wurde der Fokus mehr auf Konzepte und Hintergründe gelegt. Die Projektarbeit lief parallel zur Veranstaltung und war durch zwei Meilensteine in sich gegliedert, die im Praktikumsteil der Präsenzveranstaltung abgenommen wurden.

Kursaufbau

Die Strukturierung des Kurses wurde auf der Plattform Opal durch einen Semesterplan deutlich gemacht. Hier wurden die zu lesenden Quellen angegeben und dazu wurden die Inhalte der Präsenzveranstaltung verdeutlicht. Gleichzeitig wurde hier auch der zeitliche Ablauf des Semesters dargestellt.

Semesterplan für Eva im Wintersemester 2016/2017

Wieso gibt es die Spalte "In der Quelle vorgestellte Konzepte"?
Hier wird ein Überblick gegeben dazu, was in den Quellen jeweils besprochen wurde. Das bietet einen guten Überblick, um wesentliche Konzepte kenntlich zu machen, und kann als Checkliste für die Prüfungsvorbereitung dienen.

Der Semesterplan hat noch vorläufigen Charakter und wird im Laufe des Semesters überarbeitet werden.

Wir sind sehr an weiteren passenden Quellen interessiert. Bitte teilen Sie uns mit, wenn Sie etwas dazu finden. Auch sind wir sehr an Ihrem Feedback zu den angegebenen Quellen interessiert. Welche Quellen fanden Sie hilfreich, verständlich etc.?

Lassen sich die Quellen 1:1 umsetzen?
Mitunter nicht. Wir üben in diesem Semester und dieser Lehrveranstaltung weiter den richtigen Umgang mit Quellen und Anleitungen aus dem Internet - eine sehr wichtige Fähigkeit für einen Informatiker. Sie werden oftmals feststellen, dass sich die Lösungen nicht 1:1 übernehmen lassen. Mitunter liegt Quellen eine andere Version der Software zugrunde, oft ist die Umsetzung in Spring Boot anders als in Spring, manchmal wird XML-Konfiguration beschrieben, obwohl wir Annotationen verwenden möchten usw. Durch diese Vorgehensweise lernen Sie, zu abstrahieren, Wesentliches zu erkennen und Quellen kritisch zu hinterfragen.

Datum	Bis dahin zu lesende Quelle	In der Quelle vorgestellte Konzepte	Inhalte Vorlesung und Praktika
05.10.2016			Einführung in die Veranstaltung, Organisatorisches, Hello-World-Beispiel mit Spring
12.10.2016	<ul style="list-style-type: none"> Spring Boot-Projekt in IntelliJ anlegen: https://www.jetbrains.com/help/idea/2016.2/creating-spring-boot-projects.html Grundlegendes Spring Boot-MVC-Tutorial http://spring.io/guides/gs/serving-web-content/ Grundlegende Einführung in Git: https://rogerbujdar.github.io/git-quickindex.de/html Dokumentation zu Git SCM: https://git-scm.com/docs 	Grundlegendes Spring-MVC-Beispiel (@Controller, @RequestMapping, @RequestParam, Thymeleaf, @SpringBootApplication), Anlegen eines Spring Boot-Projekts in IntelliJ IDEA, Grundlegende Git-Befehle sowie Dokumentation zum Tool	Autowiring, Bootstrap, Ordnerstruktur für Web-App, grundlegendes JPA Praktikum (beide Gruppen gemeinsam): Basisbeispiele Taschenrechner und Studentenverwaltung
19.10.2016	<ul style="list-style-type: none"> Überblick über Komponenten des Bootstrap-CSS-Frameworks: http://holderbootstrap.de/komponenten/ (Wenden Sie Bootstrap auf das vorgestellte HelloWorld-Beispiel an, Probieren Sie verschiedene Komponenten aus und fügen Sie eine Navigation über eine Art Navigationsleiste hinzu.) Überblick über Thymeleaf von einem der Autoren der Template Engine: https://www.youtube.com/watch?v=Gv9duzPHYoQ 	Anwendung von Bootstrap, Überblick über grundlegende Funktionalität und Anwendungsmöglichkeiten von Thymeleaf (u.a. Natural Templates, Page Composition)	Fortsetzung JPA (Schwerpunkt: Assoziationen) Praktikum (beide Gruppen gemeinsam): Basisbeispiele Taschenrechner und Studentenverwaltung
26.10.2016	<ul style="list-style-type: none"> Java Persistence Wikibook: https://en.wikibooks.org/wiki/Java_Persistence (Machen Sie sich mit dem Inhalt von Kapitel 5 groß vertraut und nutzen Sie das Buch zukünftig zum Nachschlagen.) Stellen Sie die erste Praktikumsaufgabe fertig. 		Fortsetzung JPA (Schwerpunkt: Assoziationen) Praktikum (beide Gruppen gemeinsam): Pizza-Projekt Teilaufgabe 1 (Datenmodell)
02.11.2016	<ul style="list-style-type: none"> Spring Data JPA Tutorial - Einführung: https://www.petrikainulainen.net/programming/spring-framework/spring-data-jpa-tutorial-introduction/ Spring Data JPA Tutorial - CRUD: https://www.petrikainulainen.net/programming/spring-framework/spring-data-jpa-tutorial-part-two-crud/ Spring Data JPA Tutorial - Einführung zu Query-Methoden: https://www.petrikainulainen.net/programming/spring-framework/spring-data-jpa-tutorial-introduction-to-query-methods/ 	Spring Data JPA, Spring Data Commons, Repository-Interfaces, CRUD-Operationen, Query-Methoden	Weiterentwicklung Wallo-Beispielprojekt + Fortsetzung JPA (Queries) Praktikum (Gruppe Z2): Pizza-Projekt Teilaufgabe 2 (Grundfunktionalität)
09.11.2016	<ul style="list-style-type: none"> Einführung in Spring Boot Security: http://bitczerwski.eu/2014/12/spring-boot-security-application/ (Empfehlung: Fertigen Sie eine Mind-Map zu den verschiedenen vorgestellten Konzepten der Spring Security und deren Zusammenhänge an.) Weiteres sehr gutes Tutorial zu Git: https://www.ralfbert.de/git/ 	Aufbau User-Entity, User-DTO, User-Service, Passwort-Handling, Java 8 Optional, Prüfung auf Eingabefehler (BindingResult, @InitBinder, Validator), CSRF, Spring Security Konfiguration, UserDetailsService, UserDetails, Remember-me, URL-basierte Autorisierung, Methodenbasierte Autorisierung, Sicherheit für Domänenobjekte, Zugriff auf aktuellen User	Refactoring Wallo-Beispielprojekt, Spring Security, Teil 1 Praktikum (Gruppe Z1): Pizza-Projekt Teilaufgabe 2 (Grundfunktionalität)

Abbildung 9: Ausschnitt Semesterplan

Die Fragenabgabe erfolgte nicht auf der Lernplattform Opal, sondern über Redmine auf dem Informatik-Projektserver. Hier ist das Forum besser geeignet, da das Opal-Wiki keine passende Quellcodedarstellung (Syntax-Highlighting) hat.

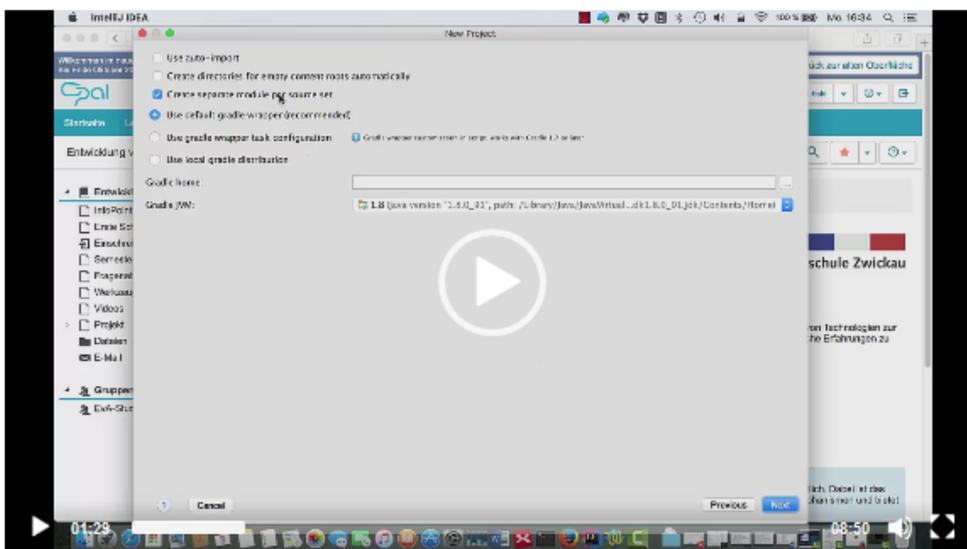
Auf der Lernplattform Opal wurden im Kurs zahlreiche Videos zur Verfügung gestellt die den Umgang mit den verwendeten Werkzeugen demonstrierten. Diese Videos wurden mit der Software Camtasia Studio erstellt. Dabei handelt es sich um einen professionellen Desktop-Camcorder, mit dem man seinen Bildschirm während einer Demonstration abfilmen kann.

Videos erste Lehrveranstaltung



Verwendete Code-Snippets sind unter <https://projektserver.informatik.fh-zwickau.de/projects/eva-2016/wiki/EvaPart1HelloWorld> zu finden.

Spring Intro



Hello World-Beispiel: Basic Tooling

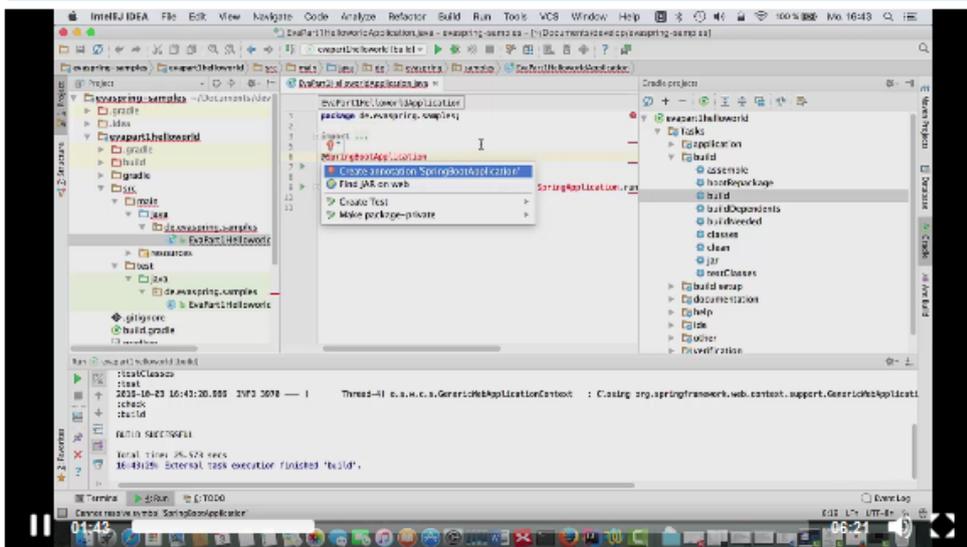


Abbildung 10: Lehrvideos auf Opal

Des Weiteren wurden im Opal-Kurs die Vorlesungs- und Praktikamaterialien zur Verfügung gestellt.

Verlauf der Erprobung

Da das Konzept aus Grundlagen der Programmierung I und II fortgeführt wurde, war keine Einweisung mehr notwendig.

Zu Beginn der Veranstaltung waren 23 Teilnehmende in den Kurs eingeschrieben, davon 4 weiblich. Zu Ende haben 12 Teilnehmende (1 weiblich) den Kurs geführt.

Die Teilnehmenden wurden durch Projektarbeit zu einer aktiven Beteiligung am Kurs animiert. Das zur Verfügung gestellte Forum für die freiwillige Fragenabgabe wurde im Semesterverlauf allerdings nur sporadisch genutzt.

Mit Hilfe des Semesterplans wurde der Online-Kurs stark strukturiert. Hier wurden Projektaufgaben und Lehrvideos Woche für Woche erstellt und im OPAL-Kurs hinterlegt.

Wissenschaftliche Begleitung:

Auf eine Befragung zu Beginn des Semesters wurde verzichtet, da es sich um die gleichen Studierenden handelte, die bereits an der Erprobung von Grundlagen der Programmierung I und II teilgenommen haben.

Dafür wurden zwei Tweedback-Befragungen durchgeführt. Das erste Tweedback wurde kurz nach Semesterbeginn durchgeführt. Es wurden folgende Fragen gestellt:

1. Mein Umgang mit dem Semesterplan
2. Mein Umgang mit dem Forum
3. Mein Arbeitsstand zur ersten Praktikumsaufgabe

Die Ergebnisse sind in Abbildung 3 - 5 dargestellt.

A	Ich lese im Wesentlichen alle Quellen gründlich durch.
B	Ich habe einige Quellen gründlich gelesen, andere eher nicht.
C	Ich habe die meisten Quellen eher oberflächlich durchgesehen.
D	Ich habe mich mit den Quellen größtenteils gar nicht beschäftigt.

Quiz: 1 (Quellenarbeit)

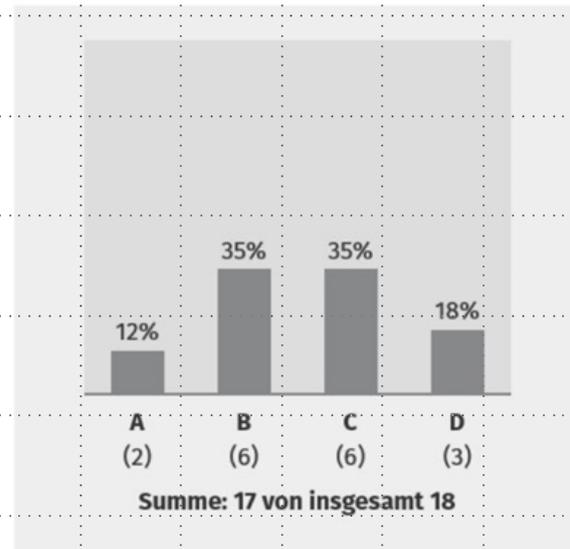


Abbildung 11: Umgang mit Semesterplan

A	Ich habe beide Teile der ersten Praktikumsaufgabe im Wesentlichen gelöst.
B	Ich habe die Taschenrechner-Aufgabe im Wesentlichen gelöst.
C	Ich habe die Aufgaben bisher nicht gelöst, weil ich mir noch nicht die nötige Zeit genommen habe.
D	Ich habe die Aufgaben bisher nicht gelöst, weil ich noch grundlegende Verständnisprobleme habe.

Quiz: 3 (Praktikum 1)

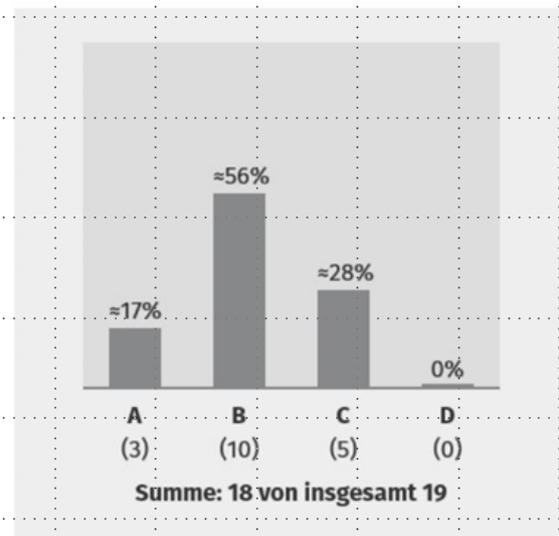


Abbildung 12: Umgang mit Forum

A	Ich habe beide Teile der ersten Praktikumsaufgabe im Wesentlichen gelöst.
B	Ich habe die Taschenrechner-Aufgabe im Wesentlichen gelöst.
C	Ich habe die Aufgaben bisher nicht gelöst, weil ich mir noch nicht die nötige Zeit genommen habe.
D	Ich habe die Aufgaben bisher nicht gelöst, weil ich noch grundlegende Verständnisprobleme habe.

Quiz: 3 (Praktikum 1)

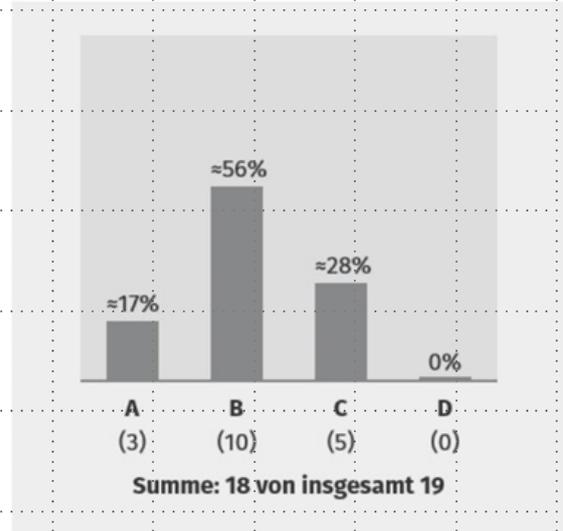


Abbildung 13: Arbeitsstand Praktikumsaufgabe

Kurz vor Ende des Semesters wurde die zweite Tweedbackbefragung durchgeführt. Die Ergebnisse sind in Abbildung 6 - 12 dargestellt.

Der OPAL-Kurs war gut strukturiert. (1)

A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu

Quiz: 1 (OPAL-Kurs)

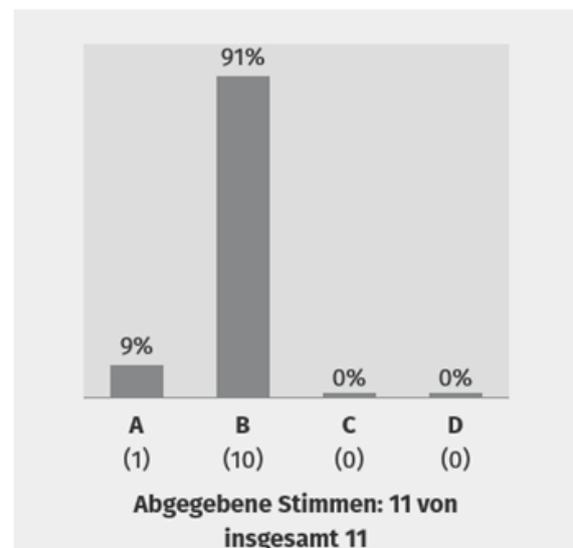


Abbildung 14: Strukturierung des Kurses

Ich empfand den Arbeitsaufwand für EvA als... (2)

Quiz: 2 (Arbeitsaufwand)



A	Viel zu hoch
B	Zu hoch
C	Angemessen
D	Zu Gering

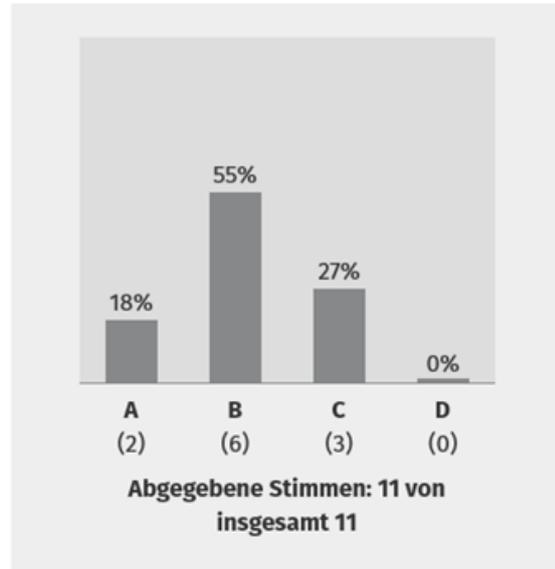


Abbildung 15: Arbeitsaufwand

Die ausgewählten Quellen waren für mich größtenteils verständlich. (3)

Quiz: 3 (Verständlichkeit Quellen)



A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu
E	Habe kaum was gelesen

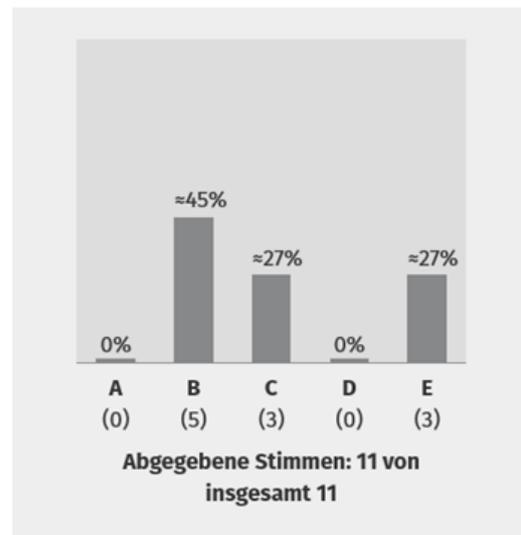


Abbildung 16: Quellenverständnis

Durch das Lesen der ausgewählten Quellen habe ich viel gelernt. (4)

Quiz: 4 (Durch Quellen gelernt?) 

A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu
E	Habe kaum was gelesen

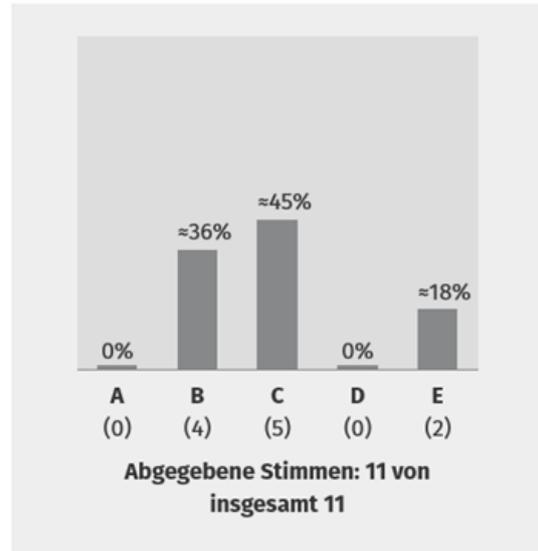


Abbildung 17: Lerneffekt Quellen

Die zur Verfügung gestellten Tutorialvideos habe ich als wertvolle Lernhilfe empfunden. (5)

Quiz: 5 (Tutorial-Videos) 

A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu

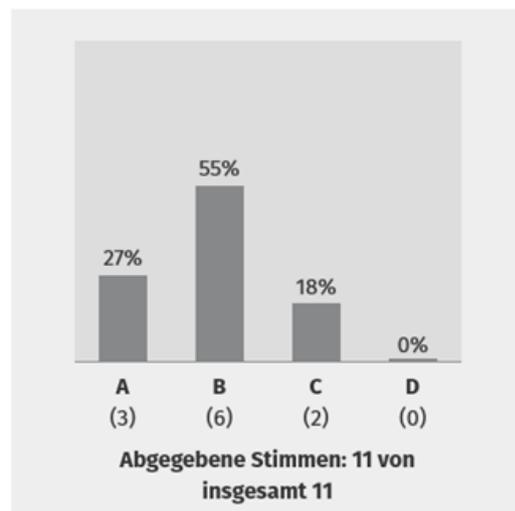


Abbildung 18: Einschätzung Tutorialvideos

Das Lehrtempo in EvA war für mich... (6)

Quiz: 6 (Lehrtempo)



A	Viel zu schnell
B	Eher zu schnell
C	Eher zu langsam
D	Viel zu langsam

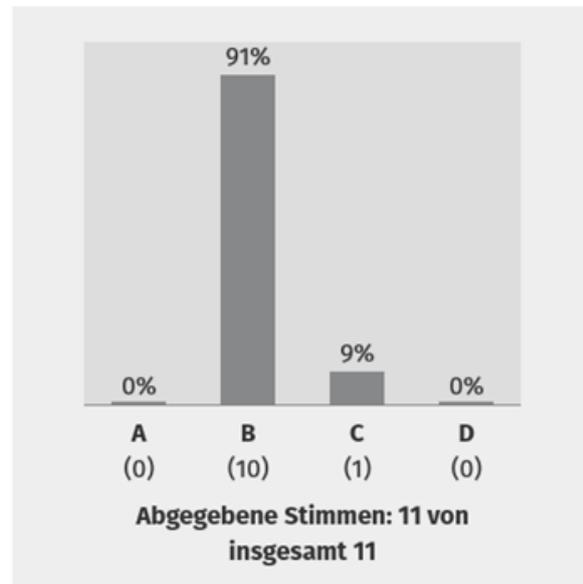


Abbildung 19: Lehrtempo

Rückblickend betrachtet war das EvA-Projekt für mich... (7)

Quiz: 7 (EvA-Projekt)



A	Viel zu schwer
B	Eher zu schwer
C	Angemessen
D	Eher zu leicht

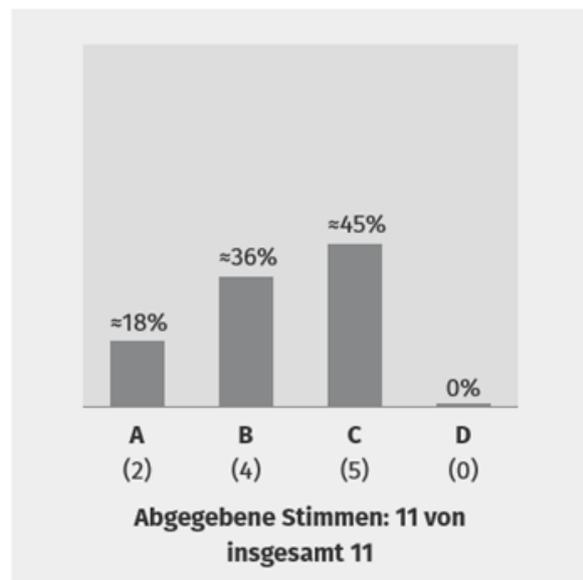


Abbildung 20: Schwierigkeitsgrad EvA-Projekt

Des Weiteren wurde durch den Projektmitarbeiter begleitend ein Tagebuch geschrieben, in dem im Wochenrhythmus Auffälligkeiten und Beobachtungen auf Seiten der Studierenden und Lehrenden notizartig festgehalten wurden:

Woche(n): 10.10 - 28.10.2016

Auffälligkeit auf Seiten von:	
→ Studierende	- Scheinen Texte nicht zu lesen
→ Lehrende	
→ MA für lehrbezogene Aufgaben	
Reaktion	- Durchführung Tweedback-Umfrage zur Nutzung der Texte [etwa 50%], des Forums [größtenteils nicht, weil noch nicht genügend Zeit für Fachaufgewandt] und der Erledigung der Praktikumsaufgaben [auch hier meist noch nicht genug Zeit investiert]
Sonstige Notizen	

Woche: 31.10. - 11.11.2016

Auffälligkeit auf Seiten von:	
→ Studierende	<ul style="list-style-type: none"> - besuchen freiwillig jede Woche das Praktikum (ca. 20 Studierende), obwohl nur 14tägig geplant in zwei Gruppen - ca. 5 eingeschriebene Studierende erscheinen gar nicht in LV - beginnen mit Bearbeitung Semesterprojekt - sind teilweise noch stark damit beschäftigt, die Entwicklungsinfrastruktur zum Laufen zu bekommen (vor allem Versionsverwaltung mit Git) - nutzen Forum nach wie nur sehr vereinzelt - sind vereinzelt unsicher über Prüfungsinhalte
→ Lehrende	- können aktuellen Kenntnisstand der Studierenden nur schlecht einschätzen
→ MA für lehrbezogene Aufgaben	- können aktuellen Kenntnisstand der Studierenden nur schlecht einschätzen
Reaktion	<ul style="list-style-type: none"> - Prüfungsinhalte sollen das nächste Mal thematisiert werden - nächste LV soll nur wenig neuen Stoff bringen
Sonstige Notizen	

Woche: 14.11 - 25.11.2016

Auffälligkeit auf Seiten von:	
→ Studierende	- stellen trotz Nachfrage kaum Fragen dazu, was

	<p>noch unklar ist</p> <ul style="list-style-type: none"> - von in dieser Woche vorgesehenen sechs vorgesehenen Studierendenteams geben nur zwei den für die PV nötigen Zwischenstand des Semesterprojekt ab
→ Lehrende	<ul style="list-style-type: none"> - etwas frustriert aufgrund mangelnder Reaktion in der LV - verunsichert, ob Fach für drittes Semester doch zu schwer - bemängeln schlechte Produktionsreife einiger verwendeter Technologien
→ MA für lehrbezogene Aufgaben	---
Reaktion	- weiter beobachten
Sonstige Notizen	<p>Lehrveranstaltung entfällt wegen Feiertag in der Woche vom 14.11.2016</p> <p>TAP-Evaluierung am 23.11.2016</p>

Woche: 28.11. - 9.12.2016

Auffälligkeit auf Seiten von:	
→ Studierende	<ul style="list-style-type: none"> - haben ihre Verbesserungsvorschläge über TAP geäußert (siehe Ergebnisse) - haben Ergebnisse des ersten Meilensteins eingereicht (einige ok, einige mit Nacharbeit) - nur noch 4 Studierende im Praktikum
→ Lehrende	<ul style="list-style-type: none"> - haben TAP-Ergebnisse mit Studierenden diskutiert - nehmen war, das größte Hürden jetzt langsam genommen sind - überlegen, ob zu viele Inhalte in LV
→ MA für lehrbezogene Aufgaben	
Reaktion	<ul style="list-style-type: none"> - Überlegungen, Git und IntelliJ bereits in GdP2 einzuführen - Vorlesung jetzt in anderem Raum (Steckdosen, WLAN) - Versuch, Live-Hacking demnächst mit IDE und Browser nebeneinander angeordnet (müsste auch durch neuen Raum möglich sein) - Teilgebiet Angular wird gestrichen und entsprechende Anpassung der Projektmeilensteine geplant
Sonstige Notizen	- TAP war sehr konstruktiv und hilfreich

Woche: 12.12. - 23.12.2016

Auffälligkeit auf Seiten von:	
--------------------------------------	--

→ Studierende	<ul style="list-style-type: none"> - entscheiden sich für den früheren Abgabetermin im Januar, um nochmals Feedback erhalten zu können - wieder 9 Teilnehmer im Praktikum
→ Lehrende	<ul style="list-style-type: none"> - streichen dritten Meilenstein des Semesterprojekts, neue Termine für Abgabe des zweiten Meilensteins mit Studierenden diskutiert - erarbeiten Liste mit Prüfungsfragen für bessere Definition der Prüfungsinhalte und geben diese an Studierende
→ MA für lehrbezogene Aufgaben	
Reaktion	
Sonstige Notizen	

Woche: 2.1. - 13.1.2016

Auffälligkeit auf Seiten von:	
→ Studierende	<ul style="list-style-type: none"> - arbeiten an Semesterprojekt - nehmen an 2. Tweedback teil - 11 Studierende in Vorlesung und Praktikum anwesend
→ Lehrende	<ul style="list-style-type: none"> - Abgabe Semesterprojekt um eine Woche verschoben
→ MA für lehrbezogene Aufgaben	
Reaktion	
Sonstige Notizen	<ul style="list-style-type: none"> - Tweedback durchgeführt, Ergebnisse passen zu bereits geführten Diskussionen

Woche: 16.1. - 27.1.2016

Auffälligkeit auf Seiten von:	
→ Studierende	<ul style="list-style-type: none"> - 12 Studierenden stellen abschließend ihre Projektarbeiten vor - eine Gruppe erhält noch letzte Frist für Nacharbeit - bei Konsultation 6 Studierende anwesend (regungslos, aber aufmerksam) - haben Schwierigkeiten, Konzepte zu verstehen → reine Anwendersicht, Framework blind verwendet
→ Lehrende	<ul style="list-style-type: none"> - schätzen abgegebene Arbeiten bis auf eine als recht gut ein - arbeiten nochmal stark mit Konzeptbildern → für nächstes Jahr fotografiert - dennoch relativ gutes Gefühl für Prüfung bei Großteil der Übriggebliebenen

→ MA für lehrbezogene Aufgaben	- bei Konsultation krank
Reaktion	- einzelne Fehler können noch bis zur Prüfung behoben werden für Bonus - Konsultationstermin für zweite Woche vereinbart - Fragen sollen im Forum gesammelt werden
Sonstige Notizen	

Woche: 30.1. - 10.2.2016

Auffälligkeit auf Seiten von:	
→ Studierende	
→ Lehrende	
→ MA für lehrbezogene Aufgaben	
Reaktion	
Sonstige Notizen	keine Veranstaltungen

Woche: 13.2. - 24.2.2016

Auffälligkeit auf Seiten von:	
→ Studierende	
→ Lehrende	- sind mit Prüfungsergebnissen im Wesentlichen zufrieden - Teilnehmer an der Prüfung: 12 - Durchgefallen: 2 - Durchschnittsnote: 2,47
→ MA für lehrbezogene Aufgaben	
Reaktion	
Sonstige Notizen	Prüfungen finden statt

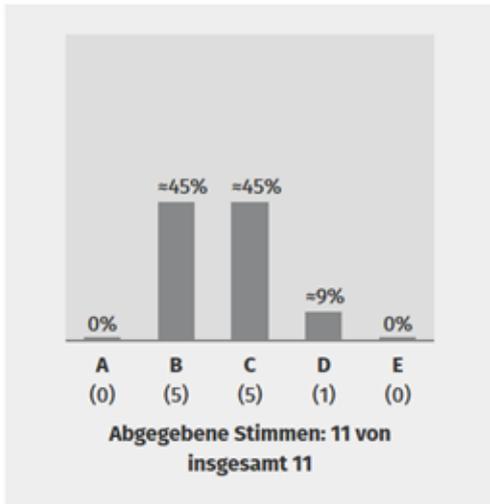
Befragung zum Semesterende:

Zum Semesterende wurde mit den Studierenden des Studiengangs eine zusammenfassende Befragung zu allen drei Modulen, Grundlagen der Programmierung I und II sowie Entwicklung verteilter Anwendungen, durchgeführt. So war es möglich noch einmal eine Rückmeldung zur Einschätzung der Studierenden zu allen drei Modulen zu erhalten. Der Zeitpunkt der Befragung war so gewählt, dass die Studierenden die Modulverläufe mit etwas zeitlichem Abstand betrachten konnten.

Fragen zu GdP I:

Ich habe rückblickend betrachtet das Greenfoot-Buch mit dem nötigen Aufwand gelesen. (1)

Quiz: 1

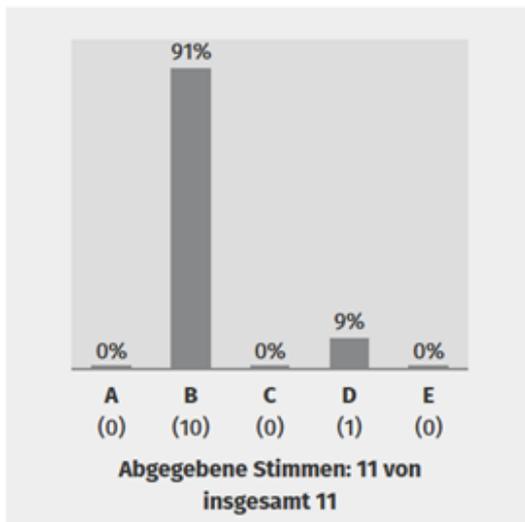


A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu
E	Keine Ahnung, da nicht viel gelesen

Abbildung 21: Leseaufwand Greenfoot-Buch

Das Greenfoot-Buch war ein gutes Lehrmittel. (2)

Quiz: 2 (Buch gut)



A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu
E	Keine Ahnung, da nicht viel gelesen

Abbildung 22: Bewertung Greenfoot-Buch als Lehrmittel

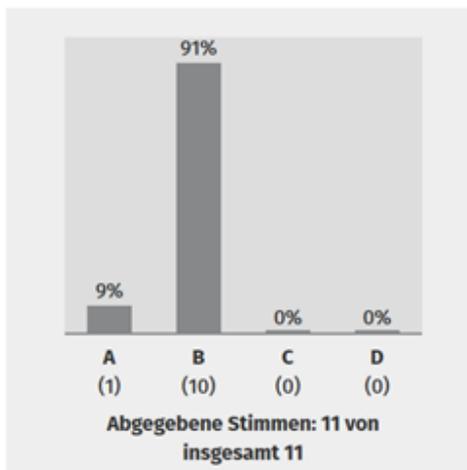
Offene Frage: Was war beim Greenfoot-Buch gut? Was eher hinderlich?

- Konzept der Objektorientierung bereits in der Schule vermittelt

- Vorkenntnisse vorhanden
- Für keine Vorkenntnisse bot das Buch gute Inhalte
- Hilfreich, dass Lösungen gegeben waren

Die Themenblöcke stellten für mich eine sinnvolle Ergänzung der im Lehrbuch dargestellten Konzepte dar. (4)

Quiz: 3 (Themenblöcke)

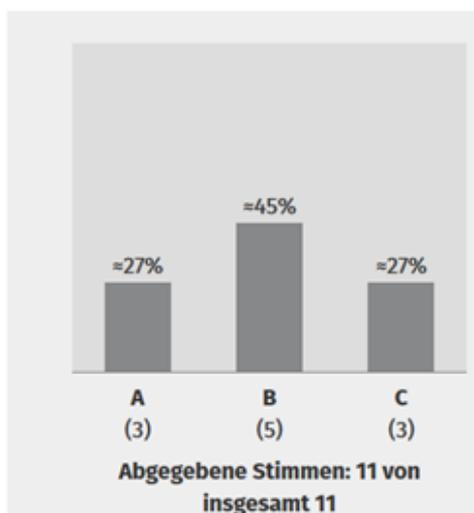


A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu

Abbildung 23: Bewertung Themenblöcke

Meine hochgeladenen Fragen in GdP1 waren hauptsächlich ... (5)

Quiz: 4 (GdP1 Fragen)



A	Echte Verständnisfragen
B	Alibifragen für die Prüfungsvorleistung
C	Mögliche Prüfungsfragen

Abbildung 24: Qualität hochgeladene Fragen

Vorschlag der Studierenden:

- Studierende mehr zum Anwenden des Wissens zwingen

Anmerkung der Studierenden:

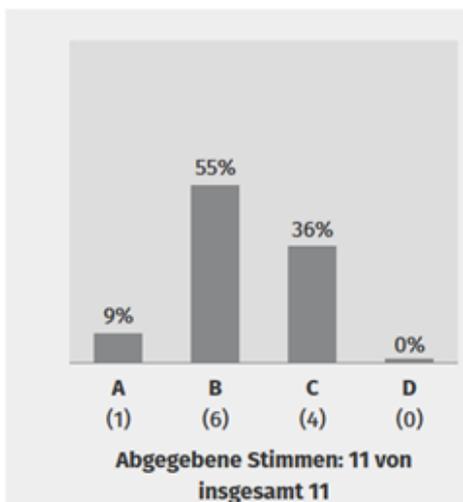
- Finden der Prüfungsfragen wurde als schwer empfunden.

Kommentar für die Verstetigung:

- Jetzt eingeplantes kontrolliertes Praktikum in GdP1 zwingt zum Anwenden des Wissens
- Wie lässt sich der hohe Anteil der Alibifragen weiter reduzieren? Wie könnte ggf. das Finden von Prüfungsfragen unterstützt werden?

Wie nützlich fanden Sie die Antworten auf die Fragen in GdP1? (6)

Quiz: 5 (Antworten nützlich)



A	Sehr nützlich
B	Nützlich
C	Ein wenig nützlich
D	Gar nicht nützlich

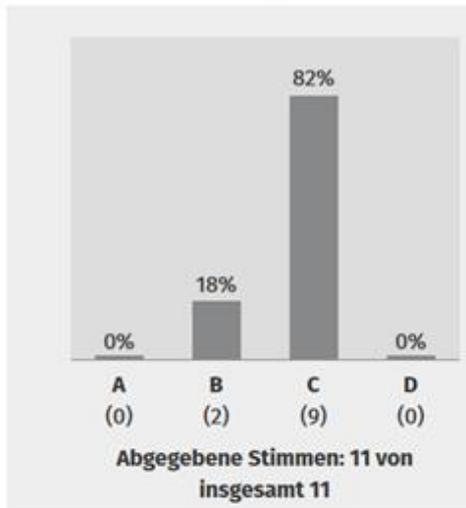
Abbildung 25: Bewertung der Fragenbeantwortung

Anmerkung der Studierenden:

- besser gestalten, indem Live-Demonstrationen zum Problem angeboten werden.

Ich empfand den Arbeitsaufwand für GdP1 als... (7)

Quiz: 6 (Arbeitsaufwand GdP1)

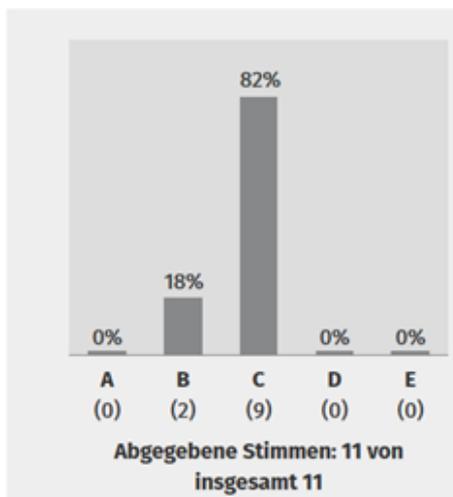


A	Viel zu hoch
B	Zu hoch
C	Angemessen
D	Zu gering

Abbildung 26: Arbeitsaufwand GdP 1

Ich empfand das Lehrtempo für GdP1 als... (8)

Quiz: 7 (Lehrtempo GdP1)



A	Viel zu schnell
B	Eher zu schnell
C	Angemessen
D	Eher langsam
E	Viel zu langsam

Abbildung 27: Lehrtempo GdP 1

Die Erwartungen an mich wurden zu Beginn von GdP1 klar kommuniziert. (9) Quiz: 8 (Erwartungen kommuniziert GdP1)

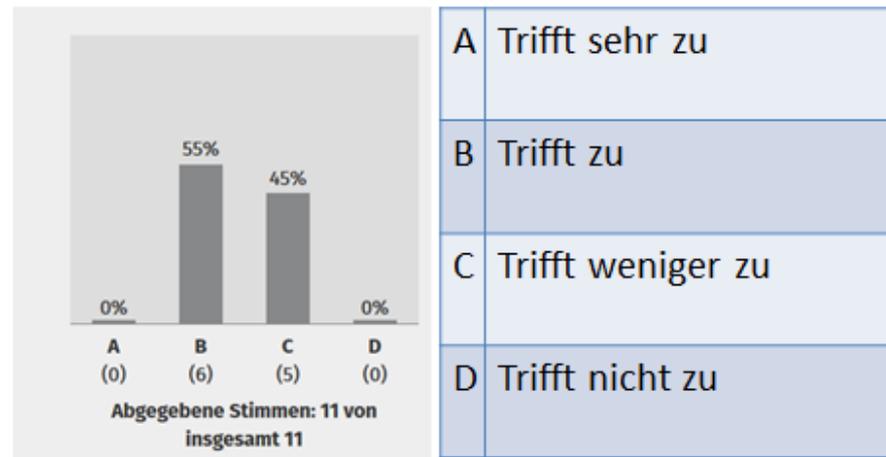


Abbildung 28: Kommunikation der Erwartungen

Mit welchem Ziel haben Sie das Modul GdP1 begonnen? (10)

Quiz: 9 (Ziel GdP1)

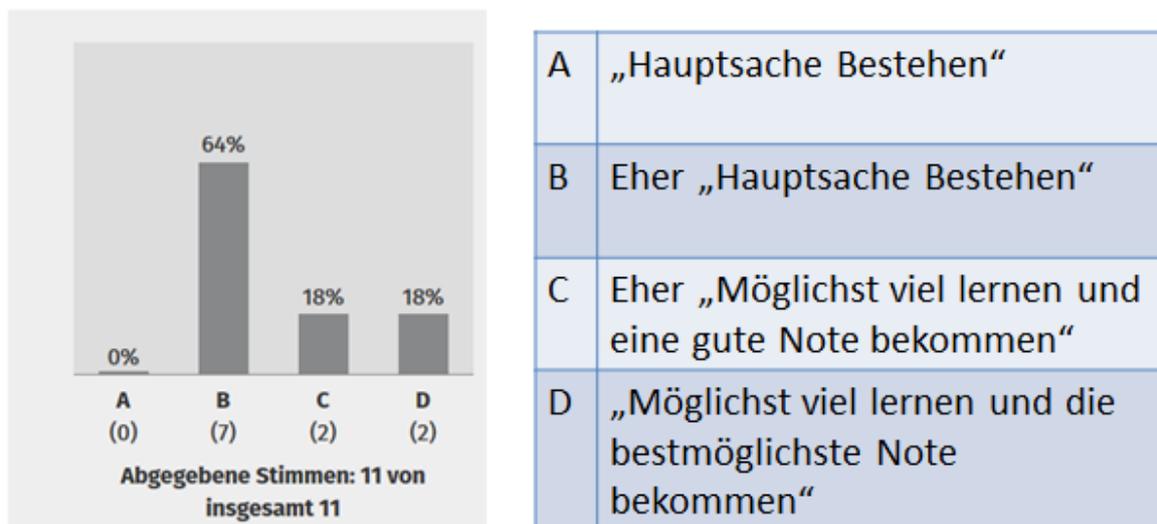
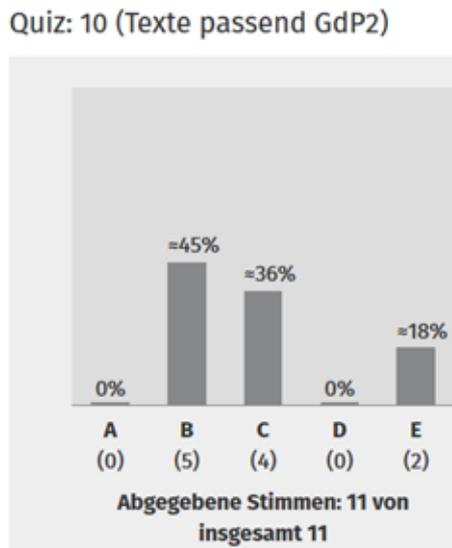


Abbildung 29: Ziel im Modul

Fragen zu GdP II

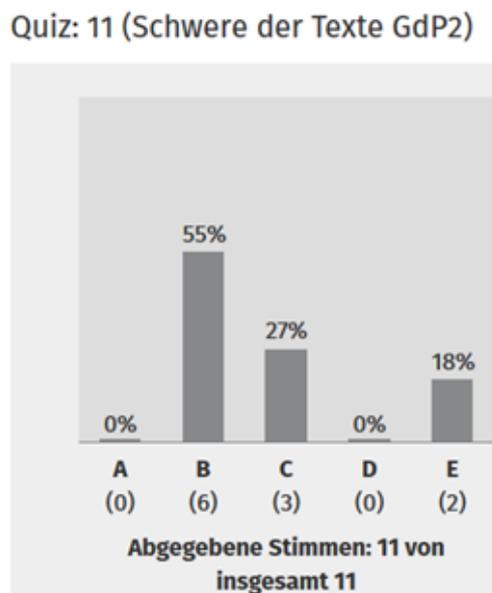
Wie inhaltlich passend waren die Texte in GdP2 ausgewählt? (11)



A	Sehr passend
B	Passend
C	Weniger passend
D	Gar nicht passend
E	Keine Ahnung, da nicht viel gelesen

Abbildung 30: Inhaltliche Passgenauigkeit Texte GdP 2

Wie schwer waren die in GdP2 ausgewählten Texte? (12)

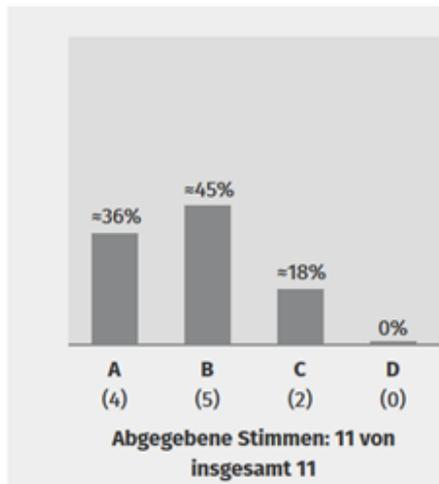


A	Sehr schwer
B	Eher schwer
C	Eher leicht
D	Sehr leicht
E	Keine Ahnung, da nicht viel gelesen

Abbildung 31: Schwierigkeitsgrad Texte GdP 2

Inzwischen halte ich es für sinnvoll, dass ein Teil der Quellen in GdP 2 in Englisch waren. (13)

Quiz: 12 (Quellen English GdP2)



A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu

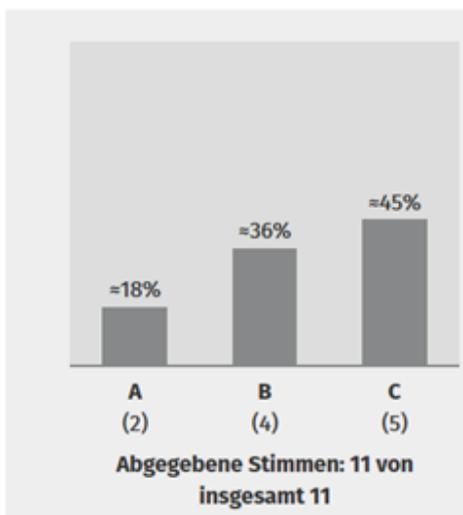
Abbildung 32: Sinnhaftigkeit englischer Quellen

Wunsch der Studierenden:

- Englisch konsistent für Themengebiete einsetzen. Kein Vermischen von Deutsch und Englisch im gleichen Themengebiet.

Meine hochgeladenen Fragen in GdP2 waren hauptsächlich ... (14)

Quiz: 13 (Fragen GdP2)

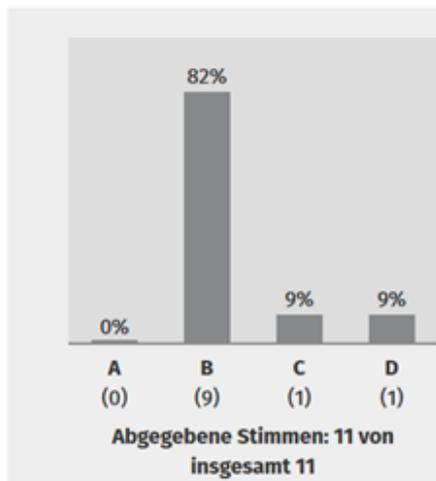


A	Echte Verständnisfragen
B	Alibifragen für die Prüfungsvorleistung
C	Mögliche Prüfungsfragen

Abbildung 33: Qualität Fragen GdP 2

Wie nützlich fanden Sie die Antworten auf die Fragen in GdP2? (15)

Quiz: 14 (Antworten GdP2)

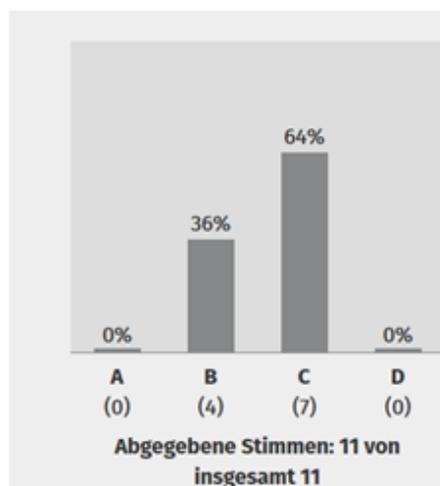


A	Sehr nützlich
B	Nützlich
C	Ein wenig nützlich
D	Gar nicht nützlich

Abbildung 34: Qualität Fragenbeantwortung GdP 2

Ich empfand den Arbeitsaufwand für GdP2 als... (16)

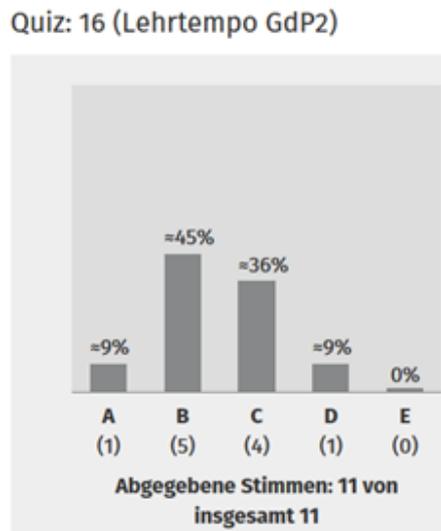
Quiz: 15 (Arbeitsaufwand GdP2)



A	Viel zu hoch
B	Zu hoch
C	Angemessen
D	Zu gering

Abbildung 35: Arbeitsaufwand GdP 2

Ich empfand das Lehrtempo für GdP2 als... (17)



A	Viel zu schnell
B	Eher zu schnell
C	Angemessen
D	Eher langsam
E	Viel zu langsam

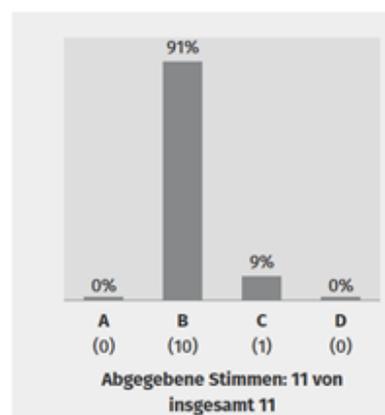
Abbildung 36: Lehrtempo GdP 2

Anmerkung der Studierenden:

- Das Durcharbeiten einzelner Themengebiete wurde als zu schnell empfunden (z. B. bei Behandlung von Lambdas).

Die Erwartungen an mich wurden zu Beginn von GdP2 klar kommuniziert. (18)

Quiz: 17 (Erwartungen kommuniziert GdP2)

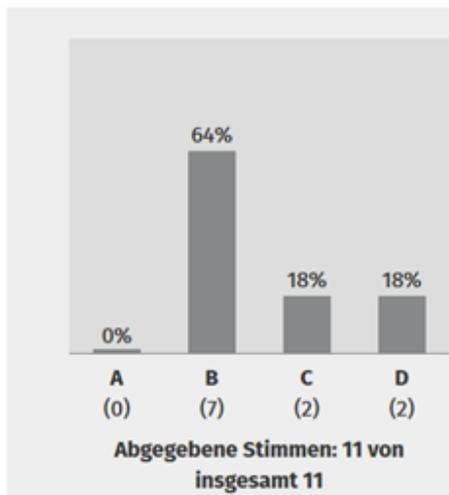


A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu

Abbildung 37: Kommunikation Erwartungen GdP 2

Mit welchem Ziel haben Sie das Modul GdP2 begonnen? (19)

Quiz: 18 (Ziel GdP2)



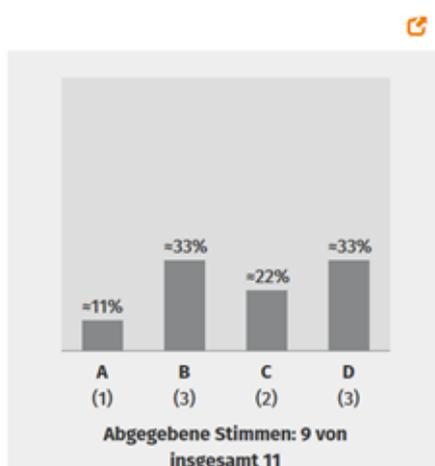
A	„Hauptsache Bestehen“
B	Eher „Hauptsache Bestehen“
C	Eher „Möglichst viel lernen und eine gute Note bekommen“
D	„Möglichst viel lernen und die bestmögliche Note bekommen“

Abbildung 38: Ziel im Modul GdP 2

Fragen zu Eva

Wie sind Sie zur Prüfungsvorbereitung mit den Quellen umgegangen? (20)

Quiz: 19 (Quellen Prüfungsvorbereitung EVA)

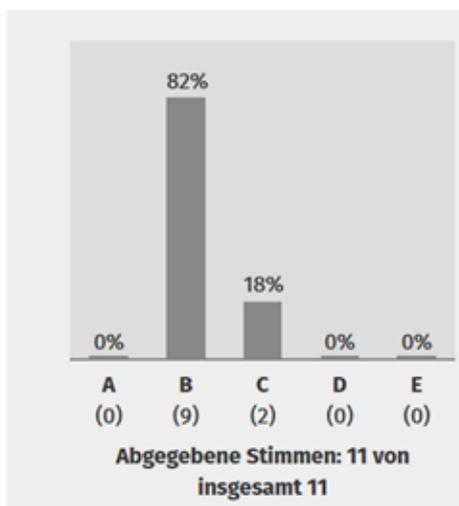


A	Alles noch einmal umfassend durchgearbeitet
B	Punktuell einzelne Themen durchgearbeitet
C	Noch einmal überflogen
D	Gar nicht noch einmal herangezogen

Abbildung 39: Umgang mit Quellen in Eva

So würde ich meine Anfangsphase in EvA beschreiben: (21)

Quiz: 20 (Anfangsphase EVA)



A	Von Anfang an intensiv mitgearbeitet
B	Von fachlichen Anforderungen überfordert
C	Von Werkzeugkette überfordert
D	Durch andere Fächer zu beschäftigt
E	Sonstige

Abbildung 40: Einschätzung Anfangsphase EvA

Anmerkungen der Studierenden:

- anfängliche Beispiele wurden als nicht gut empfunden (Taschenrechner)
- bei der „Studentenverwaltung“ fehlte das Wissen zu Datenbanken
- Das Einarbeiten in Thymeleaf wurde als schwierig empfunden
- Thymeleaf wurde als zu statisch empfunden
- Vorschlag: Angular benutzen
- Anmerkung: Folien von Prof. Grimm und Prof. Franke sind sehr ausführlich
- Wunsch: Schönerer Folien, bessere Darstellung der logischen Zusammenhänge auf Folien

Offene Frage:

Was müsste man beim Forum in EvA anders machen, damit es genutzt wird?

Hätte die Fragenabgabe in EvA wieder Prüfungsvorleistung sein sollen?

Anmerkungen der Studierenden:

- zu komplex
- Projekt wurde nicht wirklich verstanden
- Nähere Erklärungen zum Projekt am Anfang gewünscht
- Einführen der Prüfungsfragen in kleineren Schritten gewünscht
- Zeitigeres Einführen des Kataloges mit Prüfungsfragen zum schnelleren Verständnis gewünscht
- Fragen als Prüfungsvorleistung wurden als nicht sinnvoll erachtet
- Es wurde sich mehr Praktikumszeit für die Beantwortung der Fragen im Dialog gewünscht

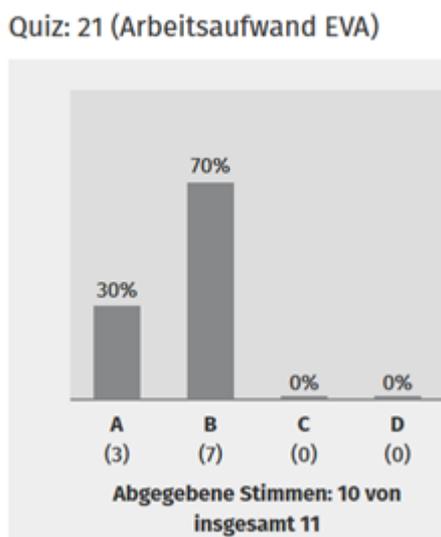
Offene Frage:

Was hätte bei dem EvA-Projekt bzw. Praktika anders laufen müssen?

Anmerkungen der Studierenden:

- Kleinere Teilabschnitte (z.B. Datenbank, Oberfläche) gewünscht
- Regelmäßigere Abstände gewünscht
- Themen entkoppelter üben

Ich empfand den Arbeitsaufwand für EvA als... (24)

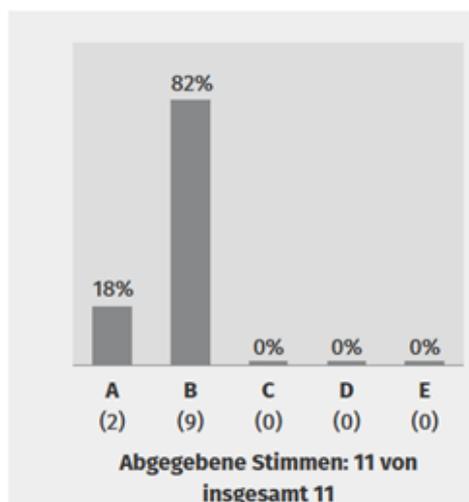


A	Viel zu hoch
B	Zu hoch
C	Angemessen
D	Zu gering

Abbildung 41: Arbeitsaufwand EvA

Ich empfand das Lehrtempo für EvA als... (25)

Quiz: 23 (Lehrtempo EVA)

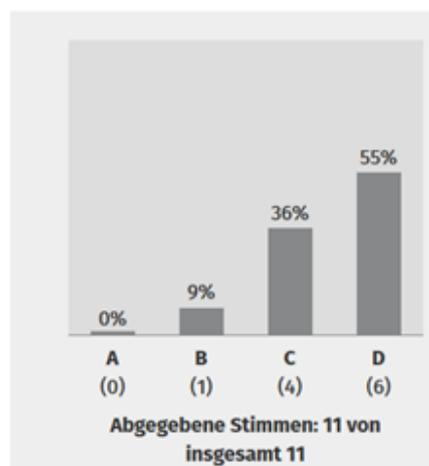


A	Viel zu schnell
B	Eher zu schnell
C	Angemessen
D	Eher langsam
E	Viel zu langsam

Abbildung 42: Lehrtempo EvA

Die Erwartungen an mich wurden zu Beginn von EvA klar kommuniziert. (26)

Quiz: 24 (Erwartungen EVA)

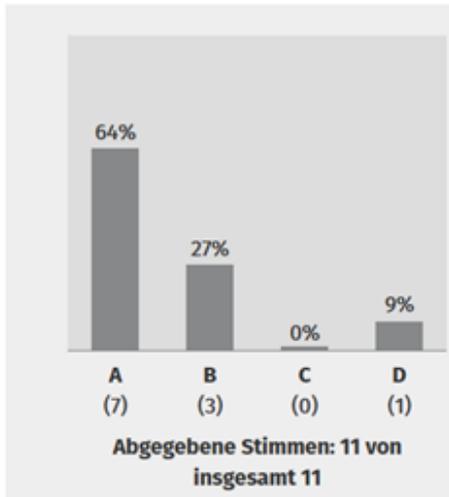


A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu

Abbildung 43: Kommunikation Erwartungen EvA

Mit welchem Ziel haben Sie das Modul EvA begonnen? (27)

Quiz: 25 (Ziel EVA)



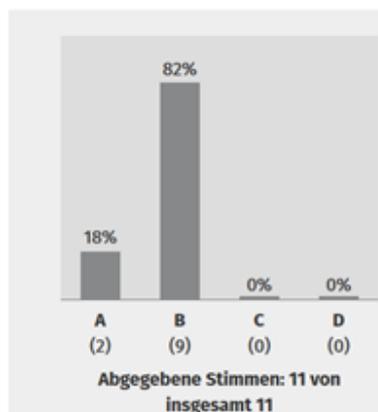
A	„Hauptsache Bestehen“
B	Eher „Hauptsache Bestehen“
C	Eher „Möglichst viel lernen und eine gute Note bekommen“
D	„Möglichst viel lernen und die bestmögliche Note bekommen“

Abbildung 44: Ziel im Modul EvA

Gesamteinschätzung der Studierenden zu allen drei Modulen:

Die Organisation der drei Module hat es mir ermöglicht, sehr selbstständig zu bestimmen, wann und wo ich lerne. (28)

Quiz: 26 (Organisation 3 Module selbständiges Lernen)



A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu

Abbildung 45: Gesamteinschätzung GdP 1 und 2 sowie EvA

Fazit

Ein Großteil der Studierenden, die das Modul erfolgreich absolviert haben, haben recht gute und anwendungsbereite Kenntnisse in einem komplexen Themengebiet erworben. Die Lehrvideos des Dozenten sind sehr gut angenommen worden und stellen eine sehr gute Ergänzung der Präsenzveranstaltungen dar. Damit wurden auch praktische Fähigkeiten im Umgang mit den Werkzeugen und Frameworks gut vermittelt.

Jedoch wurde die Fragenabgabe über das Forum kaum angenommen. Damit ist der geplante Flipped-Classroom-Ansatz nicht in ursprünglich geplanter Form möglich gewesen. Durch die Beobachtung des Lernfortschritts in der Präsenzzeit konnten aber Themen selektiert werden, die in gemeinsam durchgeführten Mitmachaufgaben umgesetzt wurden.

Bedingt durch die Probleme in der Werkzeughandhabung sind über die Anwendung hinausgehende Konzepte von Studierenden erst sehr spät im Semester erkannt und verstanden worden. Daher wird bei einer Wiederholung des Konzepts früher auf Konzepte eingegangen und über Grafiken Zusammenhänge verdeutlicht sowie intensiver auf die Arbeit mit den Werkzeugen eingegangen.

Weiterhin hat sich gezeigt, dass die Projektaufgabe genau gestreckt werden muss und um eine Teilaufgabe reduziert werden muss. Dieses Thema wird beim nächsten Durchlauf auch nicht berücksichtigt, um mehr Zeit für die Vertiefung der restlichen Themen und Konzepte zu haben.

Bisher wurde mit Online-Quellen im Semesterplan der Studierenden gearbeitet. Im Sinne einer Verstetigung gibt es jedoch nun konkrete Überlegungen zum Schreiben eines eigenen Buches als Grundlage der künftigen Veranstaltungsdurchläufe.

Die Lehrenden schätzten ebenfalls ein, dass die regelmäßig an der Lehrveranstaltung teilnehmenden Studierenden großen Einsatz gezeigt haben. Der klassische Flipped Classroom Ansatz ist modifiziert worden. Durch Beobachtung der Lernfortschritte in der Präsenzzeit sind Themen selektiert worden, die in Mitmachaufgaben behandelt worden. Dies führte zu lebhaften Diskussionen. Bedingt durch Probleme bei der Werkzeughandhabung ist der Einstieg nicht optimal gelaufen. Beim nächsten Mal wird intensiver auf die Arbeit mit Werkzeugen für Code Repositories und auf die Entwicklungsumgebung eingegangen.

1.5 Netzwerke

Rahmendaten des Moduls

Modulname	Netzwerke
Modulnummer	PTI 648
Lehrende des Moduls	Prof. Dr. Frank Grimm
Zuordnung zum Curriculum: Studierende welcher Studiengänge nehmen an den Modulen teil und in welchem Semester?	Informatik (B.Sc.)
Workload	120 h
ECTS-Credits	4
Veranschlagte Präsenzzeiten	45 h
Veranschlagte Selbststudienzeiten	75 h
Prüfungsleistung, in welcher Form?	mündlich
Beschreibung der Lernziele	Die Studierenden haben Fähigkeiten im Umgang mit Rechnernetzen und der Einschätzung der Sicherheit in Rechnernetzen. Die Studierenden lernen die prinzipielle Struktur von heterogenen Kommunikationssystemen kennen. Sie sind fähig, Infrastrukturen und Realisierungsvarianten zu analysieren und zu bewerten. Sie haben die theoretische Kompetenz und Realisierungskompetenz, Netzwerke unter TCP/IP einzurichten, zu administrieren und zu optimieren. Es werden die theoretischen Voraussetzungen für eine CISCO Zertifizierung gelegt
Zeitpunkt der Erprobung	SoSe 2017

Inhalt und Methodik

Die Inhalte des Moduls sehen laut Modulbeschreibung folgende Gliederung vor:

- Ziele und Beispiele der Vernetzung
Einordnung von Computernetzen in Kommunikationssysteme
Historische Entwicklung
Sichtweisen auf ein Computernetz
- Die geographische Sicht und die Anwendersicht auf ein Netz
LAN, WAN, MAN: Abgrenzung und Besonderheiten
Zugangsnetze

Die Anwendersicht (Bedeutung von Client und Server in einem Netz) Varianten der Client-Server-Architektur

- Grundlagen der technischen Datenübertragung und Topologien von Rechnernetzen Eigenschaften eines Übertragungskanal (Basisband, Breitband) Topologien, das Verhältnis logischer und physischer Topologien
- Schichtenmodelle und Standardisierung Bedeutung, Beschreibung, Anwendung, Begriffe Das OSI- Schichtenmodell, Erweiterung des Modells (IEEE 802), reale Schichtenmodelle
- Der TCP/IP- Protokollstack Protokollarchitektur Anwendungsschicht mit Beispiel SMTP, POP3, IMAP, http Transportschicht TCP und UDP (Kommunikationsablauf, Portbegriff, Flusssteuerung, verbindungsorientierte und verbindungslose Kommunikation) IP-Schicht (Header und Kommunikationsablauf, IP-Adressierung (IPv4), Subnettierung, ICMP, ARP, RARP, MAC-Adressen, RFCs)
- Ethernet
- Wireless Networking
- Switching und Routing
- Sicherheit im Netz unter dem TCP/IP-Protokoll
- Firewalls, IPSec

Kursaufbau

[Bearbeiten](#)

Einführung in



im Rahmen der Lehrveranstaltungen der Module **PTI648 Netzwerke** und **PTI622 Kommunikationssysteme** des Studienganges Informatik, B.Sc. an der Westsächsischen Hochschule Zwickau



Die Erstellung dieser Lerninhalte wird aus Mitteln des BMBF-Projektes "Offene Hochschule Zwickau" (Förderkennzeichen 16OH12018) gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

veröffentlicht unter der Creative Commons Attribution 3.0 License



Abbildung 46: Startseite des OPAL-Kurses im Modul "Netzwerke"

Für dieses Modul wurde ein Themengebiet komplett auf der Lernplattform OPAL als Selbstlernmaterial aufbereitet. Die Studierenden erhalten in dem entstandenen OPAL-Kurs zuerst eine theoretische Einführung in das Thema und anschließend werden die Kenntnisse mit Hilfe von praktischen Übungen vertieft.

The image shows a hierarchical table of contents for a course titled 'Einführung in OpenStack'. The structure is as follows:

- Einführung in OpenStack
 - Hinweise (Etherpad, Accounts)
 - Einschreibung
 - Theoretische Grundlagen
 - Overlay-Netze
 - Open vSwitch
 - OpenStack
 - Was ist OpenStack?
 - Komponenten
 - Funktionsweise
 - Installation
 - Praktische Übung I
 - Motivation
 - Arbeitsvorbereitung
 - Aufgabenstellung
 - Praktische Übung II
 - Motivation
 - Arbeitsvorbereitung
 - Aufgabenstellung
 - Zusammenfassung
 - Literatur
 - Weblinks
 - Feedback
- Gruppen

Abbildung 47: Gliederung des OPAL-Kurses im Modul "Netzwerke"

Für jedes der Themengebiete wurde eine Startseite angelegt, die den Studierenden einen Überblick über das Kapitel gibt. Dafür wurden Art des Inhalts, notwendige Werkzeuge, Zielstellung des Kapitels und notwendige Kenntnisse angegeben. Damit die Studierenden den Zeitaufwand besser einschätzen können, wurde ebenfalls eine Bearbeitungszeit angegeben.

Übersicht - Theoretische Grundlagen Kurz und gut

Art des Inhalts Text- und Grafikinhalte	Umfang 60 Minuten
Notwendige Werkzeuge Webbrowser	Notwendige Kenntnisse und Fähigkeiten Grundkenntnisse in Netzwerktechnik und Cloud Computing
Zielstellung Erwerb von Kenntnissen zu grundlegenden Konzepten und Technologien im Bereich Cloud Computing: <ul style="list-style-type: none">• Sie kennen die Software Open vSwitch und deren Funktionsweise.• Sie können die Bedeutung von Open vSwitch im Kontext von Cloud Computing einordnen.• Sie haben Kenntnis über die Gestaltung von Netzwerken mit dem Overlay/Underlay-Ansatz.• Sie können den Aufbau und die Wirkungsweise von Overlay/Underlay-Netzen beschreiben.	

Abbildung 48: Beispiel für die Startseite

Die theoretischen Grundlagen wurden dann als Fachtexte präsentiert:

Overlay-Netze Mit Netz und doppeltem Boden

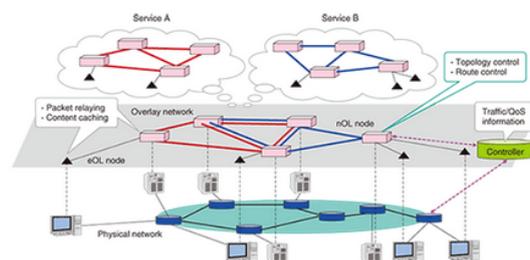
Begriffsklärung

Netzwerkcomponenten wie Switches bestehen aus mehreren Teilen. Die sogenannte *Data Plane* ist dafür verantwortlich, Pakete von einem Switch-Port zu einem anderen zu transportieren. Die Managementaufgaben werden von der *Control Plane* übernommen. Die Idee ist es nun, diese beiden Komponenten voneinander zu trennen.

Um dieses Ziel zu erreichen, teilen gängige Lösungen das Netz in zwei Schichten auf: das *Underlay* und das *Overlay*. Das Underlay umfasst alle Hosts (auch VMs), die innerhalb eines SDN-Setups einen Dienst betreiben. Im Hintergrund sind die Hosts mittels einer Tunnel-Technologie (GRE oder VXLAN) zu einem logischen Netzwerk verbunden.

Basierend auf dem Underlay sorgt das Overlay dafür, dass die Trennung zwischen den Kunden funktioniert, das Pakete den Weg von einer VM zu einer anderen VM (über virtuelle Bridges; auch auf einem anderen Hypervisor) finden und das Dienste wie DHCP in den VMs verfügbar sind. Da SDN-Lösungen das Overlay komplett selbst verwalten, sind sehr individuelle Konfigurationen möglich. Die physischen Switches sind in solchen Setups zu bloßen Data Planes degradiert, die Control Plane lebt im Overlay der SDN-Lösung. Damit entsteht ein eigener Adressraum mit einer unabhängigen Adressierung.

Overlay-Netze definieren eine logische Topologie auf der zugrunde liegenden physikalischen Topologie. Knoten, die sich im Overlay-Netz befinden, werden *Peers* genannt. Sie stellen in einem Peer-to-Peer Netz gleichberechtigte und autonome Einheiten dar. Wenn im Overlay eine direkte logische Verbindung zwischen zwei Peers besteht, dann entspricht diese im zugrunde liegenden physikalischen Netzwerk auch mindestens einem Hop, meist aber mehreren Hops. Einzelne Peers können in Gruppen organisiert werden, um gemeinsam Arbeit zu verrichten, Dienste zu nutzen bzw. zur Verfügung stellen. Peer-Gruppen werden durch eine ID identifiziert und ein Peer kann Mitglied in mehreren Gruppen sein.



Quelle: NTT Review (<https://www.ntt-review.jp>)

Mit der neu entstandenen Struktur lassen sich aktuelle Technologien wie IPv6 oder alternative Routen für Daten testen. Beispielsweise sind verteilte Systeme wie Peer-to-Peer-Netzwerke und Client-Server-Anwendungen prinzipiell Overlay-Netzwerke, da ihre Knoten (und die Datenpakete) über das Internet laufen. Das Internet wurde ursprünglich als Overlay auf dem Telefonnetzwerk aufgebaut, während heute (durch das Aufkommen von VoIP) das Telefonnetz zunehmend zu einem Overlay-Netzwerk wird, das über dem Internet aufgebaut ist. Weitere Beispiele für weitverbreitete Overlay-Netze sind u.a. Tor, CAN oder Gnutella.

Resilient Overlay Networks (RON) sind Architekturen, die es verteilten Anwendungen ermöglichen, Ausfälle oder Interferenzen zu erkennen und zu korrigieren. Die RON-Knoten überwachen die Internetpfade und bestimmen, ob Pakete direkt über das Internet oder über andere RON-Knoten umgeleitet werden sollen. Diese Knoten bilden ein Anwendungsschicht-Overlay, das bei Routing-Paketen zum Einsatz kommt. Jeder der RON-Knoten überwacht die Qualität der Internet- und nutzt diese Informationen, um die optimale Route für ein Paket auszuwählen, wodurch die benötigte Zeit für die Wiederherstellung bei schlechter Dienstqualität reduziert wird.

Abbildung 49: Beispiel 1 für die Darstellung der theoretischen Grundlagen

Die Texte wurden durch Erklärvideos ergänzt.

Was ist OpenStack? Cloud Computing im Eigenbau

Begriffsklärung

OpenStack ist eine Software zum Betrieb eines Cloud-Systems auf eigener Hardware. Statt mit physischen Servern, Netzwerken und Storage-Systemen, arbeitet man mit virtuellen Maschinen, Netzen und Speichern. Im Zentrum dieser *Infrastructure-as-a-Service-Plattform* steht dabei der Self-Service-Gedanke: Der Betreiber einer OpenStack-Cloud stellt jedem Mandanten (z.B. einem Kunden oder einer Fachabteilung) einen Pool an Ressourcen (Rechenleistung, Hauptspeicher, Festplattenspeicher, Netzwerk) zur Verfügung. Mit diesen Ressourcen können eigene VMs konfiguriert, gestartet und vernetzt werden.

Wenn man von OpenStack spricht, denkt man intuitiv zuerst an eine (große, schwergewichtige) Software. Allerdings ist OpenStack eher eine Sammlung bzw. ein modulares Ökosystem von verschiedenen Komponenten, die im Hintergrund miteinander kommunizieren. Die Software selbst ist (fast) vollständig in Python implementiert.

Dabei seien zwei Dienste erwähnt, die in praktisch jeder produktiven Open-Stack-Cloud auftauchen: **RabbitMQ** und **MySQL**. Die Open-Stack-Komponenten nutzen MySQL, um Metadaten persistent zu speichern. Dies sind u.a. Laufzeitdaten, die durch Benutzerinteraktion oder durch das Eingreifen des Admins entstehen (z.B. Nutzer anlegen oder VMs konfigurieren). RabbitMQ dient als **RPC**-Dienst im Hintergrund dazu, dass die einzelnen Dienste über Komponentengrenzen hinweg, miteinander effizient kommunizieren können.



Quelle: <https://stocksnap.io/photo/UU7RH2LO7P>

Hintergrund und Motivation

In den letzten Jahren hat sich OpenStack technisch und bzgl. seiner Infrastruktur stark weiterentwickelt. Eine Vielzahl von großen Unternehmen (Cisco, IBM, Intel, VMware, Red Hat, Walmart, Disney, ...) setzen mittlerweile auf OpenStack. Diese Entwicklungen lassen sich auf der halbjährlichen Konferenz **OpenStack Summit** sehr schön nachverfolgen.

Das Argument, das für OpenStack bzw. Cloud Computing spricht, ist, dass viele Konzepte aus den konventionellen Netzwerken (z.B. für Hardware, Netzwerklast, Stromversorgung oder Softwareeinsatz), aufgrund der Anwendungsfälle nicht mehr greifen können. Mit OpenStack möchte man diese Herausforderungen mit möglichst wenig Mehraufwand lösen.

Das folgende Konferenz-Video aus dem Jahr 2012 gibt einen guten Einblick in die Idee, welche hinter OpenStack steckt:



Quelle: <https://www.youtube.com/embed/bCsw2kklWyyw>

Ein wichtiger Aspekt beim Cloud-Computing ist das Thema **Selbstständigkeit**. Kunden bzw. Anwender wollen sich per **Webinterface** eine neue VM zusammenklicken und diese nach wenigen Sekunden verwenden. Wichtige Faktoren, wie die Netzwerk-Topologie, sollen dabei auch konfigurierbar sein, d.h. das Netzwerk soll direkt aus der Cloud-Umgebung heraus konfigurierbar sein. Auch die Trennung von Kunden-Traffic auf Hardware-Ebene spielt dabei eine entscheidende Rolle. Kunden sollen und dürfen die Datenpakete anderer Kunden nicht sehen. Trotzdem muss die Cloud dabei so flexibel bleiben, dass beispielsweise eine VM von einem Host auf den Anderen umziehen kann, ohne dass der Netzwerkverkehr beeinträchtigt wird.

Abbildung 50: Beispiel 2 für die Darstellung der theoretischen Grundlagen

Für die praktischen Übungen wurde ebenfalls jeweils eine Startseite angelegt, die den Studierenden einen Überblick über die Übung gibt. Dafür wurden Art des Inhalts, notwendige Werkzeuge, Zielstellung und notwendige Kenntnisse angegeben. Damit die Studierenden den Zeitaufwand besser einschätzen können, wurde ebenfalls eine Bearbeitungszeit angegeben.

Übersicht - Praktische Übung I Kurz und Gut

Art des Inhalts Text- und Grafikinhalte	Umfang 300 Minuten
Notwendige Werkzeuge Texteditor, Python, Git, Bash, Webbrowser, OpenStack	Notwendige Kenntnisse und Fähigkeiten Grundlagen in OpenStack, Programmierung und Netzwerksicherheit
Zielstellung Erwerb von Fähigkeiten zur Softwareverteilung mit OpenStack: <ul style="list-style-type: none">• Sie können eine Arbeits- und Entwicklungsumgebung vorbereiten.• Sie kennen die Grundoperationen von OpenStack.• Sie beherrschen die Schnittstellenkommunikation mit der OpenStack-API.• Sie kennen den Aufbau eines Installations- und Verteilungsskriptes.• Sie beherrschen den Workflow zum Aufsetzen einer funktionsfähigen Testanwendung in einer OpenStack-Umgebung.	

Abbildung 51: Beispiel Startseite für die Übungen

Weiterhin wurde zur Motivation der Sinn der Übung erläutert sowie die eine Anleitung für die notwendige Arbeitsvorbereitung gegeben.

Motivation Sinn und Zweck der Übung

Softwareeinsatz in der Cloud

Durch die Art und Weise wie mit OpenStack Ressourcen flexibel konfiguriert und in Form von virtuellen Maschinen genutzt werden können, liegt es nahe, dieses Konzept auch im Bereich der Softwareentwicklung zu nutzen.

Der Schritt der Anwendungsentwicklung bzw. -verteilung aus einem traditionellen IT-Umfeld hin zu einem cloud-basierten, verteilten System stellt eine oftmals große Hürde dar. Aber genau hier kann OpenStack als Brücke dienen.

Mit Hilfe von OpenStack SDKs ist es möglich, skalierbare Anwendungen zu bauen, die nicht nur in der Cloud ("auf einem Server liegend") sondern durch und gerade wegen der Cloud und ihrer Eigenschaften laufen und sich dadurch die Vorteile von Cloud Computing ergeben.

Die in dieser Übung vorgestellte Anwendung ist ein Fraktal-Generator, der unter Verwendung von mathematischen Gleichungen, Vorschriften und Berechnungen die namensgebenden Fraktale und passenden Grafiken erzeugt. Mit den folgenden Arbeitsschritten werden die notwendigen API-Methoden demonstriert, um eine skalierbare Anwendung in die Cloud zu packen.

UUID	12c13a6b-6c71-410b-ade0-71a8ccc205d1
Duration	5,2968 seconds
Dimensions	676 x 600 px
Refreshes	376
Parameters	ix = -1,09716 iy = 2,49134 ix = -1,44218 iy = 2,01111
Filesize	65261 bytes
Checksum	6097f4d57022a9f442f0386a12af7a2c178432a7584f799451249

UUID	8b743eb-71a1-4201-af3c-3633958ebc84
Duration	3,0514 seconds
Dimensions	825 x 556 px
Refreshes	247
Parameters	ix = -1,77824 iy = 1,11851 ix = -1,20958 iy = 2,28769

Quelle: <https://developer.openstack.org/firstapp-libcloud/>

Das folgende Video zeigt einen Initialvortrag der Autoren auf der OpenStack-Summit 2015 des hier zugrundeliegenden Tutorials. Sie beschreiben, was das Konzept hinter ihrem Leitfaden ist und wie man auf die Idee kommt, eine Anleitung für den Einsatz von Software in Verbindung mit OpenStack zu schreiben.

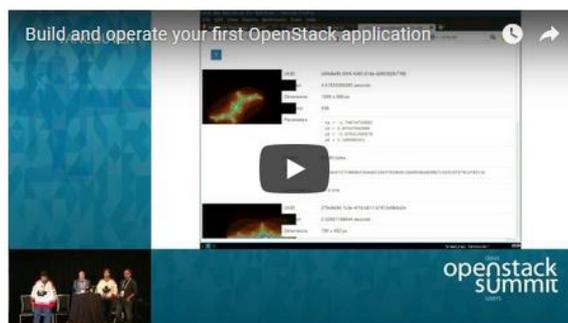


Abbildung 52: Beispiel Motivationsseite

Arbeitsvorbereitung Vorbereitung ist alles

Was wird benötigt?

Für die kommende Übung wird eine [lauffähige OpenStack-Installation](#) benötigt. Zudem benötigt man innerhalb der OpenStack-Umgebung ein Projekt (im Fachjargon auch *Tenant* genannt) mit [mindestens sechs Instanzen](#). Auf diesen Instanzen muss entweder eine *Ubuntu*-, *Debian*-, *Fedora*- oder *openSUSE*-Linux-Distribution als Betriebssystem verwendet werden. Der Hintergrund ist, dass die Fraktal-Anwendung auf diese Betriebssystemvarianten abgestimmt ist und nur damit die Lauffähigkeit auch gewährleistet werden kann.

Eine vorbereitete OpenStack-Installation ist unter der Adresse: <http://141.32.28.112/> bzw. <http://141.32.28.112/dashboard/> erreichbar.

Des Weiteren empfiehlt es sich, für die späteren Konfigurations- und Programmieraufgaben einen komfortablen Texteditor wie z.B. [Visual Studio Code](#) zu verwenden, der z.B. Syntax-Highlighting oder Code-Vervollständigung unterstützt.

Notwendige Umgebungssoftware

Installation eines OpenStack-SDKs

Für die programatische Kommunikation mit der OpenStack-Cloud ist ein entsprechendes [SDK](#) notwendig. Dieses muss [auf dem lokalen Entwicklungsrechner](#) installiert werden, damit die API bzw. die Kommandos der Komponenten verwendet werden können. Für OpenStack existieren zahlreiche SDK-Varianten in Abhängigkeit von Programmiersprachen und zusätzlichen Technologien. In dieser Übung wird mit dem python-basierten SDK [Libcloud](#) gearbeitet. Da es für die späteren Aufgabenstellungen sinnvoll und hilfreich ist, mit der Syntax von [Python](#) vertraut zu sein, findet sich unter diesem [Link](#) ein Syntax-Cheatsheet für Python als Hilfestellung.

Zur Installation von Libcloud muss folgender Python-Befehl auf der Kommandozeile eingegeben werden:

```
$ sudo pip install apache-libcloud
```

Libcloud wurde von der Apache Foundation entwickelt und unterstützt neben OpenStack auch andere Cloud-Computing-Umgebungen. Eine vollständige Liste anderer möglicher SDKs findet sich [hier](#).

Installation der Interaktionsbibliothek

Um direkt mit der Cloud interagieren zu können, muss eine weitere Python-Bibliothek verwendet werden. Diese Bibliothek trägt den Namen [shade](#) und muss mit folgendem Befehl auf der Kommandozeile installiert werden:

```
$ sudo pip install shade
```

Dabei sollte darauf geachtet werden, dass stets eine möglichst aktuelle Version dieser Bibliothek verwendet wird.

Notwendige Umgebungsinformationen

Zur Parametrierung der Kommunikation werden bestimmte Informationen aus der Cloud-Umgebung verwendet. Folgende Parameter werden für die spätere Programmierung benötigt und müssen im Vorfeld recherchiert werden:

- `auth_url` (URL bzw. IP-Adresse der OpenStack-Instanz)
- `username` (Benutzername zur Authentifizierung)
- `password` (Passwort des Benutzers)
- `project_id` oder `project_name` (eindeutige Kennung des Projekts/Tenants)
- `region_name` (Lokalisierungsparameter der OpenStack-Instanz)

Entweder erfragt man diese Daten beim Administrator der OpenStack-Umgebung oder man nutzt das [Horizon-Dashboard](#), um an diese Informationen zu kommen. Nachdem man sich unter <http://141.32.28.112/dashboard/> am System angemeldet hat, kann man unter [Project](#) -> [Access & Security](#) -> [API Access](#) -> [Download OpenStack RC file](#) eine Protokollierungsdatei herunterladen, welche die benötigten Informationen enthält.

Vorbereitung der Interaktion mit OpenStack

Mit diesen Verbindungsinformationen ist man jetzt in der Lage, die notwendige Konfigurationsdatei `clouds.yml` unter dem Pfad `~/.config/openstack/clouds.yml` anzulegen.

Abbildung 53: Ausschnitt aus den Hinweisen zur Arbeitsvorbereitung

Aufgabenstellung Einmal skalieren bitte.

Die folgenden Aufgaben beschreiben die einzelnen Schritte, die notwendig sind, um die Fraktal-App in ein laufendes OpenStack-System zu übertragen. Die Anweisungen schreiben Sie bitte in eine Python-Skript-Datei, z.B. `whz-fraktal.py`. Zu jeder Aufgabe finden Sie unter *Hinweise anzeigen...* die entsprechenden Teilschritte bzw. Informationen, wie Methoden bzw. API-Aufrufe zu verwenden sind. Mit Hilfe von *Ergebnisse anzeigen...* können Sie ihre Ergebnisse mit einer Musterlösung abgleichen.

Vorbetrachtung

Ein oft verwendetes Schlagwort beim Entwurf von Software für die Cloud ist Skalierbarkeit. Im Klartext bedeutet dieser Begriff, dass eine Software in der Lage sein soll, zusätzliche Ressourcen verwenden zu können. Wenn man das etwas formaler ausdrücken möchte, lassen sich folgende Entwurfsanforderungen ableiten:

- Das System soll in der Lage sein, zusätzliche Hard- und Softwareressourcen nutzen zu können.
- Es sollte möglich sein, dem System neue (und notwendige) Ressourcen bereitzustellen.
- Es sollte möglich sein, dem System (aktuell nicht verwendete) Ressourcen zu entziehen.

In den vorangegangenen Abschnitten wurde eine allgemeine Einführung in die Struktur und die Wirkungsweise einer Cloud-Anwendung gegeben. Gleichzeitig wurde demonstriert, wie mit einer Cloud-API kommuniziert werden kann. Im Folgenden soll anhand des modularen Aufbaus gezeigt werden, wie man durch Entkopplung der einzelnen Komponenten beispielsweise Performance-Flaschenhälse identifizieren kann. Solche Informationen sind vor allem aus Anbietersicht (z.B. in Hinblick auf Kosteneffizienz) enorm wichtig für den Erfolg von Cloud-Systemen. Das langfristige Ziel sollte es sein, dass eine Zu- bzw. Abschaltung von notwendigen Ressourcen nicht nur möglich ist, sondern idealerweise vom System im laufenden Betrieb selbst erkannt wird.

Diese Übung umfasst die Aufteilung der Services der Anwendung auf mehrere Instanzen und hebt einige Aspekte in Sinne der Skalierbarkeit hervor. Schritt für Schritt werden bis zu sechs Instanzen in die Anwendung integriert. Bisher wurden zwei virtuelle Maschinen bzw. Instanzen verwendet: ein Controller bzw. Kontroll-Dienst und eine Worker-Instanz. Da die Geschwindigkeit der Fraktalerzeugung von der Anzahl der Worker-Instanzen abhängig ist, braucht man im Laufe der Zeit, d.h. mit steigender Anzahl von Anfragen auch mehr Worker-Instanzen. Um zu testen, wie die Fraktal-Anwendung unter Last reagiert, könnte man folgendes versuchen:

- Durch das Erstellen von einer Vielzahl von Tasks für die Worker-Instanzen wird die CPU maximal ausgelastet.
- Durch das Erstellen von einer Vielzahl von API-Service-Requests überfüllt man die Task-Warteschlange.

Aufgabe 1 - Last generieren

1.1 Erstellen Sie zusätzliche Worker-Tasks.

Hinweis anzeigen...

Ergebnis anzeigen...

1.2 Erstellen Sie zusätzliche API-Service-Requests.

Hinweis anzeigen...

Ergebnis anzeigen...

Aufgabe 2 - Skalierung einrichten

2.1 Entfernen Sie die derzeit existierende Anwendung.

Hinweis anzeigen...

Ergebnis anzeigen...

2.2 Aktualisieren Sie die Security Groups-Einstellungen.

Hinweis anzeigen...

Ergebnis anzeigen...

2.3 Erstellen Sie eine Hilfsfunktion für die Floating IP-Verwendung.

Hinweis anzeigen...

Ergebnis anzeigen...

2.4 Teilen Sie die Datenbank und die Message Queue auf.

Hinweis anzeigen...

Ergebnis anzeigen...

2.5 Skalieren Sie den API-Service.

Hinweis anzeigen...

Ergebnis anzeigen...

2.6 Skalieren Sie die Worker.

Abbildung 54: Ausschnitt Aufgabenstellung

Den Abschluss des Kurses bildete neben Literaturangaben und Weblinks eine Zusammenfassung des Gelernten.

Zusammenfassung Retrospektive und Ausblick

Was wurde in diesem Kurs vermittelt?

Innerhalb dieses Kurses sollte die Idee hinter OpenStack vermittelt werden und wie man eine eigene Testumgebung dafür aufsetzen kann. Daneben wurden auch die elementaren Konzepte und Technologien vorgestellt, aus denen ein OpenStack-Ökosystem bestehen kann. Weiterhin wurde in einem Beispiel gezeigt, wie OpenStack in der Softwareentwicklung bzw. Softwareverteilung eingesetzt werden kann.

Welche Fragen sollte man jetzt beantworten können?

Was ist OpenStack und aus welchen Komponenten besteht es?

Wie funktionieren die Aspekte Networking, Authentifizierung und Instanzverwaltung in einem OpenStack-System im Detail?

Wie lässt sich der Prozess der Softwareverteilung mit Hilfe von OpenStack beschreiben und welche Vorteile ergeben sich daraus?

Mit welchen Themen könnte man sich jetzt auseinander setzen?

Auf der Grundlage dieses Kurses sind nun verschiedene Richtungen denkbar, in denen man OpenStack weiter verfolgen kann...

- interoperable (Massen-)Virtualisierung (vom einfachen Webserver bis zur VM für Hochleistungs-Rechenaufgaben)
- verteilte Datenintegration und skalierbare Datenreplikation in Hochverfügbarkeitsszenarien
- Shared Infrastructures und Services im WAN-Bereich
- Aufbau von dienstleistungsgetriebenen Self-Service-Portalen für Cloud-Dienste
- Softwareentwicklung im Bereich KI sowie Virtual/Augmented Reality

Abbildung 55: Zusammenfassung des Kurses

Wissenschaftliche Begleitung

Befragung zum Semesterende:

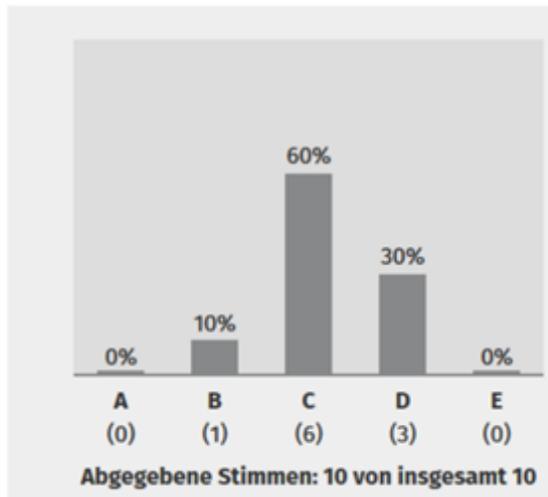
Auf eine Befragung zu Beginn des Semesters wurde verzichtet, da es sich um die gleichen Studierenden handelte, die bereits an der Erprobung von Grundlagen der Programmierung I und II sowie Entwicklung verteilter Anwendungen teilgenommen haben.

Zum Semesterende wurde mit den Studierenden des Studiengangs eine Tweedback-Befragung zum angelegten Opal-Kurs durchgeführt. Der Zeitpunkt der Befragung war so gewählt, dass die Studierenden die Modulverläufe mit etwas zeitlichem Abstand betrachten konnten.

Die Ergebnisse sind im Folgenden dargestellt.

Ich habe mir für das Lesen der theoretischen Inhalte genügend Zeit genommen. (1)

Quiz: 1 (Zeit für theo. Inhalte)

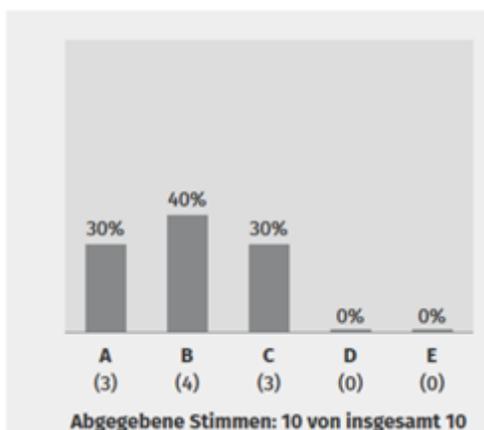


A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu
E	Kann ich nicht beurteilen

Abbildung 56: Zeitaufwand

Ich habe die Arbeitsvorbereitungen sorgfältig genug gelesen. (2)

Quiz: 2 (Arbeitsvorber. sorgfältig gelesen)

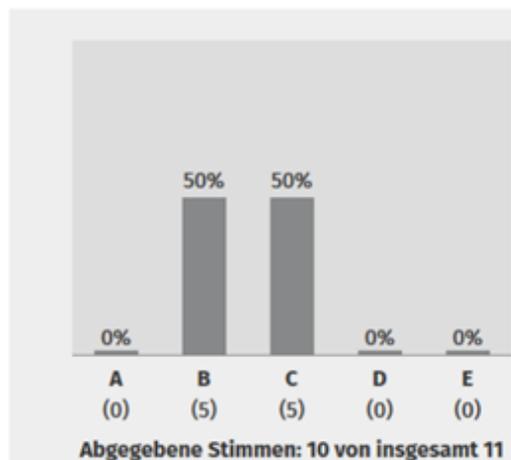


A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu
E	Kann ich nicht beurteilen

Abbildung 57: Leseaufwand Arbeitsvorbereitungen

Ich habe die praktischen Aufgabenstellungen mit dem nötigen Aufwand durchgearbeitet. (3)

Quiz: 3 (Aufgabenstellungen Aufwand)

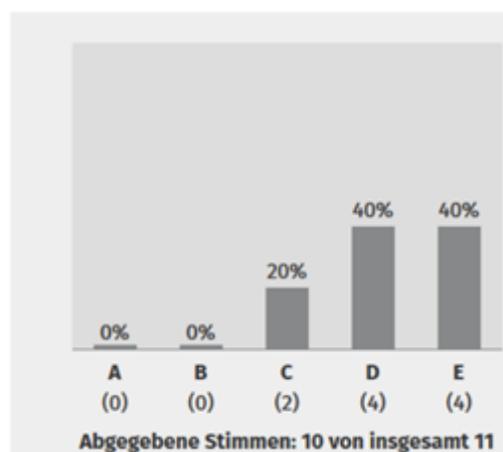


A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu
E	Kann ich nicht beurteilen

Abbildung 58: Arbeitsaufwand praktische Aufgabenstellungen

Wie hilfreich empfanden Sie die Übersichtsseiten (Arbeitsumfang, etc.) zu den einzelnen Themenblöcken? (5)

Quiz: 5 (Übersichtsseiten)

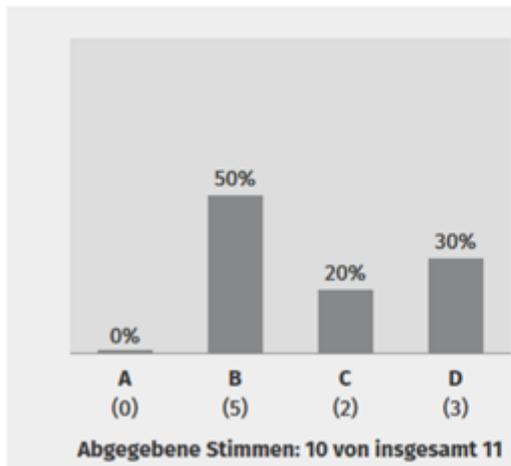


A	Sehr hilfreich
B	Hilfreich
C	Weniger hilfreich
D	Gar nicht hilfreich
E	Keine Angabe

Abbildung 59: Einschätzung Übersichtsseiten

Meine Fragen zum OPAL-Kurs waren hauptsächlich... (6)

Quiz: 6 (Fragen zum Kurs)

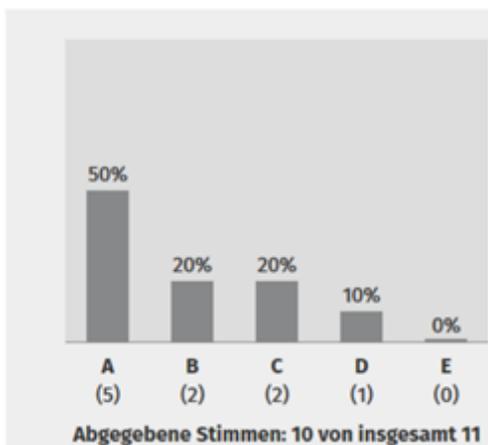


A	Fachliche Fragen
B	Technische Fragen
C	Verständnisfragen
D	Ich habe keine eingereicht

Abbildung 60: Qualität der Fragen

Wie hilfreich fanden Sie die Hinweise und Musterergebnisse zu den einzelnen Aufgaben? (7)

Quiz: 7 (Hinweise hilfreich)

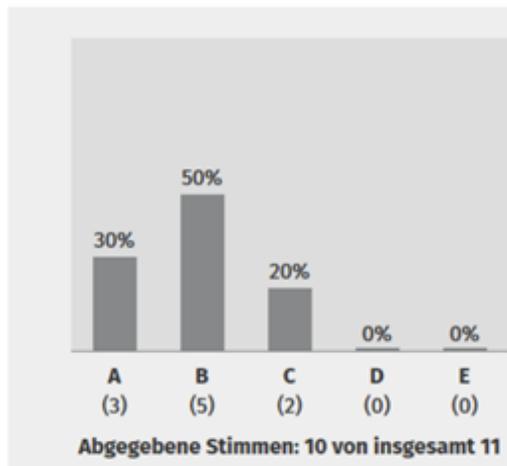


A	Sehr hilfreich
B	Hilfreich
C	Weniger hilfreich
D	Gar nicht hilfreich
E	Keine Angabe

Abbildung 61: Bewertung der Hinweise und Musterergebnisse

Ich empfand den Arbeitsaufwand für den OpenStack-Kurs als... (8)

Quiz: 8 (Arbeitsaufwand Kurs)

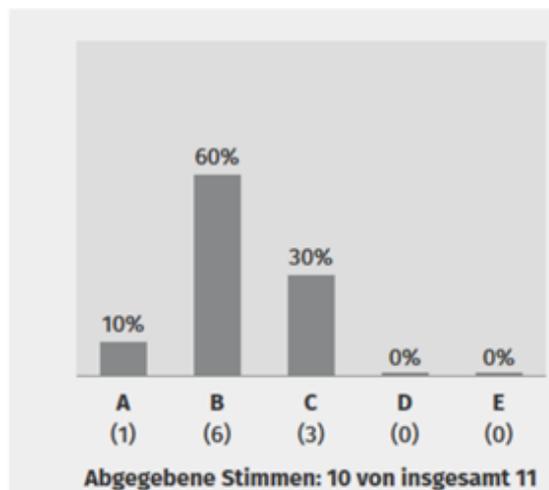


A	Viel zu hoch
B	Zu hoch
C	Angemessen
D	Zu gering
E	Viel zu gering

Abbildung 62: Einschätzung Arbeitsaufwand

Ich empfand die Lernkurve als... (9)

Quiz: 9 (Lernkurve)



A	Viel zu steil
B	Zu steil
C	Angemessen
D	Zu flach
E	Viel zu flach

Abbildung 63: Einschätzung Lernkurve

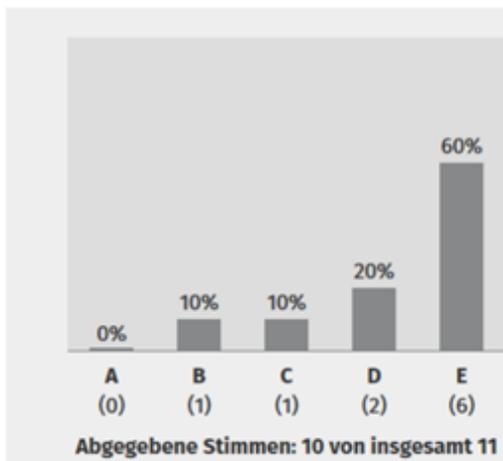
Anmerkungen der Studierenden:

- Eigentlich bzgl. der Aufgaben ok, aber Theorieteile wurden als zu schwierig empfunden
- Da die Aufgaben teilweise nicht ausgeführt werden konnten, war der weitere Verlauf der Lernkurve nicht einschätzbar

- Die Aufgabe zur Installation wurde als die schwierigste Aufgabe empfunden

Die Zielstellung des Themenbereichs war auf der Übersichtsseite klar formuliert. (10)

Quiz: 10 (Formulierung Zielstellungen)

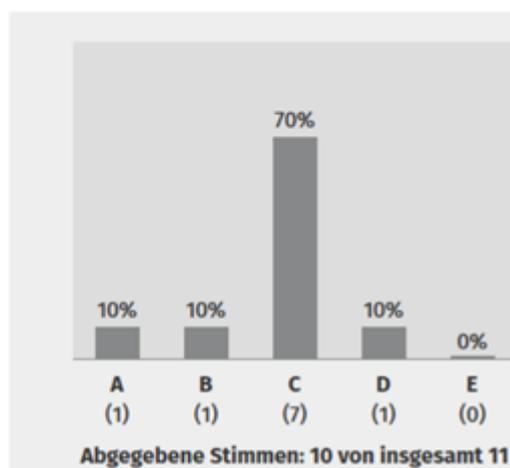


A	Trifft sehr zu
B	Trifft zu
C	Trifft weniger zu
D	Trifft nicht zu
E	Kann ich nicht beurteilen

Abbildung 64: Formulierung Zielstellung

Wie gut hat der OPAL-Kurs inhaltlich zum Modul und zu den Lehrveranstaltungen gepasst? (11)

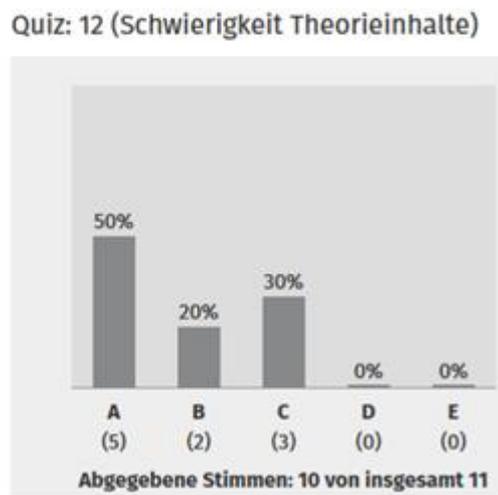
Quiz: 11 (Kurs passt zur LV)



A	Sehr gut
B	Gut
C	Mittelmäßig
D	Schlecht
E	Sehr schlecht

Abbildung 65: inhaltliche Passgenauigkeit

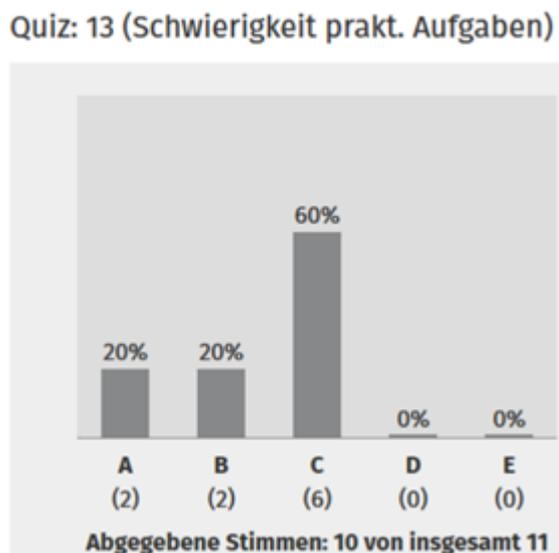
Wie schwierig empfanden Sie die dargestellten Theoretischen Inhalte: (12)



A	Viel zu schwierig
B	Zu schwierig
C	Angemessen
D	Zu leicht
E	Viel zu leicht

Abbildung 66: Schwierigkeitsgrad der theoretischen Inhalte

Wie schwer empfanden Sie die praktischen Aufgaben? (13)

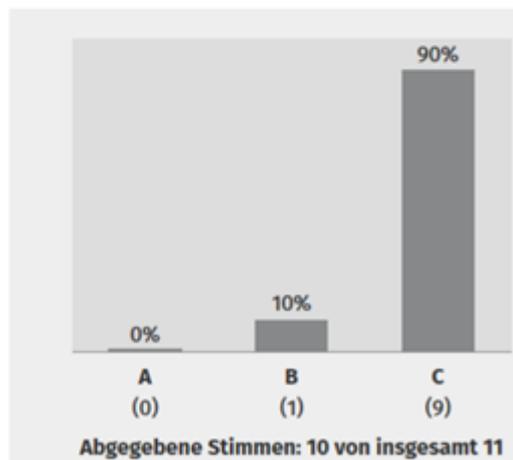


A	Viel zu schwer
B	Zu schwer
C	Angemessen
D	Zu leicht
E	Viel zu leicht

Abbildung 67: Schwierigkeitsgrad praktische Aufgaben

Wie sind Sie mit den bereit gestellten Hinweisen, Weblinks und Quellen umgegangen? (14)

Quiz: 14 (Weblinks und Quellen)



A	Oft genutzt
B	Selten genutzt
C	Nicht genutzt

Abbildung 68: Quellennutzung

Fazit

Die theoretischen Inhalte wurden von den Studierenden auf Grund von Schachtelsätzen, und nicht erklärten Fremd- und Fachwörtern als sehr schwer verständlich eingeschätzt, so dass sie sehr viel Zeit zum Lesen benötigten. Wenn der Kurs jedoch von den Lehrenden frühzeitiger freigeschaltet wird, so dass die Studierenden mehr Zeit für die Bearbeitung haben, sollte dies bei einer Wiederholung des Angebots kein Problem mehr darstellen, da es eher wenige inhaltliche Abhängigkeiten zu Vorlesungsinhalten gibt.

Weiterhin gab es technische Probleme, da mit Fortschreiten der Aufgaben die Lernumgebung immer weniger funktionierte. Hier kann gegebenenfalls nochmal deutlicher herausgestellt werden, dass für die Bearbeitung der weiteren Aufgaben keine eigene Installation nötig ist.

Dass es Übersichtsseiten auf der Ebene über den einzelnen Seiten gibt, war vielen Studierenden aufgrund mangelnder OPAL-Kenntnisse gar nicht bewusst. Hier sollten die Studierenden noch besser in die Lernplattform eingewiesen werden.

Nach Einschätzung des Dozenten wurde die zur Verfügung gestellte Zeit nicht ausreichend genutzt und es war keine Vor- oder Nachbereitung zwischen den Terminen erkennbar. Der Kurs wurde insgesamt über drei Wochen bearbeitet, wofür acht Präsenzveranstaltungen auch zur Verfügung gestellt wurden. Der Dozent schätzt das Engagement der Studierenden als recht gering ein.

2 Abbildungsverzeichnis

Abbildung 1: Darstellung der Erklärung zur Fragenabgabe im OPAL	8
Abbildung 2: Darstellung und Steuerung der Projektarbeit im OPAL	9
Abbildung 3: Ergebnis des Tweedbacks zum Umgang mit dem Greenfoot-Buch.....	10
Abbildung 4: Ergebnisse des TAPs zur Durchführung des Moduls als Flipped Classroom	11
Abbildung 5: Ergebnisse des TAPs zur Durchführung des Moduls als Flipped Classroom	16
Abbildung 6: Ergebnis des Tweedbacks zu den Texten	17
Abbildung 7: Ergebnis des Tweedbacks zur Fragenabgabe	18
Abbildung 8: Ergebnis des Tweedbacks zum Lerneffekt des Flipped Classrooms	19
Abbildung 9: Ausschnitt Semesterplan	21
Abbildung 10: Lehrvideos auf Opal.....	22
Abbildung 11: Umgang mit Semesterplan	24
Abbildung 12: Umgang mit Forum	24
Abbildung 13: Arbeitsstand Praktikumsaufgabe	25
Abbildung 14: Strukturierung des Kurses.....	25
Abbildung 15: Arbeitsaufwand	26
Abbildung 16: Quellenverständnis	26
Abbildung 17: Lerneffekt Quellen	27
Abbildung 18: Einschätzung Tutorialvideos	27
Abbildung 19: Lehrtempo	28
Abbildung 20: Schwierigkeitsgrad EvA-Projekt	28
Abbildung 21: Leseaufwand Greenfoot-Buch	33
Abbildung 22: Bewertung Greenfoot-Buch als Lehrmittel	33
Abbildung 23: Bewertung Themenblöcke.....	34
Abbildung 24: Qualität hochgeladene Fragen	34
Abbildung 25: Bewertung der Fragenbeantwortung.....	35
Abbildung 26: Arbeitsaufwand GdP 1	36
Abbildung 27: Lehrtempo GdP 1	36
Abbildung 28: Kommunikation der Erwartungen.....	37
Abbildung 29: Ziel im Modul	37
Abbildung 30: Inhaltliche Passgenauigkeit Texte GdP 2	38
Abbildung 31: Schwierigkeitsgrad Texte GdP 2	38
Abbildung 32: Sinnhaftigkeit englischer Quellen	39
Abbildung 33: Qualität Fragen GdP 2	39
Abbildung 34: Qualität Fragenbeantwortung GdP 2	40
Abbildung 35: Arbeitsaufwand GdP 2	40
Abbildung 36: Lehrtempo GdP 2	41
Abbildung 37: Kommunikation Erwartungen GdP 2	41
Abbildung 38: Ziel im Modul GdP 2	42

Abbildung 39: Umgang mit Quellen in EvA.....	42
Abbildung 40: Einschätzung Anfangsphase EvA.....	43
Abbildung 41: Arbeitsaufwand EvA.....	44
Abbildung 42: Lehrtempo EvA.....	45
Abbildung 43: Kommunikation Erwartungen EvA.....	45
Abbildung 44: Ziel im Modul EvA.....	46
Abbildung 45: Gesamteinschätzung GdP 1 und 2 sowie EvA.....	46
Abbildung 46: Startseite des OPAL-Kurses im Modul "Netzwerke".....	49
Abbildung 47: Gliederung des OPAL-Kurses im Modul "Netzwerke".....	50
Abbildung 48: Beispiel für die Starseiten.....	51
Abbildung 49: Beispiel 1 für die Darstellung der theoretischen Grundlagen.....	51
Abbildung 50: Beispiel 2 für die Darstellung der theoretischen Grundlagen.....	52
Abbildung 51: Beispiel Startseite für die Übungen.....	53
Abbildung 52: Beispiel Motivationsseite.....	53
Abbildung 53: Ausschnitt aus den Hinweisen zur Arbeitsvorbereitung.....	54
Abbildung 54: Ausschnitt Aufgabenstellung.....	55
Abbildung 55: Zusammenfassung des Kurses.....	56
Abbildung 56: Zeitaufwand.....	57
Abbildung 57: Leseaufwand Arbeitsvorbereitungen.....	57
Abbildung 58: Arbeitsaufwand praktische Aufgabenstellungen.....	58
Abbildung 59: Einschätzung Übersichtsseiten.....	58
Abbildung 60: Qualität der Fragen.....	59
Abbildung 61: Bewertung der Hinweise und Musterergebnisse.....	59
Abbildung 62: Einschätzung Arbeitsaufwand.....	60
Abbildung 63: Einschätzung Lernkurve.....	60
Abbildung 64: Formulierung Zielstellung.....	61
Abbildung 65: inhaltliche Passgenauigkeit.....	61
Abbildung 66: Schwierigkeitsgrad der theoretischen Inhalte.....	62
Abbildung 67: Schwierigkeitsgrad praktische Aufgaben.....	62
Abbildung 68: Quellennutzung.....	63