

Entwicklung einer offenen technischen Infrastruktur für HOOU-Lernarrangements an der TUHH

Axel Dürkop, Technische Universität Hamburg-Harburg, ITBH

28.04.2016

Die Arbeit in und an der Hamburg Open Online University (HOOU) führt auch in der TUHH zu einer Vielzahl von Ideen und neuen kreativen Ansätzen. Prozesse wie „Digitalisierung“ und „Öffnung“ der Hochschulen werden auf unterschiedlichsten Ebenen diskutiert und interpretiert, technische Lösungen vor dem Hintergrund aktueller Trends und Entwicklungen neu beleuchtet und evaluiert. Die Entwicklung der HOOU wirkt somit auch zurück auf die TUHH und hat in den vergangenen Wochen zur Konzeption und Implementierung erster hochschulinterner Lösungen geführt, aus denen Erfahrungen und Ergebnisse zurückfließen sollen in den Gesamtzusammenhang der HOOU. Die ersten Ergebnisse dieser Entwicklung sollen hier vorgestellt werden. Leitend in dieser Phase waren die Grundprinzipien für *Open Educational Resources (OER)*, ein Ansatz der *losen Kopplung* technischer Systeme, vor allem aber die Priorisierung didaktischer Anforderungen vor technischen Möglichkeiten.

1 Technische Anforderungen an OER

Orientiert man sich für die Umsetzung dieses Ansatzes streng an den Prinzipien, die David Wiley einst der OER-Bewegung auf die Fahne geschrieben hat, ergeben sich konkrete Anforderungen an die technische Umsetzung. Wiley hatte - analog zu Richard Stallman, der die Freiheiten im Bezug auf Software in der GPL¹ niedergeschrieben hat - seine „5R“ formuliert. Nutzer_innen von OER haben das Recht zu

- „1. **Verwahren/Vervielfältigen** – das Recht, Kopien des Inhalts anzufertigen, zu besitzen und zu kontrollieren (z.B. Download, Speicherung und Vervielfältigung)
2. **Verwenden** – das Recht, den Inhalt in unterschiedlichen Zusammenhängen einzusetzen (z.B. im Klassenraum, in einer Lerngruppe, auf einer Website, in einem Video)
3. **Verarbeiten** – das Recht, den Inhalt zu bearbeiten, anzupassen, zu verändern oder umzugestalten (z.B. einen Inhalt in eine andere Sprache zu übersetzen)
4. **Vermischen** – das Recht, einen Inhalt im Original oder in einer Bearbeitung mit anderen offenen Inhalten zu verbinden und aus ihnen etwas Neues zu schaffen (z.B. beim Einbauen von Bildern und Musik in ein Video)
5. **Verbreiten** – das Recht, Kopien eines Inhalts mit Anderen zu teilen, im Original oder in eigenen Überarbeitungen (z.B. einem Freund eine Kopie zu geben oder online zu veröffentlichen)“²

Flankierend zu den „5R“ macht Wiley auch noch einen Vorschlag zu den technischen Rahmenbedingungen, die bei der Produktion und Bereitstellung von OER berücksichtigt werden sollten. Sein so genanntes ALMS-Framework lautet wie folgt:

¹ https://en.wikipedia.org/wiki/GNU_General_Public_License

² Die deutsche Übersetzung der „5R“ unter <http://open-educational-resources.de/5rs-auf-deutsch/> basiert auf der englischen Fassung von Wiley unter <http://www.opencontent.org/definition/>

- „1. **Access to Editing Tools:** Is the open content published in a format that can only be revised or remixed using tools that are extremely expensive (e.g., 3DS MAX)? Is the open content published in an exotic format that can only be revised or remixed using tools that run on an obscure or discontinued platform (e.g., OS/2)? Is the open content published in a format that can be revised or remixed using tools that are freely available and run on all major platforms (e.g., OpenOffice)?
2. **Level of Expertise Required:** Is the open content published in a format that requires a significant amount technical expertise to revise or remix (e.g., Blender)? Is the open content published in a format that requires a minimum level of technical expertise to revise or remix (e.g., Word)?
3. **Meaningfully Editable:** Is the open content published in a manner that makes its content essentially impossible to revise or remix (e.g., a scanned image of a handwritten document)? Is the open content published in a manner making its content easy to revise or remix (e.g., a text file)?
4. **Self-Sourced:** Is the format preferred for consuming the open content the same format preferred for revising or remixing the open content (e.g., HTML)? Is the format preferred for consuming the open content different from the format preferred for revising or remixing the open content (e.g. Flash FLA vs SWF)?“³

Nimmt man die Rechte sowie die technischen Anforderungen an OER als strenge Vorgabe für die Konzeption und Implementierung einer technischen Umgebung, ergeben sich Konsequenzen auf der Detailebene und tangieren das Speicherformat für Dateien ebenso wie die Beschaffenheit des Ortes, an dem offene Lehr-Lernmaterialien zur Verfügung gestellt werden. Die verwendete Software ist zu berücksichtigen wie auch die Qualifikation und Motivation der beteiligten Akteure.

Vor dem Hintergrund dieser konzeptionellen Vorgaben, die das Handeln im Kontext der Open-Education-Bewegung⁴ leiten, kristallisieren sich erste Erfahrungen in der Early-Bird-Betreuung an der TUHH heraus.

2 Entkopplung von Inhalt und Struktur

An der TUHH hat sich die Auffassung gefestigt, dass eine Trennung eines Lehr-Lernarrangements in *strukturierende* und ein *inhaltliche* Einheiten sinnvoll ist. Diese Aufteilung bietet den Vorteil, Inhalte offen und unabhängig von einem chronologischen Ablauf zugänglich zu machen und so ihre Teilbarkeit, Weiternutzung und Rekontextualisierung zu verbessern. Für die Praxis bedeutet das, dass Interessierte sich auch jenseits eines strukturierten Ablaufs durch eigene Fragestellungen und Ideen mit einer Lernresource auseinandersetzen und diese ggf. weiterentwickeln können. Wikis und Foren sind hier als Beispiele für Werkzeuge zu nennen, um Orte dieser Art zu schaffen. Im besten Fall entsteht eine *learning community*, die viele Interessierte anzieht, weil sie hohe Freiheitsgrade in der Situierung zulässt⁵. Aber auch neuere Formate und Werkzeuge für die Bereitstellung und kollaborative Weiterentwicklung von Inhalten bieten sich an, wie im folgenden noch gezeigt werden wird.

Einige inspirierende Beispiele für die Trennung von Inhalt und Struktur finden sich bei der P2PU, so z.B. die *learning community* „Play With Your Music“⁶. Im Zentrum der Lernaktivitäten liegt hier ein modernes Forum auf der Basis der freien Software *Discourse*⁷, das die kooperative bzw. kollaborative Auseinandersetzung der Lernenden mit den Inhalten einfordert. Vorgeschaltet ist eine Webseite, die Interessierten die Wahl lässt: Alleine lernen oder in der Gruppe mit anderen? Kommen genügend Interessierte zusammen, wird die Kohorte strukturiert durch die Inhalte geführt⁸. Diese Idee wird mit einem *Mechanical MOOCs* der P2PU⁹ umgesetzt.

³ <http://opencontent.org/definition/>

⁴ https://en.wikipedia.org/wiki/Open_education

⁵ Zur Bedeutung von Situierung und Kontextualisierung vgl. Wild & Möller, 2015, S. 7ff. sowie 73f.

⁶ <http://www.playwithyourmusic.org/>

⁷ <https://www.discourse.org/>

⁸ Unter <http://reports.p2pu.org/play-with-your-music/> findet sich eine Reflektion des Projekts.

⁹ <http://reports.p2pu.org/mooc-maker/>

Einen anderen Gedanken bringt das Konzept *Course in a Box* der P2PU¹⁰ in die Diskussion: Ganz im Sinne der OER-Idee sollen Interessierte in die Lage versetzt werden, selbst ein Lernangebot zu erstellen oder vorhandene für eigene Zwecke anzupassen und zu rekontextualisieren. Durch den Ansatz, wie ein *Course in a Box* bereitgestellt wird, wird nicht nur eine Anleitung geliefert, wie man ein solches Lernarrangement aufziehen sollte - die Anleitung selbst ist die softwaretechnische Ausgangsbasis für das eigene Angebot. Auf der Seite *Create Your Course*¹¹ wird beschrieben, was zu tun ist. Der Vorgang verweist auf eine neue Kulturtechnik des Teilens, die sich in der Organisation von Arbeitsprozessen in der Softwareentwicklung etabliert hat - *fork and contribute*. Wie diese Einflüsse für das OER-Konzept und die HOOU nutzbar gemacht werden können, will ich im folgenden ausführen.

3 Entwicklungsschritte zur technischen Umgebung der HOOU an der TUHH

Eine Art Selbstversuch kann als Ausgangspunkt für den im folgenden beschriebenen Schritt zu einem Teilbereich der HOOU-Technik an der TUHH verantwortlich gemacht werden. Infolge der Neukonzeption der Lehrveranstaltung „Einführung in die Informatik I“¹² mangelte es an einem konkreten Skript oder Lehrbuch, das die Studierenden beim Erarbeiten des Veranstaltungsstoffs unterstützen sollte. Schnell war klar, dass der Mangel durch die Produktion eigenen Materials kompensiert werden musste. Dabei entstand die Idee, dieses den angehenden Berufsschullehrer_innen in der Veranstaltung in Form einer OER-Resource zur Verfügung zu stellen, die sie später in der Berufspraxis an eigene Bedürfnisse anpassen und für die eigene Unterrichtsgestaltung nutzen könnten. Geleitet von Wileys Anforderungen und Freiheitsgraden für OER suchte ich nach einer technischen Implementierung, mit der ich die Produktion des Materials umsetzen wollte. Dabei fand ich die Website bzw. Software *GitBook*¹³, die für die Erstellung von Lernressourcen bestens geeignet schien. So entstand online und begleitend zur Veranstaltung das Skript *Python.Processing.Arduino*¹⁴, das sich durch die Kritik und Anregungen der Studierenden und des Tutors während der Produktion oft veränderte¹⁵. Die Studierenden nutzten hauptsächlich die Online-Version, die in einem einfachen HTML-Konstrukt mit Gliederung und Blätterfunktion besteht und die Einbindung von Videos und anderen Medien ermöglicht, luden sich aber zeitweise auch die PDF-Variante herunter¹⁸. Meine Erfahrung mit dem Tool und dem Online-Service waren sehr positiv, es hat mir Spaß gemacht, auf diese Weise ein lebendiges Dokument zu verfassen. Einzig die Tatsache, dass ich hierfür auf einen Service angewiesen war, der hinsichtlich Datenhaltung und Funktionalität nicht meiner Kontrolle unterliegt, hat mich gestört. Wie ich in den folgenden Abschnitten zeigen werde, können diese Abhängigkeiten jedoch durch sinnvolle Kopplung einzelner freier Softwarekomponenten reduziert oder sogar aufgehoben werden.

Die Einfachheit und Klarheit von GitBooks legt noch einen weiteren Einsatzzweck nahe, der die eingangs beschriebene Trennung von Struktur und Inhalt implementiert: Lernarrangements lassen sich ebenfalls in Form eines GitBooks abbilden. Diese sind in der Regel linear aufgebaut und bestehen in ihrer elementarsten Form aus einer Abfolge von Arbeitsanweisungen und Inter-

¹⁰<http://howto.p2pu.org/>

¹¹<http://howto.p2pu.org/modules/start/create-your-course/>

¹²<http://www.itbh-hh.de/de/studium/veranstaltungen/wise-2015-16/einfuehrung-in-die-informatik-i-uebungen-2015-16>

¹³<http://gitbook.com>

¹⁴<http://www.python-processing-arduino.de/>

¹⁵Bei Software sagt man: „Release early, release often“. Dass diese Haltung mittlerweile auch für Bücher gelten kann, zeigen die Verlage O'Reilly (Early Access)¹⁶ und Manning (MEAP-Programm)¹⁷ mit ihrer transparenten Entwicklung neuer Titel.

¹⁸<https://www.gitbook.com/book/xldrkp/pyprocard/details>

aktionsangeboten (vgl. hierzu nochmals den Aufbau von *Course in a Box*¹⁹)²⁰. Folglich ergibt sich mit dem Einsatz von GitBooks in der HOOU ein doppelter Verwendungszweck und damit eine begriffliche Unterscheidung:

- **Content-GitBooks** sind linear aufgebaute multimediale HTML-Seiten, mit denen Informationen aufbereitet und ähnlich einem Lehrbuch zur Verfügung gestellt werden können.
- **Lernarrangement-GitBooks (LA-GitBooks)** sind linear aufgebaute HTML-Seiten, mit denen ein Lernarrangement in Form eines Ablaufs strukturiert werden kann. In ihnen finden sich weniger konkrete fachliche Inhalte, sondern vielmehr Zielformulierungen, Aufgabenstellungen, Methodenvorschläge, Interaktionsangebote und allgemeine Informationen zur Erarbeitung und zum Umgang mit Inhalten.

Aus dieser Idee heraus entstand ein Entwurf für die technische Landschaft an der TUHH, mit der nun einige Early-Bird-Projekte umgesetzt werden. Das Gesamtkonstrukt mit seinen einzelnen Einheiten sowie den Schnittstellen soll im folgenden genauer vorgestellt werden.

4 Technologien, Software und Schnittstellen

Die sich entwickelnde technische Umgebung der HOOU an der TUHH gliedert sich in verschiedene Komponenten, die derzeit in ersten Early-Bird-Projekten evaluiert werden. Zur Veranschaulichung der vorgestellten Strukturen und Softwarelösungen soll Abb. 1 dienen, auf die in den folgenden Abschnitten Bezug genommen wird. Dazu sei gesagt, dass die Grafik die verschiedenen Ansätze in der Umsetzung von Early-Bird-Projekten nicht umfassend abbildet. Kommende Posts in diesem Blog werden über weitere Lösungen berichten.

4.1 GitBook

GitBook ist sowohl ein Online-Service als auch eine Sammlung freier Software-Werkzeuge²¹, die für das kollaborative Erarbeiten von Textdokumenten mit linearer Struktur genutzt werden können. GitBook ist dabei nichts anderes als ein einfacher und effektiver *Static Site Generator*²². Neben dem schon erwähnten Skript *Python.Processing.Arduino* finden sich weitere Beispiele für GitBooks auf der Online-Präsenz des Services²³.

4.1.1 Markdown: Lingua franca im Netz

GitBooks werden in Markdown geschrieben. Markdown ist eine mittlerweile weit verbreitete Art, Texte auszuzeichnen, Teile also als Überschriften, Aufzählungen oder Hervorhebungen zu kennzeichnen²⁴. Einen bildlichen Eindruck des Konzepts vermittelt die Markdown-Dokumentation auf der Webseite von GitBook²⁵. Ziel des Schreibens mit Markdown ist es, die Dokumente anschließend in verschiedene Formate zu überführen, in der Regel in HTML. Dies geschieht auch im

¹⁹<http://howto.p2pu.org/>

²⁰Hinsichtlich der Planung und Umsetzung von Lernarrangements sei an dieser Stelle auf zwei einführende Texte verwiesen, die die Entwicklung von Lernarrangements als Design- oder Gestaltungstätigkeit fassen: Kerres, 2012 sowie Reinmann, 2010

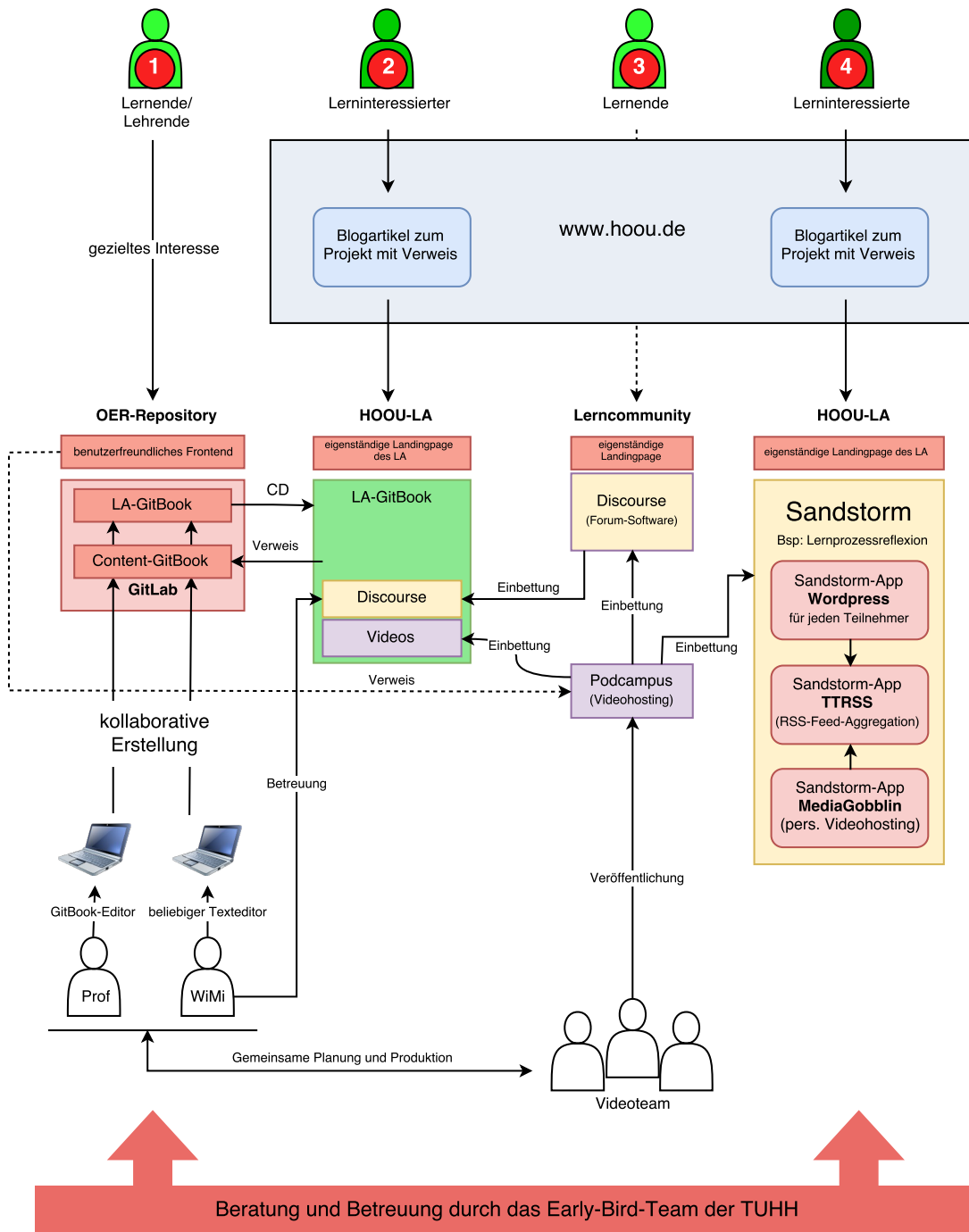
²¹<https://github.com/gitbookio>

²²Zur Aktualität von *Static Site Generators* vgl. auch <https://www.oreilly.com/ideas/static-site-generators>

²³<https://www.gitbook.com/explore>

²⁴Eine Suche im Netz (z.B. mit „lingua franca markdown“) fördert eine große Zahl von Beiträgen zutage, in denen Markdown als *lingua franca* des Internets gesehen wird.

²⁵<http://toolchain.gitbook.com/syntax/markdown.html>



LA: Lehr-Lernarrangement
 CD: Continuous Deployment
 LA-GitBook: GitBook als Strukturierungsinstrument für LA
 Content-GitBook: gegliedertes Lehr-Lernmaterial

Abbildung 1: Entwicklung von HOUU-LAs an der TUHH mit GitBook, GitLab, Discourse und Sandstorm. Quelle: Axel Dürkop

Prozess der GitBook-Produktion: Hier wird aus mehreren gegliederten Markdown-Dokumenten eine statische HTML-Seite generiert.

Für die OER-Idee ist Markdown sehr hilfreich, da Texte zusätzlich in PDF- und Office-Dokumente umgewandelt werden können. Sogar Präsentationen sind möglich²⁶. Weil beim Produzieren von Texten mit Markdown noch nicht über das Endformat entschieden wird, kann leicht der Forderung entsprochen werden, OER-Material nicht nur lizenzrechtlich offen, sondern auch technisch einfach bearbeitbar zu distribuieren. Für die Bearbeitung ist lediglich ein Texteditor (Notepad, TextEdit oder der GitBook-Editor) notwendig, der Aufwand für die Qualifikation von Autor_innen ist als gering einzuschätzen. Da Markdown sich im Netz immer mehr verbreitet, lernen Schreibende ein modernes Ausdrucksmittel, mit dem auch an vielen anderen Orten Partizipation in kollaborativen Prozessen möglich wird.

Ein weiterer Vorteil schlägt bei der Nutzung von Markdown zu Buche: Da es sich bei Markdown-Dokumenten um reinen Text handelt, können die Dateien technisch genauso behandelt werden wie solche, die Softwarecode enthalten. Das bringt, wie Abb. 4.2 zu *GitLab* zeigen soll, einen großen Vorteil hinsichtlich der Teil- und Wiederverwendbarkeit von OER-Materialien.

Markdown lässt sich problemlos mit HTML-Elementen mischen. Ein Vorteil kann darin gesehen werden, dass externe Medien und Ressourcen mithilfe des HTML-Tags `iframe` an jeder Stelle in das GitBook eingebunden werden können²⁷. Der Nutzen dieser Möglichkeit wird im späteren Abschnitt zur Verzahnung von LA-GitBooks und *Discourse* deutlich werden (s. Abschnitt 4.2.4).

4.1.2 Zusammenarbeit mit den Early-Bird-Teams

Die Produktion von LA- und Content-GitBooks erfolgt an der TUHH kollaborativ. Early-Bird-Betreuende, WiMis und Lehrende arbeiten zunächst gemeinsam an größeren Markdown-Dokumenten, die in der TUHH-Owncloud geteilt werden. Später werden diese Inhalte in einzelne Dateien nach den Konventionen von GitBook aufgeteilt und kollaborativ weiterentwickelt (s. Abb. 1). Wie wird nun aber aus diesen Dateien ein GitBook, das von anderen genutzt, kopiert, bearbeitet und weiterverwendet werden kann? Um diesen Anforderungen zu entsprechen, kommt eine Instanz der Software *GitLab* zum Einsatz. Ihre Funktionalität und Rolle soll im folgenden genauer erläutert werden.

4.2 Kollaboration mit GitHub und GitLab

Die Seite *Course in a Box: Create Your Course*²⁸ fordert mit folgendem Satz zum Handeln auf: „Fork this repository on GitHub“. *Forken* ist hierbei der Kernbegriff des Teilens in der Manier des Services GitHub²⁹ und bedeutet, die zu einem Projekt gehörenden Dateien von jemandem in den eigenen Account bei GitHub zu kopieren. Damit erlangen Nutzer_innen die Freiheit, mit den Dateien selbst schalten und walten zu können, ohne den Urheber um die Erlaubnis zum Mitmachen fragen zu müssen. Der *fork* ist hierbei in der Regel kein aggressiver Akt, der den Eintritt in einen Wettbewerb um das bessere Produkt einleitet³⁰. Vielmehr ist es Usus, dass Verbesserungen und Veränderungen aus dem *fork* an den Urheber zurückgemeldet werden - mit

²⁶Diese Umwandlung kann z.B. mit *pandoc* erfolgen. *pandoc* ist eine freie Software, mit der kreuzweise verschiedenste Textformate konvertiert werden können. Vgl. die Website des Projekts unter <http://pandoc.org/> und den Online-Konverter, mit dem die Möglichkeiten schnell ausprobiert werden können, unter <http://pandoc.org/try/>.

²⁷Ein Beispiel für die Einbettung von CC-lizenzierten Youtube-Videos in GitBook findet sich in Python.Processing.Arduino (<http://www.python-processing-arduino.de/Programmiergrundlagen/listen.html>).

²⁸<http://howto.p2pu.org/modules/start/create-your-course/>

²⁹<http://github.com/>

³⁰Die sozialen und kulturellen Implikationen des *Forkens* hat Nathaniel Tkacz (2015, S. 130 ff.) sehr aufschlussreich reflektiert.

einem *pull request*, dem Angebot, die Änderungen in den Ursprung zu integrieren. Geschieht dies, ergibt sich hinsichtlich des OER-Gedankens ein maßgeblicher Unterschied zu Wikis: In dieser Form der Zusammenarbeit sind alle Beteiligten prinzipiell immer vollständige Teilhaber der Daten, die eine Resource konstituieren. Der *fork* ist beim klassischen datenbankbasierten Wiki nicht ohne Anfrage und Herausgabe der zugehörigen Daten(bank) möglich³¹. Aber dieser Punkt ist nicht der einzige Vorteil hinsichtlich einer *maximum forkability*. Im Vergleich zum Wiki sind Rekontextualisierungen bei einem *fork* verhältnismäßig schnell gemacht, besonders wenn es sich bei den Rohdaten um Markdown-Dateien handelt, bspw. in Form eines GitBooks. Diese Änderungen können vom Ändern des Logos der Hochschule hin zur Umstrukturierung, Ergänzung oder Auslassung von Einheiten bis zur Anreicherung mit Medien reichen.

4.2.1 Maximum Forkability mit GitLab

Forken ist auch mit GitLab möglich. GitLab ist eine freie Software, die den proprietären Dienst GitHub nachbaut und den webbasierten Austausch von Dateien nach den Prinzipien des Versionskontrollsystems Git³² ermöglicht³³. Wollen Hochschulen von Services unabhängig sein, die die Regeln für Nutzung und Umgang mit Daten jederzeit ändern können und ggf. die Daten in ihrer technischen Umgebung einschließen, bieten sich freie Softwarelösungen wie GitLab an, die frei konfigurierbar und erweiterbar sind. GitLab kann auf hochschuleigenen Servern gehostet werden und kommt in der aktuellen Version für die HOOU an der TUHH zum Einsatz.

4.2.2 Teilen lernen mit Git und GitLab

Es gibt bisher wenig Forschungsergebnisse zum Einsatz von Git/GitHub/GitLab in Lehr-Lernzusammenhängen, jedoch ist die Lernkurve allgemein als hoch einzuschätzen (vgl. hierzu Zagalsky, Feliciano, Storey, Zhao & Wang, 2015). Mit meinen Studierenden erprobe ich derzeit semesterbegleitend den Einsatz von GitHub als Kollaborationsplattform in der Lehre, um die Stolpersteine zu identifizieren und mehr über die Einführung dieses vielversprechenden Tools zu lernen.

In Anbetracht der hohen Lernkurve haben wir uns an der TUHH dafür entschieden, die Kolleg_innen an den Schnittstellen zu den Early-Bird-Teams nicht mit einer Qualifikation in Sachen Git/GitLab zu belasten. Wir bitten sie lediglich, ihre Beiträge in Form von Markdown-Dateien in einem geteilten Owncloud-Ordner abzulegen. Die Aufbereitung der GitBooks übernehmen einzelne Kolleg_innen, die neugierig sind auf die Möglichkeiten dieser neuen Form der Kollaboration. Gleichzeitig versuchen wir herauszufinden, wie Git/GitLab für die Produktion und Weiternutzung von OER-Materialien an der TUHH auch für wiss. Mitarbeiter_innen, studentische Hilfskräfte und Lehrende sinnvoll und erlernbar ist.

³¹Viele Menschen verorten Git und GitHub bisher hauptsächlich in der Softwareentwicklung. Langsam erreicht das Potenzial, das die Software für kollaborative Arbeitsprozesse in sich birgt, auch andere Domänen. Robert McMillan beschrieb den Trend schon 2013 (<http://www.wired.com/2013/09/github-for-anything/>) und Services wie Gitbook.com (<https://www.gitbook.com/>) und Penflip (<http://www.madebyloren.com/github-for-writers>) setzen nun einen deutlichen Akzent auf das kollaborative Verfassen von Texten.

³²<https://de.wikipedia.org/wiki/Git>

³³Eine sehr gute Einführung in Git unter technischen Gesichtspunkten findet sich unter <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

4.2.3 Kontinuierliche Veröffentlichung von GitBooks

GitLab bringt eine Komponente mit, die automatisierte Tests in der Softwareentwicklung möglich macht (*Continuous Integration*³⁴). Dieses Werkzeug zur Qualitätssicherung wird im Kontext der GitBook-Produktion umgenutzt und sorgt für die automatische Generierung der GitBook-HTML-Seite bei jedem Hochladen (*pushen*) von Änderungen in GitLab. Das bedeutet, dass der „Quelltext“ eines GitBooks immer auf GitLab verfügbar ist, das LA- oder Content-GitBook aber automatisiert an beliebiger Stelle im Netz veröffentlicht werden kann.

Dieses *Continuous Deployment*³⁵ bietet nicht nur für die Arbeit mit und an GitBooks einen großen Mehrwert. Auch in Kombination mit anderen *Static Site Generators*³⁶, die Markdown verarbeiten und dann gut aussehende und strukturierte Webseiten bauen, macht der Workflow Sinn. GitBook ist daher nur ein austauschbarer Generator am Ende eines flexiblen technischen Ablaufs.

4.2.4 Einbettung anderer Medien und Tools

Die Einbettung anderer Medien aus dem Web ist in einem HTML-Kontext leicht möglich. Viele Dienste wie Podcampus³⁷, Youtube, Twitter u.a. bieten das Integrieren von Inhalten in eigene Zusammenhänge an. Hierbei kommt der *iframe*, ein altes HTML-Element, zum Einsatz. Der losen Kopplung von Medien und Tools wird damit ganz im Sinne der Remix-Freiheit von OER-Materialien entsprochen. Für die Early-Bird-Projekte an der TUHH werden Diskussionen, die in einem *Discourse*-Forum stattfinden, per *iframe* in ein GitBook integriert. So werden Elemente aus der *learning community* mit dem LA-Gitbook verbunden (s. Abb. 1). Auch Videos von Podcampus werden auf diese Weise eingebettet.

4.2.5 GitLab als OER-Repository

Zentral für die Idee der HOOU ist die Bereitstellung aller entstehenden OER-Materialien in einem OER-Repository. Als eine mögliche technische Implementierung wird an der TUHH auch hier GitLab gesehen: Rohdaten und fertige OER-Materialien können über GitLab der Öffentlichkeit zum Herunterladen oder *Forken* zur Verfügung gestellt werden. Dieser Ansatz soll in den nächsten Arbeitsschritten weiter ausgearbeitet und auf Tauglichkeit geprüft werden.

GitLab verwaltet Projekte, die technisch *repositories* genannt werden. In jedem *repository* können beliebig viele Dateien gespeichert werden. Diese sollten textzeilenbasiert sein, wie es bei Programmcode aber auch Markdown-Dateien der Fall ist. Dadurch wird die zuvor beschriebene Kollaboration mittels Git begünstigt. Technisch möglich ist aber auch das Speichern anderer Dateiformate wie z.B. Office-Dokumenten oder Präsentationen, denn auch diese lassen sich von Git/GitLab erfassen und können mit Hinweisen auf Änderungen annotiert werden. Nicht geeignet ist GitLab für das Speichern von Bewegtbildmaterial und nicht empfehlenswert für das Verwalten von Bildern und Grafiken.

Da die Startseite einer GitLab-Instanz sehr technisch erscheint³⁸, ist denkbar, eine ansprechendere und funktionalere Startseite mit Suchfunktion vorzuschalten. Die öffentliche Schnittstelle (API) von GitLab erlaubt eine angepasste Aufbereitung der gespeicherten Daten.

³⁴<https://about.gitlab.com/gitlab-ci/>

³⁵<https://blog.assembla.com/assemblablog/tabid/12618/bid/92411/Continuous-Delivery-vs-Continuous-Deployment-vs-Continuous-Integration-Wait-huh.aspx>

³⁶<https://www.staticgen.com/>

³⁷<http://podcampus.net/>

³⁸Die Seite <https://gitlab.com/explore> vermittelt einen Eindruck einer laufenden GitLab-Instanz.

4.3 Sandstorm als Weiterbildungsinstrument

Wie aktuelle Studien zeigen, sind längst nicht alle Studierenden so versiert im Umgang mit modernen Webapplikationen, wie ihnen immer unterstellt wird³⁹. Gleiches mag zu Teilen auch für Lehrende und andere Kolleg_innen an Hochschulen gelten, weshalb hier Informations- und Weiterbildungsbedarf angenommen werden kann. Im Rahmen der Early-Bird-Betreuung an der TUHH ist festzustellen, dass es gut ist, alle Beteiligten zunächst über die Möglichkeiten und Potenziale aktueller Werkzeuge zu informieren⁴⁰, um sie für den Prozess der gemeinsamen Entwicklung eines überzeugenden Lernarrangements zu gewinnen.

Die Software *Sandstorm*⁴¹ leistet in dieser Hinsicht einen guten Dienst, da sie über fünfzig freie Softwaretools unter einem Zugang versammelt und es Lehrenden wie Lernenden möglich macht, ihre eigene mediale Lernumgebung zu bauen. Im Zuge der Evaluation von mediengestützten Lernformen für die HOOU wird *Sandstorm* in unterschiedlichen Kontexten experimentell eingesetzt, um das Potenzial für Lehre und Weiterbildung zu ermitteln (vgl. Knutzen & Howe, 2013). Über die Funktionsweise und Möglichkeiten hatte ich an anderer Stelle⁴² schon geschrieben.

4.4 Zusammenfassung

Die Impulse, die die *Hamburg Open Online University* in der Auseinandersetzung mit den Themen *Digitalisierung*, *Offenheit* und *Hochschuldidaktik* an der TUHH gesetzt hat, sind groß. Ein Ansatz für die technische Umsetzung von HOOU-Lernarrangements an der TUHH wurde in diesem Artikel vorgestellt, wobei Bezug genommen wurde auf die Prinzipien der *Open-Education-Bewegung* und die damit einhergehenden Implikationen hinsichtlich Technik und Didaktik.

³⁹Vgl. z.B. die Studie *Digitale Lernszenarien im Hochschulbereich*, verfügbar unter https://hochschulforumdigitalisierung.de/sites/default/files/dateien/HFD_AP_Nr15_Digitale_Lernszenarien.pdf, zeigt.

⁴⁰Das Potenzial mediengestützter Lernformen lässt sich in verschiedene Kategorien einteilen, denen bestimmte Arten von Tools und Medien zugeordnet werden können. Vgl. hierzu Knutzen & Howe, 2013.

⁴¹<https://sandstorm.io/>

⁴²<https://axel-duerkop.de/blog/2016/01/25/sandsturm-in-der-digitalen-lehre/>

5 Literatur

- Kerres, M. (2012). Mediendidaktik: Konzeption und Entwicklung mediengestützter Lernangebote. München: Oldenbourg.
- Knutzen, S. & Howe, F. (2013). Digitale Medien in der gewerblich-technischen Berufsausbildung - Einsatzmöglichkeiten digitaler Medien in Lern- und Arbeitsaufgaben. foraus.de/BIBB. Zugriff am 17.12.2013. Verfügbar unter: http://datenreport.bibb.de/media2013/expertise_howe-knutzen.pdf
- Reinmann, G. (2010). Studententext Didaktisches Design. Studententext, Universität der Bundeswehr, München. Zugriff am 8.5.2015. Verfügbar unter: http://gabi-reinmann.de/wp-content/uploads/2010/04/Studententext_DD_April10.pdf
- Tkacz, N. (2015). Wikipedia and the Politics of Openness. Chicago; London: University of Chicago Press.
- Wild, E. & Möller, J. (Hrsg.). (2015). Pädagogische Psychologie (Springer-Lehrbuch). Berlin, Heidelberg: Springer Berlin Heidelberg. Zugriff am 27.04.2016. Verfügbar unter: <http://link.springer.com/10.1007/978-3-642-41291-2>
- Zagalsky, A., Feliciano, J., Storey, M.-A., Zhao, Y. & Wang, W. (2015). The Emergence of GitHub as a Collaborative Platform for Education (S. 1906–1917). ACM Press. Zugriff am 27.04.2015. Verfügbar unter: <https://doi.org/10.1145/2675133.2675284>

Dieser Beitrag ist am 28.04.2016 im Blog der HOOU unter <http://www.hoou.de/p/2016/04/28/entwicklung-einer-offenen-technischen-infrastruktur-fuer-hoou-lernarrangements-an-der-tuhh/> erschienen. Er ist lizenziert unter einer Creative Commons Namensnennung 4.0 International Lizenz⁴³.

⁴³<http://creativecommons.org/licenses/by/4.0/>