

Lasst uns Programmieren lernen!

Der Beitrag untersucht die Entwicklung der Hochschullehre speziell im Fachgebiet Softwareprogrammierung und ordnet sie vor dem Hintergrund der Kultur der Digitalität Felix Stalders ein. Es wird deutlich, dass heutige Probleme nicht ausschließlich durch die erschwerten Rahmenbedingungen wie Komplexität der Inhalte, knappe Unterrichtszeit oder fehlende Motivation der Studierenden bedingt, sondern auch Ergebnis einer kulturellen Entwicklung der vergangenen Jahrzehnte sind. Abschließend werden Ansätze vorgeschlagen, wie die Lehre zur Konstruktion digitaler Systeme mit dieser neuen Perspektive verbessert werden kann.

1. Einleitung

1.1. Problemstellung

Erfolgreiche Hochschullehre ist in den letzten Jahren immer schwieriger umzusetzen: Die Ansprüche an die Lehrinhalte sind aufgrund vielfältiger Konkurrenzsituationen hoch und der Einsatz der Studierenden muss durch formale Rahmenbedingungen und die allgemeinen Anforderungen des Lebens bis ins Kleinste optimiert werden. Aus der Perspektive der Lehrenden wird ein immer problematischeres Unterrichten beklagt, da die Aufmerksamkeit der Teilnehmer schwindet und die Wichtigkeit der Inhalte, solange sie nicht offensichtlich ist, regelmäßig hinterfragt wird. Die Problematik wird durch die Entwicklung des Hochschulwesens mit dem Ziel einer internationalen Vereinheitlichung verstärkt: Studieninhalte sollen möglichst vergleichbar werden, Fristen werden enger und der Nachweis von Studienergebnissen steht an erster Stelle.

Die ständig voranschreitende Digitalisierung in allen Lebensbereichen erweckt in der Lehre an manchen Stellen den Eindruck, dass sie eher vom Lernerfolg abhält, als diesen zu erhöhen. Gerade im persönlichen Verhalten und in der Einstellung zu den Lerninhalten wirken viele Studierende uninteressiert und ablehnend. Die Konzentration auf eine Fragestellung und mithin eine tiefere analytische Durchdringung eines Themas fällt oft schwer. Eine Erklärung ist durch die ablenkende Wirkung multimedialer und kurzfristig zu erfassender Eindrücke der heutigen Digitalisierung aller Lebensbereiche vermeintlich schnell gefunden.

Betrachtet man weiterhin den Lehrbereich der Softwareentwicklung, so fragt man sich zunächst, ob die fortschreitenden Kenntnisse im Umgang mit digitalen Medien und Plattformen nicht auch einen Vorteil beim Erlernen der Entwicklung von Algorithmen und Softwareprogrammen bringen. Das scheint jedoch keine allgemeingültige Regel zu sein, da für viele Studierende das Fach eine große Hürde darstellt, obwohl alle Teilnehmer der heutigen Einstiegskurse in die Programmierung als Digital Natives bezeichnet werden können.

Somit stellen sich viele Fragen, wie die Hochschullehre im Allgemeinen und speziell im Bereich der Softwareentwicklung verbessert werden kann. Digitale Verfahren zur Überbrückung von Raum- und Materialbeschränkungen sind ein Ansatz des heutigen E-Learnings, um Studierende zusammenzubringen und einfacher an Lehrinhalten teilhaben zu lassen. Im Blended Learning soll der Technologieeinsatz mit Präsenzanteilen der Lehre verknüpft werden. Weder das reine E-Learning noch eine Verknüpfung mit Präsenzlehre im Blended Learning sind vollständig befriedigende Problemlösungen für die oben genannten Herausforderungen, wie die aktuellen Entwicklungen zeigen.

1.2. Vorgehen

Der folgende Beitrag ordnet die Probleme der Lehre vor dem Hintergrund der aktuellen Entwicklungen zu einer Kultur der Digitalität in der heutigen Gesellschaft ein. Dazu werden zunächst die Herausforderungen der Lehre untersucht. Diese werden auf drei Stufen jeweils genauer bestimmt: von der allgemeinen Lehre an Hochschulen über die Einstellungen und Rahmenbedingungen der Studierenden bis hin zur speziellen Lehre der Softwareentwicklung. Die Kultur der Digitalität, wie sie von Felix Stalder beschrieben wird, betrachtet die Mitglieder der Gesellschaft als Nutzer digitaler Materialien und als Teilnehmer an digitalen Gemeinschaften. Dagegen haben Softwareentwickler die Aufgabe, die Grundlagen dieser digitalen Gemeinschaften zu schaffen. Hierdurch verändern sich Entwicklungstätigkeiten, da nun auch Teilnehmer in der Kultur der Digitalität sich durch eigene Beiträge in den Systemen einschreiben müssen und somit Gestaltungsanteile übernehmen. Das Nutzermodell und die Handlungsmöglichkeiten in modernen Softwaresystemen erweitern sich und die Herausforderungen für Entwickler werden damit größer.

Für die Einordnung von Teilnehmern und Entwicklern in möglichen Lehrszenarien werden zwei Perspektiven vorgestellt: Zum einen können Digitalgesellschaften nachgeahmt werden, was zu einem Konzept von Lernplattformen führt. Dieses Konzept ist bekannt und bewährt. Wie es sich zur Kultur der Digitalität verhält und welche Herausforderungen hierin bestehen, wird diskutiert. Zum anderen kann die Kultur der Digitalität mit ihren technischen Voraussetzungen als inhaltliche Vorlage einer Lehre in der Softwareentwicklung gesehen

werden. Bekannte Werkzeuge werden als Vorbild genutzt und in der Konstruktion umgesetzt bzw. sogar erweitert. Diese Praxis motiviert die Lernenden, führt aber auch zu Problemen durch die Komplexität der technischen Anforderungen. In einem Ausblick werden abschließend Möglichkeiten einer Kombination der Lehransätze diskutiert.

2. Analyse der Lehre im Fachgebiet Softwareentwicklung

2.1. Lehre an Hochschulen

Bereits 1998 wurde in Frankreich der Grundstein für eine europaweite Reform der Hochschulstruktur gelegt. Ein Jahr später folgte die namensgebende Bologna-Deklaration, die durch 29 Staaten der EU verabschiedet wurde. Die Bologna-Reform des Hochschulwesens ist charakterisiert durch eine formale Definition der Struktur des Lehrangebots jeden Studiengangs, um eine europaweite Vergleichbarkeit und damit auch Mobilität der Studierenden zu erreichen. Insgesamt soll die Transparenz der in einem Studium erbrachten Leistungen erhöht werden.¹

In der Praxis zeigt sich, dass durch die notwendige Modularisierung der verschiedenen Veranstaltungen Lehrinhalte eher komprimiert werden. Betrachten wir das Studium der Wirtschaftsinformatik am Fachbereich duales Studium der HWR Berlin: Waren zu Zeiten des Diplomabschlusses die Veranstaltungen zu den Grundlagen der Programmierung, Algorithmen und Datenstrukturen, objektorientierte Programmierung sowie Entwurf objektorientierter Systeme noch auf verschiedene Semester verteilt, führt die Modularisierung der Lehrveranstaltungen nun zu einer Zusammenfassung der einzelnen Veranstaltungen zu umfassenderen Modulen, die dann auch in nur weniger Semestern (im Beispiel zwei) stattfinden. Die Kombination sich ergänzender Lehrveranstaltungen in einem Modul wäre zwar denkbar (z. B. Grundlagen der Programmierung mit theoretischen Grundlagen der Informatik), jedoch organisatorisch wesentlich schwieriger umzusetzen, soll es sich dabei nicht nur um ein unabhängiges Nebeneinander der Veranstaltungen in einem Modul handeln. Hinzu kommt, dass die Entwürfe der Studiengänge eher auf ein kürzeres Studium ausgelegt sind.

1 Brändle, 10 Jahre Bologna-Prozess, 2010. Wiesbaden: VS Verlag für Sozialwissenschaften, S. 17 ff.

Somit werden noch mehr Inhalte in weniger zur Verfügung stehender Zeit eingeplant.²

Die Einführung exakter Maßzahlen für auch kleinere Teilleistungen (z. B. Präsenzstunden und zusätzliche Bearbeitungszeiten) macht es dann notwendig, diese Teilleistungen nachweislich zu dokumentieren. Damit entsteht Anwesenheitspflicht für Präsenzanteile des Studiums, die eine eigenständige Gestaltung des Lernprozesses aus der Sicht der Studierenden stark einschränkt. Durch unterschiedliches Vorwissen, Interessen oder auch Auffassungsgabe in den jeweiligen Lernbereichen ist der Lernaufwand aber für Studierende nicht immer gleich groß.

Insgesamt kann die Hochschullehre vor dem Hintergrund des Bologna-Reformprozesses als deutlich komprimierter im Vergleich zu früheren Zeiten eingeschätzt werden. Das Resultat für Studierende ist, insgesamt weniger Zeit zu haben und einen höheren Erfolgsdruck aushalten zu müssen.

2.2. Persönliche Voraussetzungen und Einstellungen

Die wohl wichtigsten Faktoren für einen Studienerfolg sind das *Interesse* und die *Motivation* der Studierenden.³ Bereits bei der Wahl des Studienfachs werden Studierende häufig durch externe Einflüsse geleitet, sodass es hier schon zu Einschränkungen des Interesses gegenüber den Inhalten des Studiengangs kommen kann. So werden in vielen Fällen Studiengänge gewählt, die zukünftig eine hohe Wahrscheinlichkeit für eine berufliche Anstellung bieten. Gerade die Wirtschaftsinformatik ist hierfür ein Beispiel. Im Bereich der dualen Studiengänge suchen dann auch Ausbildungsunternehmen Studierende für diese vielversprechenden Studiengänge. Liegen allerdings mehr Bewerbungen für andere Angebote vor, werden Bewerber schnell auf diese wirtschaftlich interessanteren Abschlüsse umgeleitet.

Aus der Menge unterschiedlicher Studieneinsteiger entstehen so in den ersten Semestern Studiengruppen, deren Teilnehmer ein entsprechend unterschiedliches *Vorwissen* aufweisen. In den Fächern, die bereits heute Eingang in den Schulunterricht finden, wie z. B. Programmierung oder Betriebswirtschaft, zeigen sich dann die Unterschiede im fachlichen Vorwissen besonders stark. Die Existenz umfangreicher Literatur im Sinne eines Studiengangsvorkurses bestätigt die Problematik auch über alle Studiengänge und Hochschularten hinweg. Selbstverständlich finden sich auch Beispiele aus der Lehrpraxis, bei denen oh-

2 Wiarda, „Das haben wir nicht gewollt – Der Bachelor in sechs Semestern führt zur Überfrachtung des Studiums – es wird Zeit, das zu ändern“, 2007, URL: <https://www.zeit.de/2007/44/C-Bama-Aufmacher/komplettansicht>, abgerufen am 12.06.2019.

3 Ulrich, *Gute Lehre in der Hochschule*, 2016, Wiesbaden: Springer Fachmedien, S. 17.

ne Vorkenntnisse Wissen und Fähigkeiten aufgebaut werden und schließlich zu auch sehr guten Lernergebnissen führen. Bereits vorhandene, aus schulischen Kursen entstandene oder selbstständig erarbeitete Kenntnisse lassen jedoch die Wahrscheinlichkeit steigen, eine gute Bewertung in den betroffenen Lehrveranstaltungen zu erzielen.

Eine weitere, auffällige Entwicklung in der Präsenzlehre betrifft die *Konzentrationsfähigkeit* der Studierenden bei der Vorstellung neuer Inhalte, die sich gegebenenfalls auch durch komplexere Zusammenhänge auszeichnen. Eine weitläufige Erklärung wird durch die heutige Überflutung der Menschen mit multimedialen und relativ kurzen digitalen Inhalten gegeben. Typische Beispiele hierfür sind YouTube-Videos, Facebook-Beiträge und Instagram-Fotos. Im Allgemeinen werden kürzere Lösungen für Probleme (z. B. das Erlernen von Fähigkeiten auf Plattformen wie Udemy) gegenüber länger dauernden Lernprozessen bevorzugt. Gerade in der Softwareentwicklung ist es jedoch fatal, eine schnell erstellte Lösung früh zu akzeptieren. Die Meinung „das funktioniert doch“ kann sich kurzfristig in der Praxis rächen, wenn es zu Überarbeitungen einer Software kommt, die z. B. durch fehlende Modularität nur schwer zu verstehen ist.

2.3. Lehre der Softwareentwicklung

Heutige Softwareentwicklung basiert auf einer *Vielfalt von technischen Plattformen*. Schon die Programmierung in einer grundlegenden Programmiersprache wie Java setzt in der Praxis auf einer großen Anzahl von Bibliotheken und den darin enthaltenen Klassen auf.⁴ Bibliotheksklassen enthalten für Programmierer wichtige Funktionalitäten wie z. B. die Gestaltung grafischer Benutzerschnittstellen. Verlässt man die Java-Infrastruktur und nutzt beispielsweise Javascript-Bibliotheken zur Entwicklung von Web-Anwendungen, so kommt eine zusätzliche Komplexität durch die Kombinationsmöglichkeiten von Bibliotheken vieler verschiedener Anbieter hinzu. Typische Funktionalitäten, die dann jeweils auch durch unterschiedliche Bibliotheken umgesetzt werden, sind die Darstellung spezieller Daten (z. B. in Diagrammen), die Anbindung von Datenbanksystemen und die effiziente Verarbeitung von Serveranfragen (z. B. durch Parallelisierung).⁵ Die Grundlagenausbildung in der Programmierung kann na-

4 In der ersten Java-Version 1.0 fanden sich 212 Klassen, während in der Version 11 über 6000 Klassen zur Verfügung stehen. Zur Entwicklung der Java-Versionen siehe auch *Liguori/Liguori*, Java 8 Pocket Guide, 1st ed., 2014. Sebastopol, California: O'Reilly and Associates.

5 Z. B. zur Konstruktion von Nutzerschnittstellen in Web-Anwendungen in: *Thesmann*, Einführung in das Design multimedialer Webanwendungen, 1st ed., 2010. Vieweg

+Teubner Verlag <https://doi.org/10.5771/9783748905318-15>, am 17.08.2024, 04:17:44

türlich nicht auf dieser großen Anzahl von Bibliotheken für die verschiedenen Problemlösungsbereiche aufsetzen. Hier ist der/die Dozent/-in jeweils gefordert, einen verständlichen Weg zur Auswahl unbedingt notwendiger Bibliotheken und der Reduktion von Komplexität zum besseren Verständnis zu finden.

Bezieht man sich in der Ausbildung auf die Entwicklung praxisgerechter Systeme, so kommt man nicht umhin, *weitergehende Strukturen* zu berücksichtigen, die typische Muster des Aufbaus und der Verarbeitung einer Softwareanwendung beschreiben. Diese Muster werden in der Literatur als Entwurfs- und Architekturmuster bezeichnet und dienen unter anderem der besseren Anpassbarkeit von Softwaresystemen (z. B. an neue Unternehmensprozesse).⁶ Das kann insbesondere durch Modularisierung erreicht werden; hierbei werden verschiedene Teile der Software identifiziert, die sich dann auch im Quellcode in Modulen voneinander abgrenzen. Ein Beispiel ist die Unterteilung in Funktionen, die die Nutzerschnittstelle realisieren, die Verarbeitung von Anfragen durchführen und die Speicherung in persistenten Medien (z. B. mithilfe von Datenbanksystemen) realisieren. Diese Art eines Systemaufbaus wird als Schichtenarchitektur bezeichnet. Die Vorteile des Einsatzes von Mustern werden allerdings auch durch eine höhere Komplexität der entstehenden Software erkauft. Entwickler müssen verstehen, wie die Module aufgebaut sind, und miteinander kommunizieren.

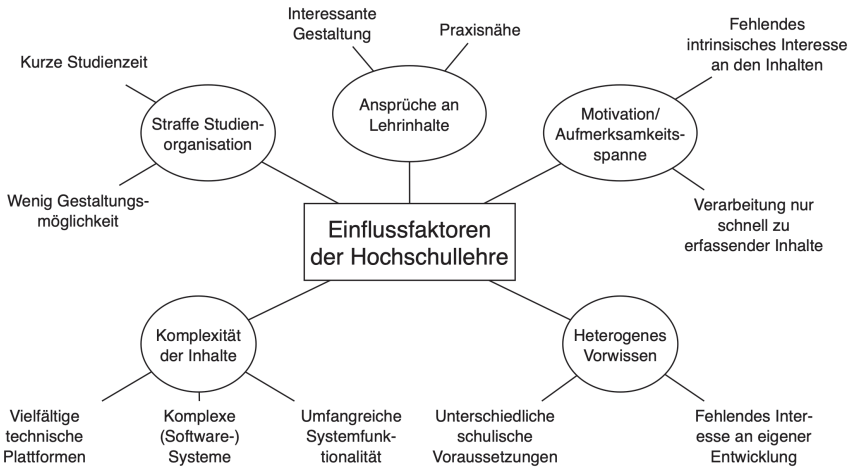
Durch die Entwicklung der Digitalisierung und die Durchdringung des täglichen Lebens mit digitalen Angeboten findet darüber hinaus eine Veränderung der Anforderungen an Entwicklungstätigkeiten statt. Nutzer werden bereits zu Gestaltern von Angeboten z. B. im World Wide Web bzw. auf entsprechenden Plattformen. Daraus resultiert, dass Entwicklungsaufgaben sich mehr auf die Gestaltung generischer Infrastrukturen beziehen, in denen Nutzer entsprechende Möglichkeiten der eigenständigen Gestaltung von Angeboten erhalten. Das erhöht wiederum den Aufwand zum Einstieg in eine Programmierstätigkeit. Wie sich diese Veränderung im Anforderungsprofil der jeweiligen Stakeholder in der Softwarenutzung und -entwicklung auswirkt, wird im folgenden Kapitel auf der Basis Felix Stalders Kultur der Digitalität erörtert.⁷

Abbildung 1 fasst die verschiedenen Einflussfaktoren, die sich auf den oben genannten Ebenen der Lehre ergeben, zusammen.

6 *Gamma/Helm/Johnson/Vlissides*, Design Patterns: Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software, 2014th ed., 2014. Heidelberg, Neckar: mitp.

7 Vergeiche dazu auch *Faustmann*, Standards und Frameworks in der heutigen Softwareentwicklung. In: *Lemke et al.*, Einführung in die Wirtschaftsinformatik, Band 2: Gestalten des digitalen Zeitalters, 2017. Springer Gabler, S. 54–58.

Abbildung 1: Beeinflussende Faktoren der Hochschullehre



3. Rollen vor dem Hintergrund Stalders Kultur der Digitalität

3.1. Teilnehmer

In Felix Stalders Ausführungen zu einer Kultur der Digitalität wird der Mensch in Bezug zur Entwicklung der digitalen Umgebung gesetzt. Kultur sind hier Prozesse, in denen soziale Bedeutung durch Handlungen realisiert wird. Digitalität ist die Menge der Relationen, die Menschen und Objekte auf Basis digitaler Netzwerke verbindet.⁸ Nach Stalder ändert sich der Diskurs bereits seit dem 19. Jahrhundert und hat eine besondere Beschleunigung in den 1960er Jahren erfahren: Es gibt immer mehr Beteiligte am Aushandlungsprozess sozialer Bedeutung. Das ist zum einen bedingt durch das Wachstum der Wissensökonomie, die durch flache Hierarchien, eine hohe Wichtigkeit von Kreativität und den digitalen Raum als Haupthandlungsraum charakterisiert ist.⁹ Darüber hinaus gibt es Tendenzen, die als Postkolonialismus bezeichnet werden und die Begriffe Peripherie und Zentrum aufweichen. Es findet eine Hybridisierung von Autorität statt: Autoritäten können umgedreht werden, es findet eine kulturelle Auseinandersetzung auch aus der bisherigen Peripherie statt, die einhergeht mit

8 Stalder, Kultur der Digitalität. Originalausgabe, 2016. Berlin: Suhrkamp Verlag, S. 16 ff.

9 Ebenda, S. 24 ff. <https://doi.org/10.5771/9783748905318-15>, am 17.08.2024, 04:17:44

einer zentralen Bedeutung des Aushandelns.¹⁰ Somit schaffen die Entwicklungen hin zu einer Kultur der Digitalität nicht nur eine breitere Basis der Beteiligung, sondern sie vervielfältigen auch die kulturellen Möglichkeiten.

Stalder identifiziert in dieser allgemeinen Strömung noch die wichtigsten kulturellen Beteiligungsformen der Digitalität. Zunächst können sich Einzelne durch den Einsatz von Material und dessen Verknüpfung in kulturellen Prozessen verankern (*Referentialität*). Dieser Prozess ist durch die historische Entwicklung, z. B. den Buchdruck, von Digitalisierungsprojekten und der massenhaften Produktion digitalen Materials durch Teilnehmer entsprechender digitaler Plattformen gekennzeichnet. Es kommt zu einer Informationsflut, die zunächst Unordnung bedeutet, aber einen subjektiven Bedeutungszusammenhang ermöglicht. Diese durch den Einzelnen erzeugte Ordnung bedarf allerdings einer subjektiven Auswahl, die beim Produzenten zu einer erhöhten Anstrengung beim „Buhlen“ um Aufmerksamkeit führt.¹¹

Eine Kultur der Digitalität vor der rasanten Entwicklung des Internets (und des World Wide Webs) als Massenmedium kann nur durch die Erzeugung von *Gemeinschaftlichkeit* entstehen. Durch die hohe Reichweite der digitalen Technologien können sich die Individuen in der Gemeinschaft konstituieren. Das ist jedoch auch die elementare Anforderung an jeden Teilnehmer, der sich in den sozialen Netzen identifizieren will: das Leisten von Beiträgen und die Anerkennung durch positives Feedback. Aufmerksamkeit wird zur wichtigsten Ressource in den Netzwerken.¹² Die freiwillige Teilnahme muss aber differenziert betrachtet werden: Es müssen Normen wie Protokolle akzeptiert werden, und abhängig von der jeweiligen Gemeinschaft herrschen subtile Codes vor, die das Feedback anderer Teilnehmer entscheidend beeinflussen können.¹³

Schließlich ist die Teilnahme an digitalen Netzwerken geprägt durch eine *Algorithmizität*, die den Teilnehmern an verschiedenen Stellen erst ermöglicht, Handlungen auszuführen. Technologisch betrachtet findet man in den Netzen eine sehr große Ansammlung von Daten („Big Data“), die nur durch automatisierte Verfahren für den Menschen überblickt werden kann. Die Entwicklung schreitet auch in diesem Bereich voran. War die Automatisierung früher eher auf den Bereich der Suche beschränkt, so können Algorithmen heute schon für kreative Aufgaben (z. B. Sprach- und Textverstehen) eingesetzt werden. Ebenso können Algorithmen sich an Situationen anpassen und sich auf das Individuum

10 Ebenda, S. 50 ff.

11 Stalder, Kultur der Digitalität. Originalausgabe, 2016. Berlin: Suhrkamp Verlag, S. 102 ff.

12 Ebenda, S. 129 ff.

13 So gibt Stalder als Beispiel die Communitys der Programmierer, in denen sprachliche Regeln herrschen und in denen sich z. B. Einsteiger schnell ins Abseits bewegen können (ebenda, S. 147).

einstellen (Beispiel Handschrifterkennung oder auch präferierte Einkäufe). Grundlegend für diese Individualisierungen von Algorithmen sind umfangreiche Personenprofile, die ebenso mit der Zeit anwachsen.¹⁴

3.2. Gestalter

Durch das Referenzieren von Materialien und die Notwendigkeit, sich in den Gemeinschaften zu konstituieren, werden die Teilnehmer der Kultur der Digitalität bereits zu Gestaltern. Sie erschaffen Beiträge und Werte, die für sich und in der resultierenden Interaktion Kultur realisieren. Das kann bereits zu eigenen Angeboten im World Wide Web führen, die entweder durch benutzerfreundliche Konstruktionsanwendungen gebaut oder direkt in HTML erstellt werden. Erste Schritte für einfache statische Darstellungen bzw. unkomplizierte dynamische Seitenbestandteile können schnell erlernt werden.

Geht es um die Schaffung von Infrastrukturen als Basis der Kultur der Digitalität, z. B. die Entwicklung einer neuen Plattform für eine spezifische Interessengruppe, so findet ein Übergang zur Entwicklungstätigkeit statt. Hierbei geht es um die Schaffung generischer Lösungen, die durch Konfiguration verschiedene Ausprägungen einnehmen können. Hinzu kommt die Frage nach der Erweiterbarkeit in Hinblick auf Ausführungskapazitäten wie Bearbeitungseffizienz (sichtbar im Antwortzeitverhalten) oder auch Datenkapazitäten (z. B. die maximale Anzahl von Benutzerkonten). Schließlich muss auch der zukünftige Nutzer in seiner Rolle als Gestalter in einer Neuentwicklung berücksichtigt werden. Wie kann beispielsweise ein Nutzerkonto eingerichtet werden oder das System auf verschiedene Server verteilt werden? Hier sind unterschiedliche Ansätze denkbar, die von geführter Konfiguration in einer Administrationschnittstelle über die Anpassung von Konfigurationsdaten in Dateien bis hin zur Definition von Prozessen mithilfe von Skriptsprachen reichen.

4. Das Konzept der Plattform als Lernumgebung

Lernplattformen werden heutzutage als Standard zur Unterstützung der Präsenzlehre eingesetzt und sollen den Onlineanteil der Lehre ausmachen. Es zeigen sich verschiedene Parallelen zur Kultur der Digitalität Stalders, aber auch Herausforderungen, die ebenso durch die allgemeinen Entwicklungen einer gesellschaftlichen Digitalität erklärt werden können.

14 Ebenda, S. 164 ff. <https://doi.org/10.5771/9783748905318-15>, am 17.08.2024, 04:17:44

4.1. Prinzipien von Lernplattformen

Lernplattformen (auch als Lernmanagementsystem LMS bezeichnet) wie Moodle, ILIAS oder Blackboard ermöglichen es, digitale Inhalte für Lehrveranstaltungen zu bündeln sowie die Kommunikation der Teilnehmer zu unterstützen. Teilnehmer erhalten ein Nutzerkonto, es kann zwischen den verschiedenen Lehrveranstaltungen mit ihren jeweiligen Inhalten unterschieden werden, und eine Rechteverwaltung regelt den Zugriff auf die Dokumente und Veranstaltungen. Hinzu kommen, meist modular organisiert, Werkzeuge zur Förderung der Kommunikation (z. B. Foren, Chats, Whiteboard), zur individuellen Unterstützung (Notizbuch, Kalender) und zur Leistungsüberprüfung (Quiz, Test). Die Inhalte können über einen Web-Browser erreicht werden und sind somit ortsunabhängig verfügbar.¹⁵ Für Dozentinnen und Dozenten stehen meist Autorenwerkzeuge zur Verfügung, mit denen Materialien sehr einfach erstellt, angepasst und auch gefunden werden können.¹⁶

In diesem Konzept zeigen sich die angesprochenen Formen einer Kultur der Digitalität: Teilnehmer von Veranstaltungen nutzen die zur Verfügung gestellten Inhalte für eigene Lernprozesse, sind aber auch angehalten, neue Materialien zu erzeugen, die als Ergebnis des Lernens z. B. von Fähigkeiten entstehen. Dabei können vorhandene Quellen genutzt und kombiniert werden (*Referentialität*). Als Beispiel sei hier die Entwicklung eines Programmcodes genannt, der als Lösung in eine von der Dozentin auf der Plattform eingestellte Aufgabe eingesetzt wird. Andere Teilnehmer können gegebenenfalls diese Lösung kommentieren und darauf aufbauend Verbesserungen vorschlagen.

Durch die Zugehörigkeit des Nutzers eines LMS zu einer Lehrveranstaltung entsteht eine Gemeinschaft. Durch die Kommunikation und die Interaktion über die auf dem LMS und speziell in dem Kurs verwendeten Werkzeuge entsteht *Gemeinschaftlichkeit*. Denkbar für das oben genannte Beispiel einer Programmierstellung ist die gemeinsame Entwicklung größerer Programme, die in Teilaufgaben unterteilt sind. Eine besondere Herausforderung, wie in der Entwicklungspraxis üblich, stellt hierbei die notwendige Integration der Codeanteile dar. Ähnliche Ansätze können für andere Prozessabschnitte der Softwareentwicklung, wie Anforderungsdefinition, Entwurf und Test, durchgeführt werden. Einfache Diskussionen können auch durch ein Forum realisiert werden.

Eine automatisierte Unterstützung der Teilnehmer im Sinne einer *Algorithmizität* hat sich im Bereich der Lernplattformen noch nicht durchgesetzt. Zwar finden sich inzwischen funktionale Erweiterungen, wie z. B. Dashboards, die

15 <https://www.e-teaching.org/technik/distribution/lernmanagementsysteme>.

16 Faustmann/Lemke/ Kirchner, „Und was lehrst Du?“ – Entwurf einer digitalen Plattform für Lehrende, in: Tagungsband der 31. AKWI-Jahrestagung, Heide, 2018. S. 72–82.

einen Überblick zur Materialnutzung durch die Teilnehmer, bevorzugte Onlinezeiten, Forumsaktivitäten u. v. m. bieten, aber eine direkte Unterstützung des Lerners zur Durchführung eines individuellen Lernprozesses, bei dem sich Fragen nach den besten Lerninhalten, Reihenfolgen und Aufgaben stellen, findet so noch nicht statt. Hier spielen auch Fragen des Datenschutzes eine Rolle, da personenbezogene Daten zu verarbeiten sind. Allerdings findet diese Verarbeitung in dem abgeschirmten Kontext des Kurses einer Lernplattform statt und darüber hinaus nur für den Teilnehmer selbst, solange keine persistente Protokollierung der Aktivitäten und der daraus berechneten Empfehlungen durchgeführt wird. Verarbeitungstechnisch könnten sich also auch Ansätze finden, die den Anforderungen des Datenschutzes genügen und eine personalisierte Unterstützung des Lernenden auf einer Lernplattform realisieren.

Durch die starke Verbreitung der Lernplattformen im Bildungsbereich (besonders im Bereich der Hochschulen) und die deutliche Anlehnung an vorhandene Netzwerke und Plattformen des Internets hat sich eine Kultur des digitalen Lernens entwickelt, die sich in der beschriebenen Ausprägung als Quasistandard etabliert hat. Hier wird auch darauf gesetzt, dass Studierende sich schnell aufgrund ihrer bisherigen Erfahrungen auf Lernplattformen zurechtfinden.

4.2. Motivation des Plattformkonzepts

Der Einsatz einer Lernplattform im Rahmen von Studiengängen an Hochschulen setzt besonders auf die Nutzung bereits vorhandener Fähigkeiten der Studierenden im Umgang mit digitalen Materialien und in der digitalen Kommunikation. Darüber hinaus soll auch eine höhere Qualität des Lernprozesses erreicht werden. Im Gegensatz zum einfachen Lesen von Büchern oder Skripten findet insbesondere beim problembasierten Lernen (Problem-based Learning) durch die Kommunikation mit anderen Teilnehmern und die Zusammenarbeit in Projekten (Project-based Learning) eine stärkere Verankerung des Wissens statt.

Häufig wird auch eine Vermischung von Präsenzanteilen mit der Durchführung von Anteilen der Lernprozesse auf Plattformen praktiziert. Im Blended Learning¹⁷ können die jeweiligen Vorteile der Präsenz- und Onlinelernphasen miteinander kombiniert werden: Während in der Präsenzphase eine sehr direkte Kommunikation möglich ist, kann die/der Studierende auf der Lernplattform die für sie/ihn besten Lernmaterialien wählen und im individuellen Tempo verarbeiten. Während also auf der einen Seite Fragen und Probleme sehr schnell geklärt werden können, kann der Teilnehmer auf der anderen Seite die Wissens-

17 https://www.e-teaching.org/lehrszenarien/blended_learning 08.2024, 04:17:44

aufnahme selbst steuern. Ein Problem stellt dann nur die Abstimmung der beiden Phasen dar, da auch die Onlinephasen durch geeignete didaktische Impulse aus der Präsenzlehre getriggert sein müssen.

Plattformen ermöglichen somit nicht nur eine Zusammenfassung von Aktivitäten und Inhalten in Bezug auf einzelne Lehrveranstaltungen, sondern

- sie berücksichtigen die *individuellen Bedürfnisse* der Teilnehmer, indem sie ein durch den jeweiligen Teilnehmer gestaltbares Angebot schaffen, das durch Nutzung von Materialien und Durchführung von Kommunikation geprägt ist,
- sie schaffen einen *vernetzten Treff- und Austauschpunkt*, auf dem Lehrinhalte angeboten und getauscht werden sowie Gemeinschaften (Communitys) entstehen, und
- sie können den *ästhetischen Ansprüchen* an eine moderne Lehre genügen, da ihre Gestaltung auf Standards der Web-Entwicklung basiert und damit an vorhandene Nutzerschnittstellen angepasst werden kann.¹⁸

4.3. Herausforderungen

Der Einsatz von Lernplattformen in der Lehre ist allerdings kein Versprechen für einen Lernerfolg. Untersuchungen zeigen, dass es unterschiedliche Faktoren sind, die von Studierenden als Erfolg gewertet werden.¹⁹ Ganz konkret stellt sich die Frage, warum Teilnehmer auf sozialen Plattformen wie Facebook mehr Zeit verbringen als Studierende auf Lernplattformen wie Moodle. Eigene Erfahrungen an der Hochschule für Wirtschaft und Recht über eine Zeit von ca. 15 Jahren zeigen, dass im Laufe der vergangenen Jahre die Nutzung von Computern im Unterricht in Phasen von eher theoriebasierten Inhalten oft zur Ablenkung durch soziale Plattformen, speziell Facebook, führt. Studien bestätigen auch, dass Studierende in Facebook eine Ablenkung von den eigentlichen Lernzielen, die z. B. in Moodle erreicht werden können, sehen.²⁰ Der Unterschied zwischen Moodle und Facebook wurde von befragten Studierenden besonders in Bezug auf eine Ablenkung vom Lernen unterschiedlich eingeschätzt. Es wurden auch Faktoren identifiziert, warum Facebook der Nutzung von Moodle vor-

18 Vgl. dazu z. B. *Daniela/Rüdolfa*, Learning Platforms: How to Make the Right Choice, in *Didactics of Smart Pedagogy: Smart Pedagogy for Technology Enhanced Learning*, L. Daniela, Ed. Cham: Springer International Publishing, 2019, S. 191–209.

19 So zeigt eine Umfrage an der HWR Berlin, dass Interaktionen, häufige Nutzung und Feedback über die Plattform kein großes Gewicht haben (im Vergleich z. B. zur ständigen Verfügbarkeit der Lernmaterialien). Vgl. *Faustmann/Kirchner/Lemke/Monett*, Which Factors Make Digital Learning Platforms Successful?, 2019. INTED 2019, Valencia.

20 Vgl. *Petrovic et al.*: Facebook vs. Moodle: What do Students really think? ICICTE, 2013.

gezogen wird. Dazu zählen der z. B. der Zwang, Moodle im Rahmen einer Lehrveranstaltung zu nutzen, die Kommunikation unter Freunden bei Facebook und die Verschiedenheit der Nutzerschnittstellen.²¹

Ein erster technisch-organisatorischer Kritikpunkt ist, dass Lernplattformen eher als Datenablage und Verwaltungsplattform eingesetzt werden, auf denen man sich alle notwendigen Informationen zu einer Lehrveranstaltung zusammensucht. Dazu zählen die prüfungsrelevanten Lerninhalte (Folien, Skripte, Buchkapitel), prüfungsvorbereitende Materialien (Musterklausuren, Beispiellösungen für Übungsaufgaben), aber auch Nachrichten zu den Inhalten (was wird von einer Prüfung ausgeschlossen) oder zur Veranstaltungsorganisation (wann fällt ein Termin aus oder findet ein Ersatztermin statt). Weder werden aus dieser Perspektive aber eigene Inhalte aus Sicht der Studierenden erzeugt noch wird so eine Gemeinschaft entstehen, die auf bilateraler Kommunikation aufbaut.

Werden Werkzeuge eingesetzt, die eine Kommunikation initiieren können, treffen sie häufig auf wenig Resonanz bei den Studierenden, Beiträge zu leisten. Weitere Studien zeigen, dass eine starke Integration in den Kursverlauf sowie die Motivation durch die Kursleiter eine weitergehende Teilnahme hervorrufen können.²² Auch durch eine starke Kopplung an bewertete Leistungen kann das studentische Engagement erhöht werden.²³ Eine Erklärung für diesen Zusammenhang kann in der Entwicklung der Gemeinschaftlichkeit innerhalb der Kultur der Digitalität liegen: Gemeinschaft entsteht dort durch die Freizügigkeit der Zugehörigkeit sowie die Entwicklung selbst definierter Regeln und Werte. Auch wenn die Gemeinschaftlichkeit in der Kultur oft auch durch einen gewissen Druck erzeugt wird, so ist dieser gesellschaftliche Druck nicht mit dem Zwang des Einschreibens in einen Plattformkurs während der Teilnahme an einer Lehrveranstaltung vergleichbar. Die Studierenden werden gezwungen, sich als Teilnehmer einer Lehrveranstaltung einzuschreiben, und haben somit auch kein intrinsisches Bedürfnis, sich in der Lehrveranstaltungsgemeinschaft zu konstituieren.

Ein weiteres großes Problem, das die digitale Kommunikation (aber auch die Kommunikation in Präsenzveranstaltungen) behindert, ist das heterogene Vorwissen der Teilnehmer bezüglich der Lehrveranstaltungsinhalte. Besonders tritt dieses Phänomen zu Tage, wenn Fähigkeiten vermittelt werden sollen. Bei-

21 *Guileva*, Moodle vs. Social Media Platforms: Competing for Space and Time. Paper presented at Conferinta „Bunele Practici de Instruire Online“, Chisinau, Moldova, Rep. of., 2014.

22 *Ezeah*, Analysis of Factors Affecting Learner Participation in Asynchronous Online Discussion Forum in Higher Education Institutions. *IOSR Journal of Research & Method in Education*, Vol. 4, Issue 5, 2014.

23 So kann man beispielsweise eine Mindestanzahl an Beiträgen in einem Forum einfordern oder auch einen Vorschlag für einen (Teil-)Entwurf als Dokument.

spiele hierfür sind Programmierkenntnisse oder auch das Lösen mathematischer Fragestellungen. Einsteiger sehen sich oft versierten Studierenden gegenüber und wechseln dann in einen Modus der Beobachtung: Aufgabenstellungen und Lösungen werden zwar betrachtet und vermeintlich verstanden, eigene Lösungsversuche erfolgen jedoch nur rudimentär oder auch gar nicht. Im E-Learning finden sich hierfür verschiedene Arten von Lösungsansätzen, die den Prozess der Wissensvermittlung anders gestalten.^{24, 25} Generell stellt sich hier natürlich die Frage, wie Studierende für Aktivitäten auf Lernplattformen motiviert werden können und welche Parallelen zu sozialen Plattformen gezogen werden können.

Untersucht man Motivationsfaktoren zur Nutzung digitaler Lehrangebote oder zum Engagement auf digitalen Plattformen genauer, so wird deutlich, dass die Akzeptanz sehr stark von den äußeren Rahmenbedingungen abhängt. So ist bei einem Präsenzstudium mit vielen Vor-Ort-Terminen die Motivation deutlich geringer, zusätzlich auch in einer virtuellen Gemeinschaft zu kommunizieren und zusammenzuarbeiten.²⁶ Auch die ausdauernde Teilnahme auf sehr großen und anonymen Plattformen (Massive Open Online Courses – MOOCs) ist eher schwierig und verlangt von den Teilnehmern ein hohes Maß an Motivation.²⁷

Insgesamt betrachtet fehlt die Entwicklungsmöglichkeit des Einzelnen als Teilnehmer in einer Kultur der Lehre im Rahmen der meist nur sehr kurzen Lehreinheiten. Im Laufe eines Studiums akzeptieren Studierende dann eher die Anforderungen und die notwendigen Beiträge bei der Nutzung von Lernplattformen und weitergehenden Lehrkonzepten. Hinzu kommt die starke Abhängigkeit der Erwartungen und der individuellen Motivation von den Rahmenbedingungen der Hochschule bzw. auch des einzelnen Teilnehmers.

24 Ein Beispiel ist die Unterstützung projektbasierten Lernens durch transparente Teilergebnisse, die von den Teilnehmern in Argumentnetzen zu erklären sind (*Faustmann*, Project-based Collaborative Learning with Arguments – Flexible Integration of Digital Resources. Presented at the 3rd International Conference on Computer-Supported Education, Noordwijkerhout, The Netherlands, 2011, vol. 2, S. 359–364).

25 Ein anderer Ansatz ist die Aufgabenstellung, am Lehrstoff orientierte Fragen durch die Studierenden entwickeln zu lassen und mithilfe der Plattform durch die anderen Teilnehmer bewerten zu lassen (*Bates/Galloway/McBride*, Student-generated content: Using PeerWise to enhance engagement and outcomes in introductory physics courses, in: AIP Conference Proceedings, 2012, vol. 1413, S. 123–126).

26 *Faustmann/Kirchner/Lemke/Monett*, Which Factors Make Digital Learning Platforms Successful?, 2019. INTED2019, Valencia.

27 *Conole*, MOOCs as disruptive technologies: strategies for enhancing the learner experience and quality of MOOCs, *RED. Revista de Educación a Distancia*, vol. 50, no. 2, 2016.

5. Das Konzept der gestalterischen Softwareentwicklung

Der folgende Ansatz benutzt die in der Kultur der Digitalität eingesetzten Interaktionsformen als Basis einer Lehre in der Softwareentwicklung. Ziel ist die Steigerung der Motivation der Studierenden durch das Erkennen des eigenen digitalen Weltbilds. Hinderlich dabei ist die Komplexität der notwendigen technischen Komponenten, die Interaktionen realisieren. Allerdings wird sich in Zukunft die Rolle der Softwareentwickler an diesen Herausforderungen messen müssen, da gestalterische Anteile immer mehr zu den Teilnehmern wandern werden.

5.1. Kultur der Digitalität als inhaltliche Vorlage

Das Erlernen von Programmierfertigkeiten wird von vielen Einsteigern als eher trockene Tätigkeit empfunden, da es, wie beim Erlernen jeder Sprache, zunächst auf den Aufbau eines gewissen Wortschatzes und die Umsetzung einfacher Beispielprogramme (also Sätze in der jeweiligen Programmiersprache) ankommt. Es finden sich verschiedene Konzepte, wie diese ersten Schritte motivierender gestaltet werden können, die letztlich alle darauf basieren, ein direktes Feedback der Programmerstellung zu erzeugen und mit sehr einfachen Beschreibungsmitteln zum Ergebnis zu kommen. Das Feedback kann z. B. durch grafische Anteile der Programmumsetzung gegeben sein (in der Sprache Logo ist eine Schildkröte über den Bildschirm zu steuern, die Linien hinterlässt), das Ziel haben, multimediale Spiele zu realisieren (in Scratch ist es einfach, Grafiken und Interaktionen umzusetzen) oder auch auf reale Objekte wirken (Lego Mindstorms bietet eine grafische Entwicklungsumgebung, in der Steuerungselemente des Lego-Roboters wie Bausteine zusammengesetzt werden können). Beschreibungselemente für Befehle werden hierbei oft durch grafische Elemente (z. B. Blöcke, die ineinandergesteckt werden) umgesetzt.

Vor dem Hintergrund der Entwicklung der Digitalität in der Gesellschaft wäre dementsprechend ein motivierender Ansatz des Programmierenlernens, mit relativ einfachen Beschreibungsmitteln bereits bekannte Konzepte im Umgang mit digitalen Artefakten zu erstellen. Mit den Interaktionsformen Referentialität, Gemeinschaftlichkeit und Algorithmizität entstehen die folgenden Anforderungen:

- die Betrachtung und Verarbeitung von digitalen Objekten. Zu solchen Objekten können Texte (formatiert und unformatiert), Bilder, Videos und allgemeine Datenobjekte (z. B. Beschreibungen von realen Objekten mit seinen Metadaten) gezählt werden. Mit dem Ziel der Referenzierung geht es also um Darstellung, Verknüpfung und Veränderung dieser Objekte;

- die Kommunikation und das Zusammenwirken von Teilnehmern auf digitalen Plattformen. Grundlegende Funktionen hierfür sind die Verwaltung von Nutzerkonten, Ansätze zur Autorisierung von Nutzern bzw. zum Schutz der digitalen Objekte, die Verarbeitung vieler, auch gleichzeitiger, Anfragen sowie allgemein die Verwaltung von Transaktionen auf den Objekten (d. h. das konsistente Zusammenspiel der Anfragen und deren Verarbeitung auf den Daten);
- die automatisierte Verarbeitung großer Datenmengen. Hier stellen sich nicht nur Fragen nach der Effizienz (sichtbar für den Nutzer besonders im Antwortzeitverhalten), sondern auch nach Methoden und Möglichkeiten zur Integration in digitale Plattformen (wer kann wie Auswertungen formulieren und auch nutzen).

In der eigenen Erfahrung zeigt sich diese unterschiedliche Motivation in verschiedenen Lehrveranstaltungen. Während in den Grundlagen der Programmierung (erstes Semester Wirtschaftsinformatik) eher auf Befehlsebene die Wirkung von (hier Java-)Programmen besprochen wird und einfache Algorithmen mit eingeschränkter Funktionalität entworfen werden, steigt die Motivation bei der folgenden Lehrveranstaltung des zweiten Semesters Objektorientierte Programmierung, in der als Aufgabenstellung der Entwurf und die Entwicklung eines vollständigen Informationssystems stehen. Schließlich findet sich im letzten, dem sechsten Semester des WI-Studiums noch die Wahlpflichtveranstaltung Web-Programmierung, in der mit großem Engagement umfangreiche webbasierte Systeme entwickelt werden, die oftmals verschiedene der oben genannten Anforderungen für Plattformen umsetzen müssen.²⁸

5.2. Komplexität der technischen Anforderungen

Die Betrachtung realer Anwendungsbeispiele unterliegt allerdings besonderen Herausforderungen: Zunächst bedarf es eines Grundverständnisses der Problemstellungen und der grundsätzlichen Lösungsansätze.²⁹ Zum anderen verkompliziert eine technische Umsetzung den Zusammenhang noch durch den

28 Zu berücksichtigen hierbei ist, dass sich auch die Studierendengruppe, besonders im Wahlpflichtfach, anhand der Interessen und des Vorwissens zusammensetzt. Es finden sich aber auch immer Einsteiger in die fortgeschrittenen Technologien, die mit hoher Motivation und beeindruckenden Ergebnissen die Veranstaltung absolvieren.

29 Es muss beispielsweise für den Studierenden klar sein, welche Zustandsänderungen im System zu Inkonsistenzen führen können, wie die Eigenschaften von Transaktionen solche ungewollten Zustände vermeiden können und mit welchen technischen Ansätzen dann Transaktionen auch tatsächlich umgesetzt werden können.

Beschreibungsformalismus (in Form von Programmiersprachen) und etwaige versteckte Funktionalitäten.

Sind Programmiersprachen bzw. die für bestimmte Funktionalitäten eingesetzten Bibliotheken verständlich entworfen, so kann auch mit geringem Aufwand ein funktional reichhaltiges Ergebnis erzielt werden. Ein Beispiel zeigt Abbildung 2, in der ein Server mit der RMI-Bibliothek (Remote Method Invocation) definiert wird. Die innerhalb des Servers definierte `sayHello`-Methode kann von anderen Java-Programmen, die nicht auf demselben physischen Rechner laufen, aufgerufen werden. Damit das funktioniert, wird in dem Beispiel der Server bereits als `RemoteObject` definiert (Zeile 2), ein für alle Programme verfügbarer Namensdienst gestartet (Zeilen 7/8) und der Server dort mit einem eigenständigen Namen eingetragen (Zeilen 9/10). Die Infrastruktur (Verwaltung der Server, Erreichbarkeit über Servernamen) sowie bestimmte Servereigenschaften (Protokollierung, Parallelisierung etc.) sind bereits in der Bibliothek implementiert.

Abbildung 2: Definition eines Servers in der RMI-Bibliothek

```

1  public class RMI-Server
2      extends UnicastRemoteObject implements MyInterface{
3      public String sayHello(){
4          return "hello";
5      }
6      public static void main(String[] args){
7          LocateRegistry.
8              createRegistry(Registry.REGISTRY_PORT);
9          Registry reg = LocateRegistry.getRegistry();
10         reg.rebind("MyServer", new RMI-Server());
11     } }

```

Dieses Beispiel kann auch auf weniger komplexe Anforderungen übertragen werden. So gibt es in der Programmiersprache Java und ihren bereits mitgelieferten Bibliotheksklassen umfangreiche Funktionalitäten zur Verarbeitung von Zeichenketten (Länge, Zugriff auf einzelne Zeichen, Umwandlungen etc.), Da-

tumsobjekten, Ausgabeformatierungen u. v. m. Der Begriff des abstrakten Datentyps beinhaltet genau diese Kapselung elementarer Daten.³⁰

5.3. *Zukünftige Entwicklung der Rollen*

An dieser Stelle wird deutlich, dass die Grenzen zwischen Nutzern digitaler Angebote und den Gestaltern ebendieser in Zukunft noch mehr verschwimmen werden. An den Rändern des Kontinuums wird es wahrscheinlich immer die jeweiligen Extreme an Rollen geben: die Nutzer, die nur konsumieren und nie eigene Beiträge, geschweige denn komplexere multimediale oder algorithmische Inhalte einstellen. Und auf der anderen Seite die Entwickler, die auf Basistechnologien aufbauen, damit Fragen der Effizienz und der Maschinenabhängigkeit selbst lösen und als Anbieter von standardisierten und konfigurierbaren Lösungen für andere Entwickler fungieren.

Die Komplexität von Anforderungen wird sich nicht unbedingt in einer vergleichbaren Komplexität der Lösungen widerspiegeln, da Bibliotheken und Rahmenwerke bereits vieles vorimplementieren, was notwendig ist. Die Einfachheit der Umsetzung komplexer Systeme wird sich dann an der Nutzerschnittstelle der Entwicklungswerkzeuge messen. Die Entwicklergemeinde hat auch darauf schon reagiert und bietet verschiedene Ausbaustufen von vorgefertigten Softwareanteilen an: von in den Programmiersprachen integrierten Bibliotheken, die über entsprechende Quellcodes genutzt werden können, über Frameworks, die den Aufbau größerer Systeme erleichtern, bis hin zu Standardsoftware, die durch Konfigurationsmöglichkeiten an die unterschiedlichsten Einsatzbedingungen angepasst werden kann (z. B. das typische Entwicklungs-vorgehen bei Unternehmenssoftware). Selbst in technischen und algorithmischen Problemstellungen unerfahrene Nutzer können heute Anpassungen an der Nutzerschnittstelle ihrer Officesysteme vornehmen bzw. sogar typische Abläufe als Makrosequenzen interaktiv aufzeichnen und dann als Funktion per Knopfdruck immer wieder verwenden.

6. Zusammenfassung und Ausblick

Es wurde deutlich, dass die heutige Hochschullehre im Allgemeinen, aber besonders auch mit dem Ziel der Entwicklung bestimmter Fertigkeiten wie der Programmierung von Softwaresystemen mit verschiedenen Problemen umge-

30 Beispiele für abstrakte Datentypen sind auch Warteschlangen, Stapel und Listenobjekte, in denen andere Objekte gespeichert werden können, die jedoch nur bestimmte Arten des Zugriffs auf diese Objekte erlauben, z. B. nur von einer Seite.

hen muss. Da die moderne Gesellschaft nicht nur mit digitalen Inhalten durchdrungen ist, sondern auch eine eigene Kultur auf der Basis von digitalen Inhalten und Plattformen entwickelt hat, ist die Frage nach dem Bezug der Programmierausbildung zu diesen gesellschaftlichen Entwicklungen sinnvoll.

Festgehalten werden kann, dass sich die Rollen der Beteiligten in Bezug auf die Digitalität verändern: Die Notwendigkeit, sich in einer digitalen Gemeinschaft mit einem eigenen Auftreten zu behaupten und durch Referentialisierung bereits vorhandene Materialien einzusetzen und zu kombinieren, erfordert erweiterte Kenntnisse im Umgang mit Datenobjekten und digitalen Plattformen. Diese nutzerorientierte Gestaltungssicht wird durch einfache Schnittstellen erst möglich. Softwareentwickler müssen dann hinter diese Nutzerschnittstellen sehen und Anpassungen oder auch ganze Neuentwicklungen durchführen können. Letztlich geht es in der Lehre der Softwareentwicklung um die Motivation der Studierenden, sich mit Entwicklungstätigkeiten auf verschiedenen Ebenen auseinanderzusetzen.

Das Konzept der Lernplattformen in der Hochschullehre trägt den Entwicklungen der Kultur der Digitalität durchaus Rechnung. Hier geht es um die Motivation der Studierenden zur Beteiligung an Gemeinschaften. Stalder zeigt auf, woran der Aufbau von Gemeinschaftlichkeit scheitern kann. Es geht um Akzeptanz von (eher formal-technischen) Normen, dem Verstehen kultureller Codes und einer unter allem liegenden freiheitlichen Entscheidung der Zugehörigkeit. An dieser Stelle werden schon die Schwierigkeiten der Digitalität in der Lehre deutlich: Lernplattformen als Datenablage lassen keine Gemeinschaftlichkeit im Sinne einer Kultur entstehen, da Beiträge der Teilnehmer sowie die Aufnahme und Bewertung durch andere Teilnehmer schlichtweg fehlen. Auch ist ein freiwilliger Beitritt zu Plattformen als Basis der Entwicklung einer Gemeinschaft in Lehrszenarien so nicht gegeben, was unter anderem durch die vordefinierten Stundenpläne der reformierten Studiengänge bedingt ist. Ein sehr heterogenes Vorwissen der Teilnehmer behindert zusätzlich die Entstehung von Gemeinschaftlichkeit, da keine Kommunikation zustande kommt.

Wie kann nun mit diesen Erkenntnissen die Lehre allgemein, aber auch speziell in der Programmierausbildung verbessert werden?

Die **Motivation der Inhalte** sollte bereits in frühen Phasen der Ausbildung neben Grundkonstrukten an weitergehenden Funktionen und deren Umsetzung mit komplexeren Programmkonzepten (allerdings mit einfachen Nutzerschnittstellen) orientiert sein. Somit kann der Weg nicht nur wie bisher Bottom-up (von einfachen zu komplexen Befehlen bzw. Softwaresystemen), sondern auch Top-down (von komplexen Bibliotheksfunktionen zu grundlegenden Programmiersprachenbefehlen) gegangen werden. Während der erste Weg die Funktionen vor den einfachen Schnittstellen betrachtet, blickt der zweite Weg in die Umsetzung und den inneren Aufbau der Lösungen.

Eine **Kultur des digitalen Lernens** kann durch Blended-Learning-Ansätze im Zusammenspiel mit Lernplattformen erreicht werden, wenn die Hürden zum Aufbau von Gemeinschaftlichkeit beseitigt werden können. Das ist kein leichtes Unterfangen, da bestimmte Vorbedingungen schwer auszugleichen sind (z. B. die schulische Vorbildung).

Durch eine andere formale Gestaltung der Lerninhalte kann hier aber durchaus reagiert werden: Der Art des heutigen Konsums digitaler Informationen sollte mit einer angepassten Unterteilung der Lerninhalte Rechnung getragen werden. Nicht nur verschiedene Medien sollten Einsatz finden, auch die Lerneinheiten selbst sollten eine gewisse Größe nicht überschreiten. Das gilt besonders zu Beginn des Studiums, während in den anschließenden höheren Semestern auch umfangreichere Inhalte und Aufgabenstellungen (z. B. Projektarbeiten über ein oder auch mehrere Semester) angeboten werden dürfen.

Die Verwendung bereits vorhandener Materialien zeigt nicht nur mögliche Lösungswege auf, sondern animiert auch zur Wiederverwendung und somit zum Lernen durch Nachahmen. Gerade in der Softwareentwicklung ist das eine gute Strategie, die sich sogar in verschiedenen Vorgehensmodellen zur Softwareerstellung findet.³¹ Insgesamt sollte bei der Entwicklung von Online-Lehrangeboten darauf geachtet werden, dass diese in Einklang mit den Rahmenbedingungen sowohl der Studierenden als auch der Lehrorganisation stehen. Nicht immer muss zusätzlich Gemeinschaftlichkeit auf digitalen Plattformen entstehen, wenn bereits Präsenzunterricht eine solche erzeugt.

Literatur

Bates/Galloway/McBride, Student-generated content: Using PeerWise to enhance engagement and outcomes in introductory physics courses, in: AIP Conference Proceedings, 2012, vol. 1413, S. 123–126.

Brändle, 10 Jahre Bologna-Prozess, 2010. Wiesbaden: VS Verlag für Sozialwissenschaften, S. 17 ff.

Conole, MOOCs as disruptive technologies: strategies for enhancing the learner experience and quality of MOOCs, *RED, Revista de Educación a Distancia*, vol. 50, no. 2, 2016.

Daniela/Rüdolfa, Learning Platforms: How to Make the Right Choice, in *Didactics of Smart Pedagogy: Smart Pedagogy for Technology Enhanced Learning*, L. Daniela, Ed. Cham: Springer International Publishing, 2019, S. 191–209.

e-teaching.org, Blended Learning, URL: https://www.e-teaching.org/lehrszenarien/blended_learning, abgerufen am: 12.06.2019.

e-teaching.org, Lernmanagementsysteme, URL: <https://www.e-teaching.org/technik/distribution/learnmanagementsysteme>, abgerufen am 12.06.2019.

31 Z. B. in dem Ansatz des Pair-Programmings, bei dem jeweils zwei Entwickler zugleich an einem Computer/Software entwickeln.

- Ezeah*, Analysis of Factors Affecting Learner Participation in Asynchronous Online Discussion Forum in Higher Education Institutions. IOSR Journal of Research & Method in Education, Vol. 4, Issue 5, 2014.
- Faustmann*, Project-based Collaborative Learning with Arguments – Flexible Integration of Digital Resources. Presented at the 3rd International Conference on Computer-Supported Education, Noordwijkerhout, The Netherlands, 2011, vol. 2, S. 359–364.
- Faustmann*, Standards und Frameworks in der heutigen Softwareentwicklung. In: *Lemke et al.*, Einführung in die Wirtschaftsinformatik, Band 2: Gestalten des digitalen Zeitalters, 2017. Springer Gabler, S. 54–58.
- Faustmann/Kirchner/Lemke/Monett*, Which Factors Make Digital Learning Platforms Successful?, 2019. INTED 2019, Valencia.
- Gamma/Helm/Johnson/Vlissides*, Design Patterns: Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software, 2014. Heidelberg, Neckar: mitp.
- Guileva*, Moodle vs. Social Media Platforms: Competing for Space and Time. Paper presented at Conferinta „Bunele Practici de Instruire Online“, Chisinau, Moldova, Rep. of., 2014.
- Liguori/Liguori*, Java 8 Pocket Guide, 1st ed., 2014. Sebastopol, California: O’Reilly and Associates.
- Petrovic et al.*: Facebook vs. Moodle: What do Students really think? ICICTE, 2013.
- Stalder*, Kultur der Digitalität. Originalausgabe, 2016. Berlin: Suhrkamp Verlag
- Thesmann*, Einführung in das Design multimedialer Webanwendungen, 2010. Vieweg+Teubner Verlag.
- Ulrich*, Gute Lehre in der Hochschule, 2016. Wiesbaden: Springer Fachmedien, S. 17.
- Wiarda*, „Das haben wir nicht gewollt – Der Bachelor in sechs Semestern führt zur Überfrachtung des Studiums – es wird Zeit, das zu ändern“, 2007, URL: <https://www.zeit.de/2007/44/C-Bama-Aufmacher/komplettansicht>, abgerufen am 12.06.2019.

