

Andreas Schmietendorf

Implizite Praxiskooperationen in der IT-Lehre und – Forschung

Abstract

Moderne, die Digitalisierung treibende Lösungen finden sich mit mobilen Applikationen (Apps), Anwendungen der künstlichen Intelligenz (Data-Science), im Telematikbereich (Smart City) oder auch beim Internet der Dinge (Smart Home). Entsprechende Softwarelösungen machen umfangreichen Gebrauch von offenen und cloudbasiert zur Verfügung gestellten APIs (Application Programming Interface), über die Daten, Funktionen und Prozessabläufe global bezogen werden. Die Entwicklung derartiger Anwendungen ist zunehmend gekennzeichnet von Aufgaben der prozess- und datengetriebenen Integration jener (Open) APIs. Auch die zur Entwicklung eingesetzten Werkzeuge sind selbst Gegenstand der Virtualisierung, d. h., diese werden nicht mehr lokal installiert, sondern aus der Cloud bezogen. Konkrete Cloud-Lösungen beziehen sich auf Entwicklungs-, Test- und Laufzeitumgebungen. Dariüber hinaus erfährt die Organisation damit einhergehender Projekte eine Virtualisierung, d. h., in einem zeitlich begrenzten Rahmen arbeiten die Beteiligten über die Grenzen von Unternehmen, Behörden, Gremien und Forschungseinrichtungen international zusammen. Die akademische Ausbildung in der Informatik muss diesem Trend Rechnung tragen, woraus vielfältige Veränderungen an den Hochschulen und Universitäten resultieren. In diesem Beitrag soll auf konkrete Anforderungen, potenzielle Lösungsansätze, aber auch auf Risiken und Problembereiche eingegangen werden.

1 Veränderte Anforderungen

Seit fast 20 Jahren lässt sich eine zunehmende Virtualisierung und Globalisierung bei Informations- und Kommunikationssystemen beobachten. Gegenstand der Virtualisierung sind sowohl technische Systeme als auch Projekt- und Betriebsorganisationen bis hin zum Extremfall eines ausschließlich als Algorithmus existierenden Unternehmens. Für die Softwareentwicklung kann z. B. auf lokal installierte Entwicklungs-, Test- und Laufzeitumgebungen (z. B. durch

Einsatz der Online-Entwicklungsumgebung wie Eclipse Che¹ verzichtet werden, aber auch mit Teams an weltweit verteilten Standorten zusammengearbeitet werden. Gegenstand der Virtualisierung waren zunächst benötigte Plattformen zum Informationsaustausch. Zunehmend werden allerdings auch die Komponenten und Werkzeuge sowie kurzzeitig involvierte Aufgabenträger virtualisiert in den Softwareentwicklungsprozess einbezogen. Unter der Voraussetzung einer breitbandigen Internetintegration etablieren sich so Softwareentwicklungsprojekte, deren Bearbeitung von Ort und Zeit unabhängig erfolgen kann².

Es ist naheliegend, dass die Softwareentwicklung, die selbst als primär technologischer Treiber der Digitalisierung fungiert, auch sehr frühzeitig mit der Kultur der Digitalität konfrontiert war.

Korrespondierende Formen³ lassen sich als immanenter Veränderungsdruck auf Projektorganisationen zur Softwareentwicklung beobachten. Die charakteristischen Formen der Digitalität nach Stalder (2016)⁴ finden sich z. B. in der folgenden Weise (ohne Anspruch auf Vollständigkeit):

- *Referentialität* – Auswahl globalisiert und virtualisiert angebotener Komponenten, interdisziplinäre Zusammenarbeit, agile Prozessorganisationen oder auch bedarfsoorientierte Einbeziehung von Ressourcen (z. B. Offshoring).
- *Gemeinschaftlichkeit* – Etablierung interessengetriebener Projektorganisationen (Communitys) über Unternehmensgrenzen hinweg, mit Bezug auf vergemeinschaftete Verantwortlichkeiten.
- *Algorithmizität* – die vielfältigen Aufgaben innerhalb der Softwareentwicklung werden selbst zum Gegenstand der Automation, mit einhergehenden Veränderungen der benötigten Professionen.

Es bestehen unterschiedlichste Fragen hinsichtlich der Berücksichtigung der Digitalität innerhalb der Lehre, wie z. B. hinsichtlich des Bedarfs einer veränderten Wissensvermittlung, neuer didaktischer Konzepte oder auch benötigter infrastruktureller Voraussetzungen⁵.

„Während Wissenschaft und Forschung das Potenzial der Digitalität im Blick auf ihre Möglichkeiten und Grenzen zumindest partiell diskutieren und in seiner Bedeutung für die Entwicklung der jeweils eigenen Disziplin kritisch reflektieren, blieben entsprechende Fragen im Bereich der Lehre lange Zeit ungestellt.“

1 <https://www.eclipse.org/che/>.

2 Vgl. Nitze et al. (2018), S. 22.

3 Vgl. Stalder (2016).

4 Vgl. ebenda.

5 Gleim (2019).

Gerade für die Entwicklung innovativer Lösungsansätze wie z. B. mobiler Applikationen, Lösungen in der Data-Science, im Telematikbereich oder auch beim Internet der Dinge gilt es der Kultur der Digitalität in der akademischen Ausbildung respektive bei Veranstaltungen zum Softwareengineering Rechnung zu tragen. Es gilt für den „Blick über den Tellerrand“ zu sensibilisieren, um so neue Wege der Lösungsfindung anzuregen bzw. Kreativität explizit zu unterstützen.

Neben dem Zugriff auf diversifizierte Informationsquellen gilt das Interesse dabei vor allem der Berücksichtigung agil einsetzbarer Kollaborationsplattformen, der interdisziplinären Zusammenarbeit mit Praxis- und Forschungspartnern, der Verwendung modellbasierter Methoden, aber auch dem unkomplizierten Einsatz herstellerspezifischer Technologien, Techniken und Produkte.

Ein weiterer Treiber dieser Entwicklung findet sich mit weltweit arbeitenden Open-Source-Communitys, wie z. B. der *Apache Software Foundation* (<https://www.apache.org>). Zunehmend beteiligen sich auch Studenten an internationalen Entwicklungsprojekten, die nicht selten in den Vorlesungen der Informatik bzw. Wirtschaftsinformatik initiiert werden.

2 Beispiele für virtuelle Serviceangebote

Gerade in den höheren Fachsemestern kann eine implizite Kooperation mit entsprechenden Partnern aus der Industrie bzw. Forschung durch die Nutzung cloudbasierter Laborumgebungen beobachtet werden. Zumeist erfolgt für die Entwicklung der Einsatz eines eigenen Endgeräts (BYOD – bring your own device), wobei die Palette vom klassischen Laptop bis zum Smartphone reicht.

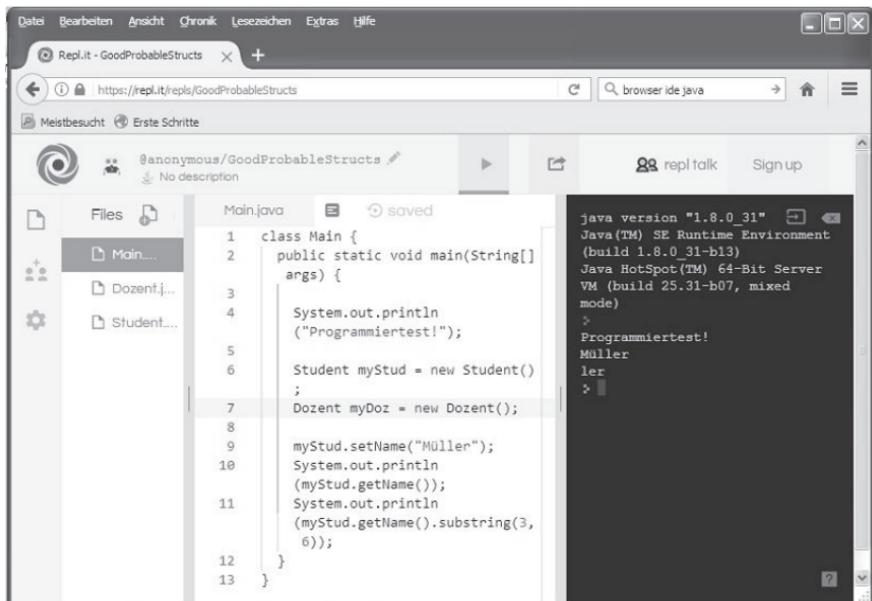
Aufgrund der einfachen Verwendung von Cloud-Services bietet deren Einsatz gerade in der akademischen Ausbildung und Forschung eine ideale Grundlage für experimentelle Untersuchungen, wie diese z. B. unter Weiss (2017) in folgender Weise charakterisiert werden:

„.... Entwicklungen von neuen Anwendungen, die in der Cloud eine größere Plattform- und Middleware-Vielfalt vorfinden, mit denen dort zunächst experimentiert werden kann, bevor entsprechende eigene Investitionen getätigten werden.“⁶

Unternehmen wie auch Interessengruppen unterstützen diesen Trend, um eigene Methoden, Produkte und Services gegenüber potenziellen Berufseinsteigern zu vermarkten. Darüber hinaus kann so Feedback in die Weiterentwicklung dieser Lösungen einfließen bzw. Fehler können durch crowdbasierte Tests (d. h. eine große Menge involvierter Studenten) gefunden werden.

6 Weiss (2017).

Abbildung 1: Beispiel einer einfachen Web-IDE für die Java-Entwicklung



Quelle der Abbildung: <https://repl.it/repls/GoodProbableStructs>

Abbildung 1 zeigt exemplarisch eine ausschließlich browserbasiert zu nutzende Java-Entwicklungsumgebung, d. h., auf die lokale Installation von Systemkomponenten kann hier komplett verzichtet werden.

Beispiele für cloud- und damit webbasierte Werkzeuge bzw. APIs finden sich mit der Verwendung von (korrespondierende Internetquellen siehe Anlage):

- Versions- und Konfigurationssystemen wie *GitHub* zur Unterstützung der kollaborativen und im Internet verteilten Bearbeitung von Softwareprojekten,
- Frage- und Antwortverwaltung (sukzessiver Aufbau einer freien Wissensdatenbank) zum Thema der Softwareentwicklung unter *Stack Overflow*,
- Open APIs die über Unternehmensportale oder Verzeichnisdienste (u. a. API-Marktplätze) zur Verfügung gestellt werden, wie z. B. *Programmable Web*,
- virtualisierten Laufzeitumgebungen, wie diese z. B. mit den *Amazon Web Services – Simple DB* angeboten werden,

- Werkzeuge zur Prozess- und Softwaremodellierung wie z. B. *BPMN.IO* oder auch *UML Letino*,
- Projektdokumentation/-präsentation mithilfe webbasierter Office-Produkte wie z. B. *Google Docs*, *Microsoft Office 365* und *Prezi*,
- statistischen Auswertungen mit vorgefertigten Werkzeugen wie z. B. *Google Trends* oder *Google Analytics*,
- APIs für KI-Algorithmen, wie z. B. das *Machine Learning Studio* der Firma Microsoft oder auch die *Bluemix-Platform* der Firma IBM.

Die exemplarisch genannten Beispiele zeigen die Vielfältigkeit virtualisierter Angebote. Werden diese eingesetzt, besteht der Bedarf, innerhalb der Softwareentwicklung mit Interessengruppen bzw. konkreten Industrieunternehmen kollektiv, interdisziplinär und vor allem ad hoc zusammenzuarbeiten. Beispielsweise sei auf die Nutzung der Kollaborationsplattform *GitHub* mit aktuell mehr als 36 Millionen Nutzern weltweit verwiesen. Von den Anbietern der virtualisierten Werkzeuge, Services und APIs wird typischerweise ein abgestuftes Nutzungskonzept verfolgt:

- anonyme und unlimitierte Nutzung,
- anonyme, aber zeitlich begrenzte Nutzung,
- Registrierung und kostenfreie Nutzung,
- Registrierung und limitierte (u. a. Zugriffe, Daten, Zeit) kostenfreie Nutzung,
- Verrechnung genutzter Leistungen nach Zugriffen, Daten oder auch Zeit,
- Erfolgsbeteiligung an implementierten Lösungen (z.B. *Revenue Sharing*).

Eine weitere Form der Virtualisierung findet sich mit crowdbasert durchgeführten Aufgaben der Softwareentwicklung. Die Entlohnung entsprechender Aufgaben orientiert sich an Einzel- oder auch Projektaufgaben. Unter Peng (2014)⁷ finden sich allgemeine Modelle für eine crowdbaserte Zusammenarbeit, damit einhergehende Anforderungen und Problembereiche, aber auch praktische Beispiele für konkrete Crowd-Plattformen. In Nitze (2014) finden sich detaillierte Erläuterungen zur Idee des Crowd-Testings⁸. Exemplarisch für konkrete Lösungen seien die folgenden Beispiele genannt:

7 Peng (2014), S. 12.

8 Nitze (2014), S. 48.

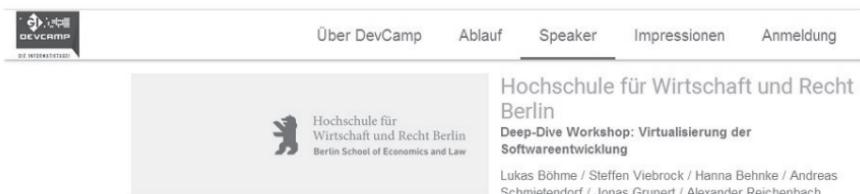
- *Topcoder* – globale Interessengemeinschaft zur Softwareentwicklung mit mehr als einer Million Softwareentwicklern und Datenwissenschaftlern,
- *Testbirds* – Anbieter von crowd- und cloudbasierten Softwaretests, mit weltweit mehr als 300.000 Testern.

Wird die verteilte und cloudbasierte Entwicklung innovativer Softwarelösungen auch im Diskurs von Vorlesungen als didaktisches Merkmal verwendet, kommt es zwangsläufig zu zeitlich begrenzten und ad hoc eingegangenen Kooperationsbeziehungen. Zumeist entziehen sich diese einer hochschulspezifischen Kontrolle bzw. die Standardprozesse der Hochschul-IT können diese nicht adäquat managen. Korrespondierende Risiken finden sich in Bezug auf die Informationssicherheit, aber auch auf die IT-Compliance.

3 Beispiel eines studentischen Workshops

Der im Folgenden dargestellte Deep-Dive-Workshop (veranstaltet im Rahmen der GI-Jahrestagung 2018) entstand ausschließlich auf der Basis studentischer Projektarbeiten. Diese wurden im Rahmen einer Spezialvorlesung (höheres Fachsemester) zu komplexen Integrationsarchitekturen beim Autor dieses Beitrags erarbeitet. Im Mittelpunkt des Workshops auf der GI-Jahrestagung stand die Virtualisierung der Softwareentwicklung (vgl. Abbildung 2).

Abbildung 2: Hinweis auf den Workshop im Rahmen der GI-Jahrestagung 2018



The screenshot shows a website for a workshop. At the top, there is a navigation bar with links: 'Über DevCamp', 'Ablauf', 'Speaker', 'Impressionen', and 'Anmeldung'. The 'Anmeldung' link is underlined. Below the navigation bar, there is a logo for 'Hochschule für Wirtschaft und Recht Berlin' (Berlin School of Economics and Law) featuring a bear logo and the text 'Hochschule für Wirtschaft und Recht Berlin' and 'Berlin School of Economics and Law'. To the right of this logo, the text 'Hochschule für Wirtschaft und Recht Berlin' is repeated, followed by 'Deep-Dive Workshop: Virtualisierung der Softwareentwicklung'. Below this, a list of names is provided: Lukas Böhme / Steffen Viebrock / Hanna Behnke / Andreas Schmietendorf / Jonas Grunert / Alexander Reichenbach.

Quelle: <https://berlin.dev-camp.com/> (letzter Abruf: 22. Februar 2019)

Die korrespondierende Vorlesung wurde durch die Lehrplattform Moodle, aber ebenfalls durch einen eigenen Internetauftritt des Dozenten unterstützt. Neben der Informationsbereitstellung erfolgte auch die Abgabe der anzufertigenden Protokolle terminiert über das Moodle-System. Die hochschuleigene Laborumgebung kam dabei nur sporadisch zum Einsatz, zumeist wurden von den Studenten eigene Endgeräte (BYOD – WLAN-Integration über eduroam) und

cloudbasierte (virtualisierte) Services für die prototypische Entwicklung genutzt.

Im Detail wurden studentische Arbeitsgruppen durch die folgenden Einflüsse (detailliert beschriebene Laboraufgaben) motiviert:

- Suche, Analyse und Bewertung im Internet angebotener Open APIs bzw. Open-Data-Services, die zur nachhaltigen Unterstützung fachlicher Anwendungsszenarien eingesetzt werden können,

- Spezifikation (z. B. Swagger/Open API), programmiertechnische Einbindung bzw. Entwicklung fachlich orientierter Service-APIs mithilfe von Programmiersprachen wie Java und JavaScript,
- fachdomänenspezifische Auseinandersetzung mit benötigten Referenzmodellen für Prozesse, Daten, Applikationen und APIs (z. B. Ansatz der Telekommunikationsindustrie [TMF 2018]),
- Herleitung von geschäftsprozess- und geschäftsdatenorientierten API-Kompositionen, die u. a. das Rückgrat der benötigten Softwarearchitektur für mobil eingesetzte Apps bilden,
- Berücksichtigung der Funktionen eines über den gesamten Lebenszyklus (Plan – Build – Run) kompositorisch erstellter Softwarelösungen benötigten API-Managements,
- Einsatz cloud- bzw. webbasierter Werkzeuge zur Modellierung, Implementierung, Test und Ausführung (virtualisierte Entwicklung), sodass auf lokale Installationen weitgehend verzichtet werden konnte.

Die Nutzung oder gar Beteiligung an nationalen und internationalen Interessengruppen bzw. die Ad-hoc-Etablierung von Wissenschafts- und Praxiskooperationen wurde als didaktisches Mittel bewusst eingesetzt. Die studentischen Arbeitsgruppen waren daher mit der Auswahl technologischer Komponenten, der Festlegung zu nutzender Organisationen und Prozesse, dem Umgang mit Sicherheits- und Compliance-Fragen, der Bewältigung von Problemsituationen, aber auch dem betrieblich benötigten Management für zumeist prototypisch implementierte API-Kompositionen konfrontiert.

Die konkreten „Hands on“-Beiträge innerhalb des Workshops griffen die folgenden Themengebiete auf:

- *Hanna Behnke, Alexander Reichenbach*: Kompositorische App-Entwicklung mit Web-APIs,
- *Steffen Viebrock*: Spezifikation und Prozessabbildungen (RESTful Prozesse) mit Open APIs,
- *Lucas Böhme, Jonas Grunert*: GraphQL (Hintergründe, Einsatzgebiete und Typsystem) – BEYOND REST/SQL für Web-APIs.

Details bzw. die korrespondierenden Präsentationen finden sich unter der Web-Präsenz des Autors <https://blog.hwr-berlin.de/schmietendorf>.

4 Ausrichtung der IT im Hochschuldiskurs

Ähnlich den Aufgabenstellungen im unternehmerischen Kontext fokussiert die Informationsverarbeitung an den Hochschulen primär auf prozessbezogene Verwaltungsaufgaben und hochschulweit etablierte Standardservices. Die Aussage von Pongratz (2017) zur Entwicklung der IT-Architektur einer digitalen Hochschule stützt diese Einschätzung.

„Die IT-Architektur der digitalen Hochschule hat die nutzerorientierte Abbildung der Geschäftsprozesse zum Ziel und ermöglicht eine nahtlose Einbindung von externen Diensten auf allen Ebenen des Schichtenmodells.“

Hochschulweit angebotene Standardservices, die durch alle Mitglieder der Hochschule konsumiert werden, beziehen sich häufig auf

- Office-Lösungen, Termin- und Projektmanagement sowie Anwendungen für den gesicherten Datenaustausch (inklusive klassischer E-Mails),
- spezialisierte Videokonferenzsysteme, die zunehmend durch webbasierte Pendants wie z. B. Adobe Connect ersetzt werden,
- E-Learning-Plattformen, auf deren Grundlage digitale Lehr- und Lernangebote (Foren, Wikis, virtuelle Klassenzimmer, ...) erarbeitet werden,
- Content-Management-Systeme für benötigte Web-Auftritte, wobei gegebenenfalls auch ad hoc benötigte Internetpräsenzen abgedeckt werden,
- Massive Open Online Courses (MOOCs) für digitale Hochschulen auf der Grundlage virtualisierter Lehrangebote (ohne Präsenzpflicht).⁹

Bei den Aufgaben des Softwareengineering für innovative Lösungsansätze zeigen sich klare Grenzen im Zusammenhang mit den aufgezeigten Standardservices, da die Anforderungen hinsichtlich der benötigten Rahmenbedingungen nur unzureichend erfüllt werden. Zumeist erfolgt der Versuch, diese Herausforderungen mithilfe von produktzentrierten Speziallaboren aufzugreifen, welche allerdings nur einen Teil der Anforderungen befriedigen und darüber hinaus einen hohen Arbeitsaufwand implizieren. Die folgende These kann so insbesondere auf die Ausbildung im Software-Engineering übertragen werden kann:

⁹ Im Bereich der IT-orientierten MOOCs (auch Aspekte des Softwareengineering) werden z. B. Veranstaltungen durch das HPI (Open HPI – <https://open.hpi.de>) an der Universität Potsdam angeboten.

„Der Schlüssel zu einer besseren und individuelleren Hochschulbildung liegt nicht nur in der Verlagerung von Lehre auf digitale Plattformen, sondern vor allem in der Ermöglichung kollaborativen Lernens und neuer Formen der Zusammenarbeit zwischen Individuen und zwischen Institutionen.“¹⁰

Gerade die Ermöglichung des kollaborativen Lernens und die ad hoc benötigte Zusammenarbeit zwischen Individuen bzw. Institutionen (Unternehmen und Hochschule) stoßen in der Realität schnell an Grenzen. Aus Sicht des Autors finden sich dafür die folgenden Ursachen:

- Unkenntnis der Hochschulangehörigen,
- unzureichend zur Verfügung stehende Ressourcen,
- unzureichender Veränderungswille der potenziell Beteiligten,
- Fokussierung auf klassische Office-Anwendungen,
- schwerfällige IT-Verwaltungsprozesse,
- Dominanz sicherheitsrelevanter Restriktionen,
- unzureichendes Netzwerkmonitoring zur Gefahrenabwehr,
- unklarer Rechtsrahmen beim kollaborativen Arbeiten,
- fragmentierte Prozessabläufe.

Nicht selten werden von den verantwortlichen Dozentinnen und Dozenten „Notlösungen“ akzeptiert bzw. sogar eigenständig realisiert. Um mit wissenschaftlichen und industriellen Partnern überhaupt kooperieren zu können, erfolgt typischerweise der Einsatz externer cloudbasierter Lösungen oder auch die Nutzung ad hoc aufgesetzter WLANs (mit unkompliziert administrierbaren Portfreigaben), anstatt hochschuleigene IT-Systeme zu nutzen. Eine institutionelle Beantragung von benötigten Hardware- und Softwareressourcen bzw. der im Diskurs einer verteilten Softwareentwicklung benötigten Portfreigaben mündet häufig in unklaren und zeitlich langen Beantragungswegen. Ebenso typisch ist der sicherheitskritische Datenaustausch mithilfe von Fileshare-Services wie z. B. der Dropbox¹¹ oder auch die Terminabstimmung via Doodle¹². In diesem Zusammenhang entstehen Grauzonen, die Risiken in Bezug auf Lizenzverstöße, Urheberrechte oder auch Sicherheitslücken implizieren.

10 Hochschulforum Digitalisierung (2015), S. 11.

11 <https://www.dropbox.com/de/>.

12 <https://doodle.com/de/>.

5 Benötigte Veränderungen

Für die Gewährleistung innovativer Lehr- und Forschungsansätze im Software-Engineering bedarf es aus Sicht des Autors gravierender Veränderungen bei den Beteiligten. Im Folgenden findet sich der Versuch, einige davon aufzuzeigen, ohne allerdings den Anspruch auf Vollständigkeit zu erheben:

Überführung der Speziallabore

Es gilt, die klassisch eng abgegrenzten Labore des Softwareengineering in virtualisierte Laborumgebungen zu überführen, die sich im Rahmen eines nationalen und internationalen Kompetenznetzwerks mit Partnern aus dem akademischen und industriellen Umfeld einfach integrieren lassen. Dementsprechend bedarf es einer drastischen Spezialisierung gegebenenfalls eigens angebotener (virtualisierter) Services, da diese mit hoher Qualität und vor allem Aktualität zu erbringen sind. In einem ersten pragmatischen Schritt könnte z. B. die Etablierung einer gemeinsamen digitalen Plattform auf Landesebene für alle Berliner Hochschulen erfolgen.

Insbesondere wird aber Unterstützung beim Management der ad hoc eingegangenen Kollaborationen benötigt. Hier gilt es, durch die Laborverantwortlichen sowohl juristische und vertragliche Implikationen aktiv zu gestalten. Zur Gewährleistung von Compliance-Anforderungen bedarf es eines Rahmenwerks, das grundlegende Kriterien zur Zusammenarbeit unter Berücksichtigung des gesetzlich Machbaren festlegt.

Aus technischer Sicht wird ein intelligentes und automatisiertes Monitoring durchgeföhrter Netzwerkinteraktionen benötigt. Auf dieser Grundlage sollte es möglich sein, Gefahrensituationen im Sinne eines „Fraud-Managements“ service- und portbezogen zu erkennen. Auf keinen Fall darf es dabei zur Behinderung innovativer Lösungsansätze kommen!

Erwartungen der Studenten

Im Zusammenhang mit der Lehre erwarten die Studierenden aktivierende, motivierende und reflektierende Lernerlebnisse, die sowohl im Diskurs von individualisierten als auch teamorientierten Lernformen entstehen. Auswertungen zu virtuell unterstützten Lehrveranstaltungen zeigen bei vielen Studenten die Ablehnung ausschließlich virtuell durchgeföhrter Lehrveranstaltungen, d. h., hier

wird eher ein Mix (Blended Learning) aus Präsenz- und Onlineveranstaltungen präferiert¹³.

Besondere Motivation erfahren studentische Teams durch die Mitwirkung an praxis- und forschungsorientierten Aufgabenstellungen. Neben der Schaffung von internetbasierten Informationspräsenzen gilt es hier die gegebenenfalls benötigte Zusammenarbeit (z. B. mithilfe von virtuellen Arbeitsräumen) mit Partnern aus Industrie und Forschung, aber auch unabhängigen Gremien zu gewährleisten. Die Studierenden haben so die Möglichkeit, innovative Ideen einzubringen und umzusetzen, aber auch einen digitalen Kompetenzabdruck innerhalb einschlägiger Interessengruppen zu hinterlassen.

Veränderungsdruck für Dozenten

Die Aufgaben des Dozenten erfahren in diesem Zusammenhang massive Veränderungen hin zum Selbstverständnis als Coach für studentische Teams in den höheren Fachsemestern (aus Sicht des Autors gilt dieses nicht für die Grundlagenfächer!). Dabei geht es weniger um die Vermittlung des Funktionsumfangs herstellerspezifisch einzusetzender Werkzeuge als vielmehr um die Auseinandersetzung mit Produkt-, Daten- und Prozessmodellen, die Berücksichtigung etablierter (gegebenenfalls standardisierter) Spezifikationen, um Risiko- und Sicherheitsaspekte, aber auch um methodische Fähigkeiten und soziologische Implikationen (u. a. Motivation, Streitkultur und Konfliktlösungen). Die studentische Auseinandersetzung mit den zu vermittelnden Lehrinhalten muss durch den Dozenten daher ergebnisoffener, aber dennoch kritisch reflektierend (Fehler sind z. B. offen zu diskutieren) betreut werden. Das schafft Raum für innovative Lösungen, impliziert aber auch einen höheren Aufwand für den Dozenten (infiniter Lösungsraum) und birgt gegebenenfalls auch die Gefahr „abgehängter Studenten“ in sich. Neben der qualitativen Sicherung der fachlich orientierten Lehrinhalte bedarf es dabei auch der Berücksichtigung hochschuldidaktischer Anforderungen.

Für Dozenten geht mit dieser Art von Lehrveranstaltungen darüber hinaus die Gefahr einer 24/7-Verfügbarkeit einher, da die hohe Onlinepräsenz (neben der eigentlichen Vorlesung) auch eine allgegenwärtige Ansprechbarkeit suggeriert. Hier gilt es einen disziplinierten Umgang mit der eigenen Ressource zu finden, was sich auch in der Deputatsabrechnung niederschlagen sollte.

13 Vgl. Schmietendorf (2008), S. 54.

6 Risiken virtualisierter Laborumgebungen

Die bisherigen Ausführungen zum Einsatz virtueller Elemente innerhalb der Ausbildung zur Softwareentwicklung wurden eher von den Chancen und Herausforderungen her als von den gegebenenfalls damit einhergehenden Risiken aus betrachtet. Bei diesen Risiken innerhalb der Virtualisierung von Lehre und Forschung sei allen voran die Abhängigkeit vom breitbandigen Zugriff auf das Internet genannt. Da dieses Problem für nahezu alle Industriebereiche eine Herausforderung darstellt und dementsprechend infrastrukturell sicherzustellen ist, soll an dieser Stelle darauf nicht weiter eingegangen werden.

Darüber hinaus finden sich die folgenden Problemstellungen, die allerdings einer genaueren empirischen Analyse bedürfen:

- unkontrollierter Einsatz virtueller Ressourcen, woraus gegebenenfalls unklare Compliance-Bedingungen resultieren,
- Angebote mit einem besseren Support werden durch studentische Arbeiten stärker berücksichtigt als Nischenanbieter,
- einseitige Präferenzen der Dozenten aufgrund unternehmerischer Affinitäten, Gefahr des Verlusts der akademischen Marktneutralität,
- Ausspähen innovativer Lösungen durch Konkurrenz bzw. unbefugter Zugriff auf Forschungsergebnisse,
- von Studierenden eingerichtete Accounts werden nach dem Ausscheiden nicht weiter verwaltet und bilden ein Sicherheitsrisiko,
- Gefahr überforderter Dozenten, die sich selbst als inhaltsorientierte Vermittler von Wissen (Problem der Abgrenzung) sehen,
- Verlust von Wissen im Zusammenhang mit dem Management lokal betriebener Softwareservices.

Die folgenden empirisch gewonnenen Ergebnisse entstammen einem Forschungsprojekt beim Autor, welche sich dem Einsatz von Web-APIs innerhalb der Softwareentwicklung (speziell Reengineering) widmeten. Befragt wurden industrielle Entwicklungsteams u. a. zu den Gründen, die gegen den Einsatz von Web-APIs (im Sinne eines virtualisierten Serviceangebots) sprechen. Schwerpunkte ergaben sich bei den folgenden beiden Aspekten:

- Sicherheitsbedenken sowie hoher Arbeits- bzw. Integrationsaufwand,
- unvollständige Dokumentation und Ausfallzeiten.

Beide Problemstellungen sind aus Sicht des Autors auch im Diskurs der Lehre und Forschung als potenzielle Risiken zu berücksichtigen.

Abbildung 3: Einsatzhemmnisse bei der Nutzung von Web-APIs

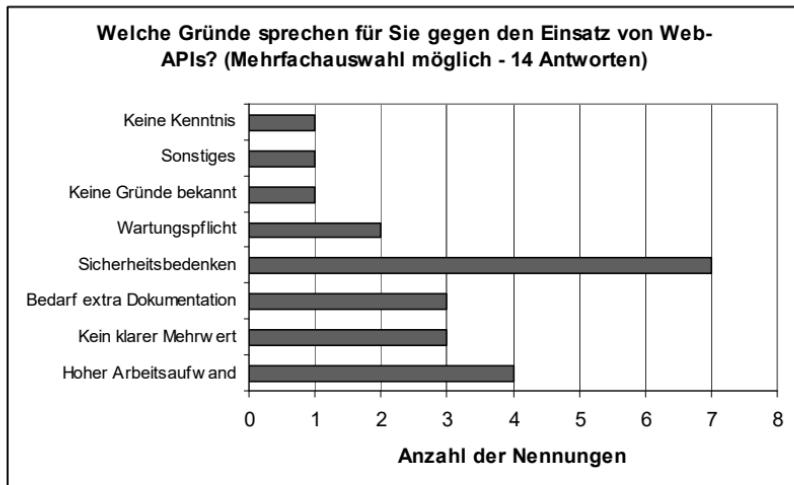
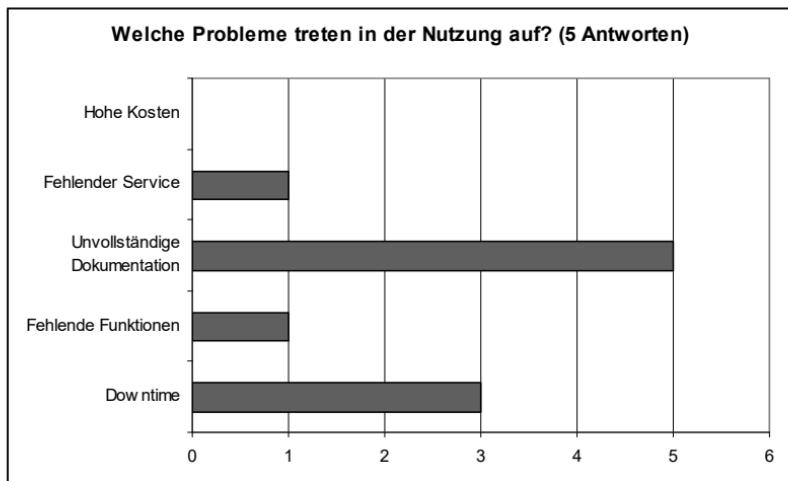


Abbildung 4: Probleme bei der Nutzung von Web-APIs



7 Zusammenfassung

Die Herausforderungen der Digitalisierung für die Ausbildung und Forschung im Bereich der Entwicklung von Softwareanwendungen sind immens. Innovative Lösungen lassen sich aus Sicht des Autors nur durch ein hohes Maß an Gestaltungsspielraum der kooperierenden Projektpartner und vor allem durch Kreativität der global und gegebenenfalls interdisziplinär eingesetzten Entwicklungsteams gewährleisten. Hier nehmen vertiefende Vorlesungen der Informatik eine Sonderrolle als Innovations-, aber auch Bedürfnistreiber im universitären Alltag ein. In Anlehnung an Stalder (2018)¹⁴ gilt es, dem gemeinschaftlichen Aspekt der Wissensproduktion innerhalb und zwischen den Disziplinen sowie zwischen Wissenschaften und Nichtwissenschaften (Praxispartner) als auszuprägende Kernkompetenz bei den Studierenden Rechnung zu tragen.

Beim Autor des Beitrags erfolgt sowohl im studentischen Alltag als auch im Zusammenhang mit Forschungsprojekten das Experimentieren mit virtuellen Arbeitsräumen. In diesem Zusammenhang sei auch auf die überholte Pflicht zur körperlichen Anwesenheit von studentischen und forschungsorientierten Mitarbeitern am Dienstort verwiesen. Diese macht im Zusammenhang mit globalisiert bearbeiteten Projekten schon aus Gründen der Zeitzonen wenig Sinn.

Im Sinne einer Erfolgsstory sei noch eine Forschungsarbeit im Data-Science-Diskurs angesprochen, dabei ging es speziell um Fragen des Performance-Verhaltens (konkret Geschwindigkeit der Datenübernahme) von Big-Data-Implementierungen. Durch die cloudbasierte Nutzung einer Hadoop-basierten¹⁵ Testinstallation konnte diese Fragestellung zeitnah innerhalb einer studentischen Abschlussarbeit untersucht werden. Gemietet wurde die Lösung für wenige Stunden von einem Praxispartner, wobei weniger als 30 Euro Kosten entstanden. Noch vor 20 Jahren wären solche Teststellungen nur wenigen Rechenzentren an forschungsstarken Universitäten in Deutschland vorbehalten gewesen!

14 Vgl. Stalder (2018), S. 14.

15 <https://hadoop.apache.org/>.

Literaturverzeichnis

- Bachert, L.: Eclipse Che – die IDE der Zukunft?, Heise Developer, Juli 2016, <https://www.heise.de/developer/artikel/Eclipse-Che-die-IDE-der-Zukunft-3266413.html?seite=all>
- Gleim, U.: Digitalität in der Lehre, Einladung zur Diskussion, Hochschule Darmstadt, <https://ikum.mediencampus.h-da.de/termin/digitalitaet-in-der-lehre>, Juni 2019.
- Hochschulforum Digitalisierung: Diskussionspapier-20 Thesen zur Digitalisierung der Hochschulbildung, Arbeitspapier Nr. 14. Berlin: Hochschulforum Digitalisierung, 2015
- Nitze, A./Schmietendorf, A.: Qualitative und quantitative Bewertungsaspekte bei der agilen Softwareentwicklung plattformübergreifender mobiler Applikationen, Forschungsstudie (Monografie), 166 Seiten, Logos-Verlag, Berlin, August 2014.
- Nitze, A./Schmietendorf, A./Nadobny, K./Kunisch, M.: Deutscher Mittelstand auf Wolke 7? – Cloud Computing in kleineren und mittleren Unternehmen 2015 und 2017, Redaktion/Layout: Möller Horcher Public Relations GmbH/Der Punkt GmbH, Januar 2018.
- Peng, X./Ali Babar, M./Ebert, C.: Collaborative Software Development Platforms for Crowdsourcing, IEEE Software magazine, August 2014.
- Pongratz, J. C. A.: IT-Architektur für die digitale Hochschule, Dissertation, Technische Universität München, Oktober 2017.
- Schmietendorf, A./Menke, S.: Blended-Learning-Konzepte an Hochschulen. Ein Erfahrungsbericht, in: Rundbrief 2/08 des Fachausschusses WI-MAW, S. 54–63, Gesellschaft für Informatik, September 2008.
- Stalder, F.: Kultur der Digitalität, edition suhrkamp, Mai 2016.
- Stalder, F.: Herausforderungen der Digitalität jenseits der Technologie, in Synergie – Fachmagazin für Digitalisierung in der Lehre, Ausgabe #5, Mai 2018.
- Open Digital Framework, TM Forum, <https://www.tmforum.org>, abgerufen Juli 2019.
- Weiss, H.: Das Rechenzentrum wird neu erfunden, VDI-Nachrichten – Ausgabe 28, 13.07.2017, <https://www.vdi-nachrichten.com>

*Allgemeine Internetquellen**Beispiele für cloudbasierte Werkzeuge:*

- <https://trends.google.de>
- <https://analytics.google.com>
- <https://github.com/>
- <https://stackoverflow.com/>
- <https://azure.microsoft.com/de-de/services/machine-learning-studio/>
- <https://bpmn.io/>
- <https://www.umletino.com/>
- <https://docs.google.com>
- <https://www.office.com>
- <https://prezi.com>

Beispiele für cloudbasierte APIs:

<https://www.programmableweb.com/>

<https://aws.amazon.com>

<https://www.ibm.com/de-de/cloud>

Beispiele für cloud- und crowd-basierte Entwicklerplattformen:

<https://www.topcoder.com/>

<https://www.testbirds.de/>