

Carl-Christian FEY¹, Christina WEKERLE, Inke BECKMANN,
Arne SCHRÖDER & Ingo KOLLAR (Augsburg/Göttingen)

Agile Methoden in Entwicklungsprojekten zur Innovation digitaler Hochschullehre

Zusammenfassung

Hochschulen stehen vor der Herausforderung, ihre Lehre im Rahmen der digitalen Transformation laufend zu innovieren. Dabei zielen sie vielfach auf eine Anreicherung der Lehre mit neuen digitalen Technologien ab. Um derartige Technologien nachhaltig und bedarfsorientiert zu entwickeln, bietet sich der Einsatz agiler Entwicklungsmethoden an, deren Umsetzung jedoch häufig mit universitären Strukturen und Prozessen kollidiert. Entsprechende Projekte sehen sich mit der Schwierigkeit konfrontiert, die Werte agiler Projektmethoden dennoch umzusetzen und zu leben. Dieser Beitrag berichtet von einem Good-Practice-Beispiel, in dem mithilfe von angepassten agilen Methoden interdisziplinär nutzbare Plugins zur digitalen Unterstützung von Feedback- und Gruppenkooperationsszenarien für das an der Universität Augsburg genutzte LMS entwickelt wurden.

Schlüsselwörter

Digitalisierung, Agile Methoden, Interdisziplinäre Hochschulentwicklung, Softwareentwicklung, Mediendidaktik

1 E-Mail: carl-christian.fey@uni-a.de



Adapting agile methods to innovate digital learning in higher education

Abstract

Universities are challenged to constantly innovate teaching and learning methods in order to adapt effectively to the transformative nature of digitalisation. One major focus is enriching teaching scenarios with new digital technologies. Agile methods can be applied to achieve a sustainable, needs-based development process. However, these methods often collide with the structures and processes prevalent in higher education organisations. As a result, innovative projects in this context struggle to uphold the inherent values and principles of agile methods. This paper provides insight into a successful good-practice implementation (based on agile methods) that develops digital plugins for feedback and group collaboration within our universities' LMS to be used in interdisciplinary learning scenarios.

Keywords

digitalisation, agile methods, interdisciplinarity, software development, media didactics

1 Problemstellung

Der Wissenschaftsrat betont in seinen Empfehlungen zur Digitalisierung in Lehre und Studium die hohe Bedeutung einer strategisch-zielgerichteten digitalen Hochschulentwicklung als Herausforderung und Zukunftsaufgabe für die einzelne Hochschule (vgl. WISSENSCHAFTSRAT, 2022). Dies stellt die Hochschulen vor eine große Herausforderung, vor allem wenn es darum geht, digitale Innovationen mit eigenem Know-how voranzubringen und entweder eigenständig oder mit externen Partnern zu entwickeln. Neben der vom Wissenschaftsrat fokussierten „strategischen“ Ebene erscheint dann vor allem die „operative“ Ebene, d.h. die Ebene der konkreten Umsetzung von hochschuldidaktischen Innovations- und Transformationsprojekten, von großer Bedeutung. Sie kann als das zentrale Stellrad begriffen

werden, an dem über Misslingen oder Gelingen entsprechender Projekte entschieden wird (z. B. über Akzeptanz, Nachhaltigkeit, Reichweite).

Um diese Ziele zu erreichen, eignen sich insbesondere *agile Projektmanagementmethoden*, da sie der besonderen Komplexität gerecht werden, die Digitalisierungsprojekten im Kontext von organisationalen Strukturen auszeichnen (vgl. PETRY & KONZ, 2021): informationstechnische, organisationale und rechtliche Anforderungen, Fragen der Integration und Interoperabilität mit bestehenden Lösungen der Hochschulinfrastruktur, der Passung von (software-)technischen Lösungen und medien-/hochschuldidaktischen Anforderungen sowie die Beteiligung der vielfältigen hochschulinternen Stakeholder (vgl. KERRES, 2018, S. 492ff.).

Dieser Beitrag beschreibt den Einsatz agiler Projektmanagementmethoden im von der Stiftung Innovation in der Hochschullehre geförderten Projekt „Kompetenzentwicklung durch digitale, authentische und feedbackbasierte Lehr-Lernszenarien stärken“ (KodiLL) zur evidenzbasierten Weiterentwicklung des Learning Management Systems der Universität Augsburg in Zusammenarbeit mit einer externen Softwareentwicklungsfirma (data-quest). Zunächst wird erläutert, was unter einer evidenzbasierten Gestaltung digitaler Hochschullehre zu verstehen ist. Danach folgt die Darstellung der Kennzeichen agiler Softwareentwicklung am Beispiel des Scrum-Frameworks sowie einer kurzen Problematisierung im Kontext von Hochschulstrukturen. Anschließend wird modellhaft beschrieben, wie das an der Universität Augsburg genutzte LMS Stud.IP mithilfe agiler Projektmanagementmethoden weiterentwickelt wird. Im Vordergrund steht dabei die Darstellung der umgesetzten Adaption der agilen Projektmanagementmethode Scrum, die in unserem Setting vor allem Erfolge im Kontext der interdisziplinären Nutzbarkeit, einer hohen Akzeptanz durch Stakeholder sowie eines dynamischen und transparenten Umgangs mit Anforderungen erzielt. Der Beitrag schließt mit einer Darstellung von Learnings und Transferempfehlungen.

2 Zur Notwendigkeit einer evidenzbasierten Gestaltung digital angereicherter Hochschullehre

Um die Qualität digitaler Lehr-Lern-Angebote für die Hochschullehre zu sichern, bietet sich bei ihrer Gestaltung eine enge Orientierung an Evidenzen der empirischen Lehr-Lern-Forschung an (WEKERLE et al., 2022). Diesen Evidenzen zufolge wird digitalen Medien ein erhebliches Potenzial zur Optimierung von Lehr-Lern-Prozessen zugeschrieben (NIEGEMANN & WEINBERGER, 2020). Dieses Potenzial wird bislang allerdings oft nur unzureichend ausgeschöpft, und stärker aktivierende Lehr-Lern-Formate wie etwa Flipped-Classroom-Szenarien oder Möglichkeiten zur digitalen Kollaboration bleiben häufig ungenutzt (LOHR et al., 2022). Derartige Befunde sind als sehr ernüchternd einzuschätzen. So wird in der empirischen Lehr-Lern-Forschung argumentiert, dass digitale Medien so gewählt, gestaltet und eingesetzt werden sollten, dass sie (a) bei den Studierenden eher produktive (z. B. Online-diskussion in Kleingruppen) als rezeptive Lernprozesse (z. B. einen Artikel lesen) auslösen (vgl. CHI & WYLIE, 2014), (b) Studierende im Sinne problembasierten Lernens (HMELO-SILVER, 2004) zur Auseinandersetzung mit authentischen Problemen anregen (z. B. mithilfe interaktiver Videos, in die Prompts eingebettet sind) und (c) sicherstellen, dass Studierende möglichst adaptives, prozessbezogenes Feedback auf ihren Lernprozess erhalten (HATTIE & TIMPERLEY, 2007).

3 Prinzipien agiler Softwareentwicklung

3.1 Was bedeutet agiles Projektmanagement?

Das Konzept des agilen Projektmanagements stammt aus der Softwareentwicklung und damit im weitesten Sinne aus der IT-Industrie. Im Gegensatz zu klassischem Projektmanagement nach dem sog. „Wasserfallmodell“, das Anforderungen zu Projektbeginn definiert und sequentiell in einer Abfolge von Planung, Analyse, Design/Entwicklung, Implementation, Testing, Release abarbeitet, setzt agiles Projektmanagement darauf, die Prozesse der Produktentwicklung mit alternativen Methoden und Techniken als Regelungsprozess zu gestalten, der der Dynamik und Komplexi-

tät des Produktumfelds besser gerecht wird (vgl. PREUSSIG, 2020, S. 11ff.). Inzwischen ist agiles Projektmanagement in großer Breitenwirkung im Kontext der freien Wirtschaft angekommen und etabliert – nicht nur im Bereich der Softwareentwicklung, sondern bei allen Arten von Produkten und Projekten, deren kontextuelle Bedingungen eine solche Vorgehensweise nahelegen (vgl. HASEBROOK et al., 2019)².

Das Prinzip der dynamischen Anpassung führt zu einem Hauptmerkmal agiler Methoden, nämlich ihrer Zyklizität, d. h. dass das entstehende Produkt „inkrementell“ weiterentwickelt wird, neue Anforderungen und Bedarfe kontinuierlich erkannt und integriert werden, aber auch Fehleinschätzungen in Bezug auf bereits implementierte Funktionen und Features laufend korrigiert werden.

3.2 Das Scrum-Framework

Eine vor allem in der Softwareentwicklung etablierte Vorgehensweise agilen Projektmanagements ist die Anwendung des sog. Scrum-Prozesses. Scrum definiert sich als ein aus dem sog. „Agilen Manifest“ (BECK et al., 2001) abgeleitetes reaktives Framework auf der Basis empirischer Prozesskontrolle (SCHWABER & SUTHERLAND, 2020) und beschreibt die Etablierung und Aufrechterhaltung eines Zyklus von „Transparency“, „Inspection“ und „Adaptation“ als notwendig, um diesem Anspruch gerecht zu werden (MCGREAL & JOCHAM, 2018, S. 139ff.). Auffällig und für den wissenschaftlichen Kontext hochanschlussfähig ist die prinzipielle Ähnlichkeit zum Design (Based)-Research-Ansatz, der sich ebenfalls durch ein zyklisches Vorgehen mit realen Erprobungsphasen auszeichnet (vgl. REINMANN, 2020).

In Scrum wird angestrebt, von Beginn des Entwicklungsprozesses an in kurzen Zyklen (sog. Sprints) lauffähige Software Releases zu produzieren (sog. Inkremente), die dann mit allen Stakeholdern und im „Feld“ (d. h. mit realen Nutzer:innen) eingesetzt und in ihrer Zielerreichung bewertet werden können. Diese Releases bauen aufeinander auf (Iterativität) und unterliegen einem kontinuierlichen Verbesserungsprozess hinsichtlich ihrer Funktionen (Features). Dazu definiert das Framework die Rolle des „Product Owners“, die/der als zentrale:r Entscheider:in mit unternehmerischer Funktion über die Funktionen und Features des zu entwickelnden Produkts

² Vgl. auch die im Zusammenhang von Hochschullehre eingesetzte Adaption eduScrum (STURM & RUNDNAGEL, 2021).

fungiert, welche vom sog. „Development Team“ (den Softwareentwickler:innen) umgesetzt werden. Unterstützt werden beide vom sog. „Scrum-Master“, der dafür Sorge trägt, dass die Elemente des Frameworks methodisch sauber und vollständig umgesetzt werden.

Die umzusetzenden Funktionen der Software werden vom Product Owner in einem sog. „Product Backlog“ gesammelt und in ihrer Wichtigkeit für das Produkt (Value) sowie vom Development Team in ihrem zeitlichen Aufwand bewertet (Cost). In der Regel wird dafür mit sog. „User Stories“ gearbeitet, die die Funktion in narrativer Konnotation aus der Sicht der Nutzer:innen allgemeinverständlich beschreiben und in Zusammenarbeit mit den Entwickler:innen mit Akzeptanzkriterien anreichern, die die Umsetzung der Funktion überprüfbar machen. Das Product Backlog ist eine emergente und geordnete Liste von Features und Funktionen, die das Produkt ausmachen und verbessern sollen. Es ist das Dokument, aus dem alle Entwicklungsarbeit abgeleitet und definiert wird (vgl. SCHWABER & SUTHERLAND, 2020). Für die Sprints werden aus dem Product Backlog nach einer Aufwandsabschätzung für die Entwicklung jeden Inkrements User Stories zur Umsetzung ausgewählt und in ein sog. „Sprint Backlog“ integriert, das die Arbeitsleistung für das nächste Inkrement für die Entwickler:innen verbindlich definiert. Dies geschieht in einem sog. „Sprint Planning“-Event, in dem alle vorgesehenen Rollen des Scrum-Prozesses beteiligt sind. Das Rahmenmodell sieht zudem das sog. „Sprint Review“ vor, in dem die Ergebnisse des letzten Inkrements unter Einbezug weiterer Stakeholder der Organisation gemeinsam begutachtet und getestet werden. Hinzu kommt die sog. „Sprint Retrospective“, in der das Team auf einer Meta-Ebene den letzten Sprint reflektiert, um gemeinsam Verbesserungen mit Blick auf Qualität und Effizienz zu eruieren.

SCRUM FRAMEWORK

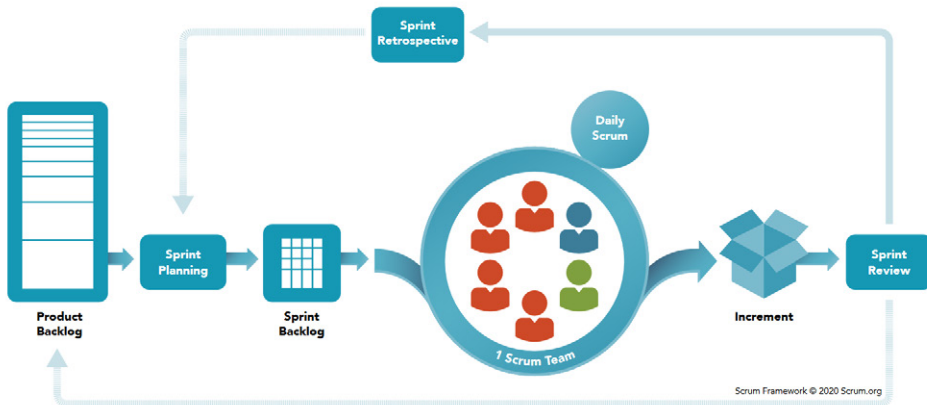


Abb. 1: Visualisierung des Scrum-Prozesses – <https://www.scrum.org>

3.3 Herausforderungen im Kontext Hochschule

Mit den Herausforderungen der Umsetzung agiler Projekte im Kontext etablierter Strukturen, Arbeits-/Prozess-, „kulturen“ und komplexen Verantwortlichkeiten im Bereich der Hochschule werden Implementierungsprojekte immer wieder konfrontiert. So beschreibt PREUSSIG (2020), wie klassische Projekte unter universitären Rahmenbedingungen „agiler“ gemacht, und SÄLZLE et al. (2021), wie experimentell/inkrementelle Vorgehensweisen angewandt werden können, die zu den in unserem Projekt gewählten vergleichbar sind. EULER (2016) zufolge sind die Strukturen an Hochschulen geprägt von einer eher losen Koppelung der Organisationseinheiten mit jeweils hohen Freiheitsgraden und einer nur gering ausgeprägten Kontroll- und Koordinationsstruktur auf übergeordneter Ebene – dies gilt vor allem für das professorale Personal, das zudem auch unterschiedlichen Loyalitäten unterliegt (der eigenen Disziplin, der wissenschaftlichen Forschung). Auch gibt es eine ganze Reihe an Stakeholdern, deren systematische Integration bei Entwick-

lungsprozessen eine große Herausforderung darstellt. Diese Herausforderung lässt sich nicht über in Unternehmen verbreitete Top-Down-Strukturen bearbeiten (vgl. GRAF-SCHLATTMANN et al., 2020). Diesen besonderen „Akteurskonstellationen“ gilt es Rechnung zu tragen (vgl. MÜLLER, 2016). Agile Projektmanagementmethoden und insbesondere die individualisierte und spezialisierte Rollenverteilung des Scrum-Frameworks, die z. B. Entscheidungen in der Regel bei einer einzelnen Person (dem Product Owner) verortet, können daher nicht ohne Weiteres im Hochschulkontext angewandt werden.

Um trotzdem wesentliche Vorteile dieser Methoden erhalten zu können, ist es nötig, Anpassungen vorzunehmen, ohne jedoch die Werte und Prinzipien agiler Entwicklung aufzugeben, indem die methodische Umsetzung an die Gegebenheiten universitärer Organisationsstrukturen angepasst werden.

4 Umsetzung zur Weiterentwicklung des LMS Stud.IP an der Universität Augsburg

4.1 Das LMS Stud.IP

Stud.IP ist ein open-source Lern-Management-System, das im deutschsprachigen Raum primär von Universitäten und Hochschulen, aber auch von weiteren Bildungseinrichtungen und Verbänden eingesetzt wird (BOUZO, 2017; SCHNEKENBURGER, 2009)³. Stud.IP unterstützt das Lehren, Lernen und dessen Verwaltung innerhalb des zugangsbeschränkten Raums der jeweiligen Instanz. Die Universität Augsburg setzt Stud.IP seit 2004 unter dem Namen „Digicampus“ für die aktuell ca. 20.000 Studierenden in 86 Studiengängen auf Bachelor- und Masterniveau (Stand November 2022) als zentrales Lehr- und Lernmanagementsystem mit individuellen Anpassungen ein.

In erster Linie wird durch das System die Kommunikation zwischen Lehrenden und Teilnehmenden (lehr-)veranstaltungsbezogen und webbasiert unterstützt. Darüber hinaus bietet Stud.IP zahlreiche Funktionen zur Abbildung von Strukturen und Prozessen an Hochschulen (z. B. Ablaufpläne für Veranstaltungen, Evaluationen,

³ Vgl. <https://www.studip.de/>

Verwaltung von Prüfungen und Modulstrukturen sowie Schnittstellen für externe Systeme wie z. B. HISinOne).

Über den frei zugänglichen Plugin-Marktplatz (<http://plugins.studip.de>) stehen Erweiterungen der Basisfunktionalität von Stud.IP zur Verfügung. Über 160 Erweiterungen für das Kernsystem stehen dort lizenzkostenfrei und quelloffen bereit. Jede Bildungseinrichtung kann sich hier nach eigenen Bedarfen an den Open-Source-Komponenten bedienen oder eigene Lösungen einstellen. In der Lehre sind – nicht erst seit der Corona-Krise – neue Trends und Vermittlungsweisen von Lehrstoff hinzugekommen. Beispiele sind die Integration von Videolehre über Livestreams oder Aufzeichnungen, ePrüfungen oder die Möglichkeit, unterschiedliche Medien, Tests und Aufgaben für die Aufbereitung von Lehrinhalten miteinander zu kombinieren (BECKMANN, 2021). Die im Rahmen dieses Beitrags beschriebenen Entwicklungen des KodiLL-Projekts an der Universität Augsburg werden ebenfalls über die Plugin-Schnittstelle in das System integriert und stehen zukünftig auch anderen Bildungseinrichtungen zur Verfügung.

4.2 Das universitätsweite Projekt „KodiLL“

Seit August 2021 läuft an der Universität Augsburg das interdisziplinäre, von der Stiftung Innovation in der Hochschullehre geförderte Projekt KodiLL. In vier Teilprojekten arbeiten insgesamt 46 Mitarbeiter:innen an der Entwicklung von digital angereicherten Lehr-Lern-Szenarien, in denen den beschriebenen Erkenntnissen der empirischen Lehr-Lern-Forschung entsprechend Studierenden (a) eine produktive statt einer rein rezeptiven Mediennutzung ermöglicht wird, (b) authentische, für zukünftige berufliche Kontexte relevante Probleme präsentiert und bearbeitet werden können, und (c) adaptives, elaboriertes Feedback realisiert wird. Jedes Teilprojekt ist interdisziplinär zusammengesetzt (<https://www.uni-augsburg.de/de/forschung/projekte/kodill/>).

Zwei der vier Teilprojekte arbeiten zusammen mit der Firma data-quest an der Entwicklung zweier Plugins für die universitätsweit genutzte Stud.IP-Instanz. Im ersten Projekt wird auf Basis der Forschung zum computerunterstützten kollaborativen Lernen (z. B. VOGEL et al., 2017) ein Plugin namens „coLearn!“ entwickelt, das Lehrende bei der Planung, Strukturierung und Umsetzung von Kleingruppenkooperationsphasen unterstützen soll. „coLearn!“ bietet hierfür u. a. die Möglichkeit,

Studierende zu Gruppen zusammenzufassen, Aufgaben zu definieren, unter den Lernenden einer Kleingruppe unterschiedliche Rollen zu verteilen und den Kooperationsprozess rollenabhängig in distinkte Phasen zu unterteilen. Auch ermöglicht es „coLearn!“, Studierenden mit jeder beliebigen Rolle und für jede beliebige Phase weitere Aufgaben, Interaktionsprompts und Ressourcen zur Verfügung zu stellen, um eine hohe Qualität des Lernprozesses sicherzustellen (siehe Abb. 2).

Plugin coLearn!

Anweisungen für das Kooperationskript "Phishing Mails"

	Vorbereitung in Einzelarbeit (15 Minuten)	Gemeinsam eine Phishing-E-Mail gestalten (20 Minuten)	Phishing-E-Mails anderer Gruppen kommentieren (20 Minuten)	Reflexion (10 Minuten)
01 Teamleiterin	Teamleiterin - Phase 1	Teamleiterin - Phase 2	Teamleiterin - Phase 3	Teamleiterin - Phase 4
02 Technikerin	Technikerin - Phase 1	Technikerin - Phase 2	Technikerin - Phase 3	Technikerin - Phase 4
03 Social-Engineerin	Social-Engineerin - Phase 1	Social-Engineerin - Phase 2	Social-Engineerin - Phase 3	Social-Engineerin - Phase 4
04 Designerin	Designerin - Phase 1	Designerin - Phase 2	Designerin - Phase 3	Designerin - Phase 4

Aktuelles Szenario
 Phishing Mails (nicht aktiv)

Ansichten
 Szenarien
 Rollen
 Phasen
Anweisungen
 Kleingruppen

Abb. 2: Screenshot des coLearn!-Elements zur Ausgestaltung von Phasen-Rollen-Kombinationen in coLearn!. Die Lehrperson kann hier unterschiedliche Rollen und Kooperationsphasen definieren und strukturieren.

Das zweite Teilprojekt beschäftigt sich mit der Entwicklung eines Plugins, mit dem Lehrende tutorielle sowie peer-basierte Feedbackszenarien umsetzen können. Das Plugin „getFeedback!“ ermöglicht u. a., dass Tutor:innen editierbare Vorlagen für die Formulierung ihres Feedbacks erstellen (Abb. 3). Im peer-basierten Modus bietet „getFeedback!“ zudem die Möglichkeit, zu bestimmen, welche Studierende einander Feedback auf ihre Ausarbeitungen geben sollen. Auch können Lehrende den Prozess des Feedbackgebens und der Feedbackrezeption durch die Definition entsprechender Anweisungen (Prompts) feiner strukturieren.

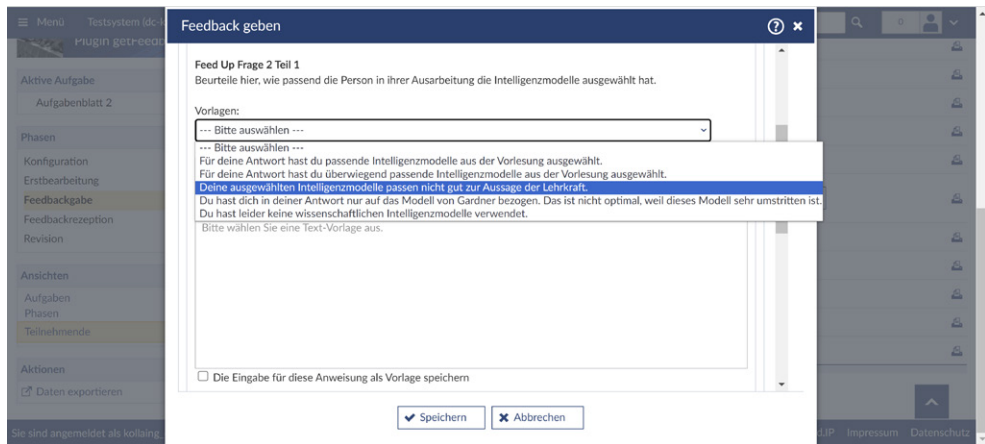


Abb. 3: Screenshot des Eingabeformulars für Tutor:innen zur Feedbackgabe in „getFeedback!“. Die/der Tutor:in kann Textbausteine vorformulieren und an die jeweilige studentische Ausarbeitung anpassen. Ebenfalls möglich ist das Geben von Feedback in Form von Freitext oder Punkten.

4.3 Adaption und Umsetzung agiler Methoden im Projekt

Die Entwicklung der beiden Plugins folgt im Kern den oben dargestellten Prinzipien agiler Softwareentwicklung:

1. Es wurden für jedes zu entwickelnde Plugin Teams gebildet, die aus den Projektverantwortlichen der unterschiedlichen Fakultäten sowie Mitarbeiter:innen der zentralen Unterstützungseinheit (mit mediendidaktischer und medientechnischer Expertise) bestehen. Diese Teams übernehmen gemeinschaftlich die Rolle des Product Owners und sind gleichzeitig auch aktive Nutzer:innen des Plugins in konkreten Einsatzszenarien. In den Teams sind außerdem weitere Stakeholder vertreten (z. B. Rechenzentrumsmitarbeiter:innen).
2. Zu Projektbeginn erfolgten Kurzschulungen für alle Projektbeteiligten zu agilem Projektmanagement und der intendierten methodischen Vorgehensweise.

3. Im Rahmen von regelmäßig stattfindenden Workshops mit den Entwickler:innen der Firma data-quest wird ein Product Backlog bestehend aus User Stories, Aufwandsabschätzungen und Akzeptanzkriterien für die softwaretechnische Umsetzung erstellt.
4. Während dieser Workshops kommt es zur Diskussion und Schärfung der User Stories und Akzeptanzkriterien mit dem Development Team, durch die eine gemeinsame Sprache und ein gemeinsames Verständnis über die Produktentwicklung und die benötigten Funktionen und Features entwickelt wird. Auch der Kontext der Nutzung (Einsatzszenarios) wird dabei thematisiert und unter interdisziplinärer Perspektive transparent gemacht.
5. Dabei erfolgt vor dem Start einer neuen Entwicklungsphase eine gemeinschaftliche, diskussionsorientierte, aber zuletzt numerische Einschätzung des „Wertes“ der einzelnen User Stories für das Plugin durch die Product Owner (Vergabe von „Value Points“ mittels der Methode „Scrum-Poker“).
6. Parallel dazu geben die Entwickler:innen (data-quest) eine grobe Aufwandsabschätzung der zeitlichen Umsetzungsbedarfe für jede User-Story ab (im Product Backlog für alle sichtbar dokumentiert).
7. In einer spezifisch dafür vorgesehenen Sitzung findet die Selektion der für jede Iteration des Plugins umzusetzenden User Stories auf Basis einer Abwägung von „Wert“ (Value) und „Aufwand“ (Cost) durch die Product Owner statt (Sprint Planning und Sprint Backlog-Erstellung). Nur eine begrenzte Anzahl an User Stories kann pro Iteration umgesetzt werden. Daher wird auf die Auswahl von Funktionen, die für die nächste Entwicklungsphase (Iteration) am relevantesten sind, größter Wert gelegt.
8. Daher erfolgt auch eine nochmalige Prüfung der selektierten User Stories anhand von exemplarisch beschriebenen Einsatzszenarien für konkrete Lehrveranstaltungen aller an der Entwicklung beteiligten Fakultäten/Professuren und Lehrstühle, um Gefährdungen der Umsetzbarkeit der Szenarien durch fehlende oder unzureichend implementierte Funktionen auszuschließen (Vermeidung sog. „Show-Stopper“).

9. Danach erfolgt die Programmierung einer lauffähigen Version der Plugins durch die Entwickler:innen auf Basis der für das Backlog selektierten User Stories (Sprint Backlog).
10. Nach Abschluss der Entwicklungsarbeiten für die jeweilige Iteration erfolgt ein gemeinsames Review und eine Demo-Session, in denen die nun entwickelten Funktionen der Plugins vorgestellt, ausprobiert und diskutiert werden.
11. Die Taktung der Iterationen/Releases geschieht in Anlehnung an den Semesterwechsel. Zu diesem Zeitpunkt wird auch die jeweils aktuelle Version der Plugins auf ein Testsystem aufgespielt, welches dann in regulären Lehrveranstaltungen zum Einsatz kommt. Dabei werden auch mehrere disziplinspezifische Lehr-Lern-Szenarien unter Verwendung der aktuellen Plugin-Version umgesetzt und getestet.
12. Nach dem Einsatz in konkreten Lehrveranstaltungen wird in Zusammenarbeit mit Mitarbeiter:innen des zentralen Unterstützungsteams eine Evaluation der Plugins mittels quantitativer und qualitativer Erhebungen aufseiten der Nutzer:innen (Dozierende und Studierende) durchgeführt. Die Evaluationsergebnisse fließen in die nächste Phase der Überarbeitung des Product Backlogs ein.
13. Begleitend zum Einsatz der Plugins in Lehrveranstaltungen besteht für die Nutzer:innen die Möglichkeit, auftretende Fehler (Bugs) in eine Versionsmanagement-Applikation einzutragen, die dann von den Entwickler:innen behoben werden.
14. Der Prozess beginnt von Neuem (ab Punkt 3).

Die dieser Adaption zugrunde liegende Rollen- und Prozessstruktur wird in den Abbildungen 4 und 5 verdeutlicht.

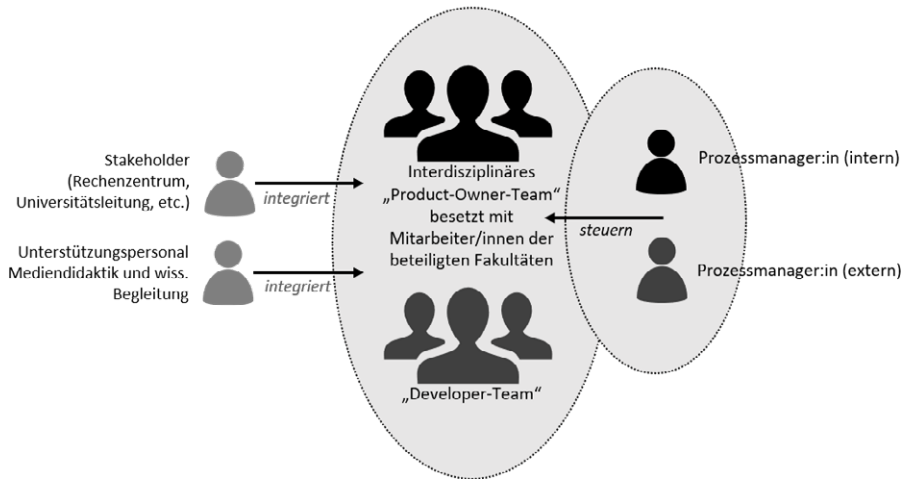


Abb. 4: Rollenstruktur

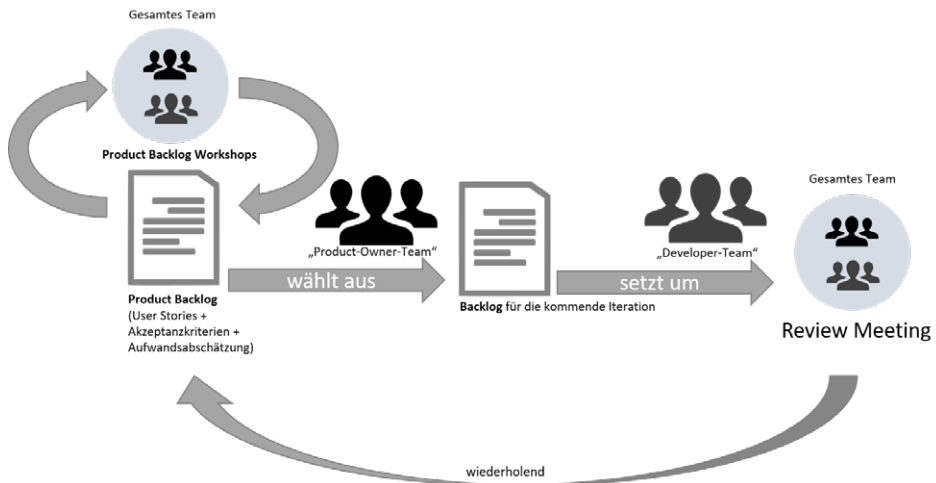


Abb. 5: Prozessstruktur

In der Adaption der unter 3 aufgeführten Prinzipien agiler Softwareentwicklung gab es auch bewusste Abweichungen vom originären Scrum-Framework. Beispielsweise wurde die Rolle der/des Prozessverantwortlichen sowohl aufseiten der Hochschule als auch aufseiten der Firma data-quest eingeführt, die die organisatorische und methodische Umsetzung gemeinsam verantworten (vergleichbar zur Funktion des Scrum-Masters im Scrum-Framework). Bedeutsam erscheint die Abweichung vom Scrum-Framework auf der Ebene der Entscheidungsprozesse (sozusagen als „demokratisierter“ Scrum Owner), wobei die Beibehaltung der prinzipiellen „Ownership“ aufseiten der Gruppe der Product Owner (Hochschule) mit Nachdruck verfolgt wurde und keine Delegation von diesbezüglicher Verantwortung in Richtung extern an das Development Team (data-quest) oder intern (übergeordnete Steuerungsgremien) erfolgte. Mit diesen Abweichungen wird auf die besonderen strukturellen Bedingungen der Hochschule reagiert (vgl. Kap. 3.3).

5 Learnings und Transferempfehlungen

Die hier skizzierte Vorgehensweise erwies sich in der Umsetzung als praktikabel und erfolgreich. Der Projektkontext ist durch eine enorme Breite und Heterogenität bzgl. der beteiligten Fakultäten, Lehrstühle und Professuren gekennzeichnet und zielt auf einen breiten gesamtuniversitären Transfer der Entwicklung ab. Die Reflexion der Vorgehensweise im Rahmen der Review- und Retrospektiven-Meetings zeigte eine hohe Akzeptanz aufseiten aller Beteiligten aufgrund eines ausgeprägten Erlebens von Partizipation, Verantwortung und Entscheidungsmöglichkeiten und erhöht damit die Chancen auf gelingenden Transfer. Auch wird deutlich, dass durch die mehrfachen Iterationen, die an den Semesterverlauf gekoppelt waren, relativ kurzfristige Erfolgserlebnisse erzielt wurden, indem jeweils neue lauffähige Versionen der Plugins zur Verfügung standen. Als besonders wertvoll wurde im Rahmen der durchgeführten Retrospektiven von den Mitarbeiter:innen bewertet, dass Erkenntnisse aus der Nutzung in realen Einsatzszenarien direkt in den nächsten Entwicklungsschritt einfließen konnten. Durch diese Vorgehensweise konnten die zu Beginn der Projektphase formulierten Produktziele kontinuierlich geschärft und andere als obsolet verworfen werden. Bedeutsam war auch der durch die Methode sichergestellte laufende kommunikative Austausch mit den Entwickler:innen, durch den unterschiedliche Akteur:innen aus der Wirtschaft und dem universitä-

ren Umfeld ein gemeinsames Verständnis bzgl. der Arbeitsprozesse und der umzusetzenden Funktionen (Features) entwickelten. Die interdisziplinäre Nutzbarkeit der entwickelten Plugins konnte abgesichert werden, indem die Entscheidung über den Wert einzelner Funktionen und die notwendigen Akzeptanzkriterien für eine erfolgreiche Umsetzung „demokratisiert“ wurde und die Mitarbeiter:innen der unterschiedlichen Fakultäten und Arbeitseinheiten als „Product Owner“ ein gleichberechtigtes Gewicht im Prozess hatten und Anforderungen für „ihre“ konkreten Einsatzszenarien formulieren konnten. Schließlich zeigte die erste Evaluation dieser Einsatzszenarien, dass aufseiten der Studierenden eine überdurchschnittliche Akzeptanz des Plugins vorlag und auch der (subjektive) fachliche Kompetenzerwerb gefördert werden konnte bzw. von der Vor- zur Nacherhebung signifikant zunahm.

Neben den positiven Ergebnissen der Umsetzung müssen jedoch auch kritische Anmerkungen gemacht werden. So zeigte sich das Vorgehen durch die teilweise hochfrequenten Workshops und Meetings als relativ zeit- und abstimmungsaufwändig. Es stellte sich außerdem bald heraus, dass Nachschärfungen im Bereich Review und Retrospektive nötig wurden, die erst nach Ablauf des ersten Drittels der Projektphase umgesetzt werden konnten. Was die Integration der Stakeholder:innen anbelangt, wurde von den Projektbeteiligten insbesondere der eher indirekte Einbezug der Gruppe der Studierenden als Nutzer:innen der Plugins bemängelt, die zwar über Maßnahmenevaluationen und teilweise auch über Beteiligung in der Definition von Features und Funktionen einbezogen, jedoch nicht systematisch mit Entscheidungskompetenz in die entsprechenden Teams integriert waren. Eine direkte Integration von Studierenden in die Teams (als Product Owner) wird daher inzwischen im Projekt auch umgesetzt.

Als Desiderata lassen sich die Bereiche der Forschung in den Handlungsfeldern „operative/methodische Projektumsetzung“ sowie „Prozessimplementierung und -optimierung“ identifizieren. Neben der in universitären Projekten generell bisher weitgreifend berücksichtigten effektorientierten Evaluation von Maßnahmen tritt damit auch die Evaluation von Prozessen und Methoden als notwendiges Forschungsfeld in den Fokus. Die umgesetzte Methode ist u. a. im Rahmen der Retrospektiven auf der operativen Ebene für wissenschaftliche Evaluationen grundsätzlich sehr gut anschlussfähig und bietet damit Raum für entsprechende Aktivitäten. Methodologisch wäre zukünftig von Interesse, inwieweit eine konkrete Verzahnung

von agilen Methoden mit evidenzbasierten Forschungsansätzen wie Design Based Research (DBR) gelingen und gestaltet werden kann.

Aus der hier vorgestellten Umsetzung ergibt sich die grundsätzliche Empfehlung, in Digitalisierungsprojekten auch im Hochschulkontext nicht mehr mit klassischem Projektmanagement zu arbeiten, sondern mit großer Offenheit agile Methoden in Anpassung an die lokalen Gegebenheiten zu adaptieren und dafür sukzessive Überzeugungsarbeit zu leisten. Dies stellt eine kommunikative Herausforderung auf unterschiedlichen Ebenen dar, bei der es hilfreich erscheint, sowohl intern als auch extern entsprechend geschultes Personal in die Projekte (zumindest auf Leitungsebene) zu integrieren. Hierbei ist auch der Aspekt der Herstellung von Transparenz über Methoden und Entscheidungsabläufe bedeutsam – und die Bereitschaft (echte) Entscheidungsverantwortung an eine partizipative Teamstruktur zu delegieren. Gelingt dies, können agile Methoden äußerst effektiv zur nachhaltigen und bedarfsorientierten Innovation digitaler Hochschullehre eingesetzt werden.

6 Literaturverzeichnis

Beck, K. et al. (2001). *The Agile Manifesto*. Agile Alliance. <http://agilemanifesto.org/>

Beckmann, I. (2021) Offenes Ökosystem Stud.IP – Nach innen und außen offen. *eLearning Journal, Themenheft: Bildungstechnologien für Corporate Learning*, 2021/2022, 46–47.

Bouzo, A. (2017). Was ist eigentlich ein Learning Management System? *eLearning Journal, Themenheft Praxisratgeber. Strategie, Konzeption, Didaktik: eLearning in Unternehmen richtig einsetzen*, 2017/2018, 32–41.

Chi, M. T. H. & Wylie, R. (2014). The ICAP Framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, 49(4), 219–243. <https://doi.org/10.1080/00461520.2014.965823>

Euler, D. (2016). Gestaltung von Veränderungsprozessen im Rahmen der pädagogischen Hochschulentwicklung. In T. Brahm, T. Jenert & D. Euler (Hrsg.), *Pädagogische Hochschulentwicklung: Von der Programmatik zur Implementierung* (1. Aufl., S. 261–280). Springer VS.

- Graf-Schlattmann, M., Meister, D. M., Oevel, G. & Wilde, M.** (2020). Kollektive Veränderungsbereitschaft als zentraler Erfolgsfaktor von Digitalisierungsprozessen an Hochschulen. *Zeitschrift für Hochschulentwicklung*, 15(1), 19–39. <https://doi.org/10.3217/zfhe-15-01/02>
- Hasebrook, J., Kirmße, S. & Fürst, M.** (2019). *Wie Organisationen erfolgreich agil werden: Hinweise zur erfolgreichen Umsetzung in Zusammenarbeit und Strategie* (1. Auflage 2019). essentials. Wiesbaden: Springer Fachmedien.
- Hattie, J. & Timperley, H.** (2007). The power of feedback. *Educational Research Review*, 77(1), 81–112. <https://doi.org/10.3102/00346543029848>
- Hmelo-Silver, C. E.** (2004). Problem-based learning: what and how do students learn? *Educational Psychology Review*, 16, 235–266. <https://doi:10.1023/B:ED-PR.0000034022.16470.f3>
- Kerres, M.** (2018). *Mediendidaktik: Konzeption und Entwicklung digitaler Lernangebote*. De Gruyter. <https://doi.org/10.1515/9783110456837>
- Lohr, A., Vejvoda, J., Schultz-Pernice, F., Maier, R., Jiang, S., Fischer, F. & Sailer, M.** (2022). *Digitale Bildung an bayerischen Hochschulen während der Corona-Pandemie*. München: Vereinigung der bayerischen Wirtschaft.
- McGreal, D. & Jocham, R.** (2018). *The professional product owner: Leveraging Scrum as a competitive advantage. The professional scrum series*. Addison-Wesley.
- Müller, P. D. W.** (2016). Vom „Durchwurschteln“ zur kontinuierlichen Verbesserung? – Akteurskonstellationen deutscher Universitäten bei Innovationsprozessen von Lehre und Studium. In T. Brahm, T. Jenert & D. Euler (Hrsg.), *Pädagogische Hochschulentwicklung* (S. 189–202). Springer Fachmedien Wiesbaden.
- Niegemann, H. & Weinberger, A.** (Hrsg.) (2020). *Handbuch Bildungstechnologie. Konzeption und Einsatz digitaler Lernumgebungen*. Berlin: Springer.
- Petry, T. & Konz, C.** (2021). *Agile Organisation – Methoden, Prozesse und Strukturen im digitalen VUCA-Zeitalter*. Dr. Götz Schmidt.
- Preußig, J.** (2020). *Agiles Projektmanagement: Agilität und Scrum im klassischen Projektumfeld* (2. Aufl.) (Haufe Fachbuch: Bd. 10248). Haufe.
- Reinmann, G.** (2020). Ein holistischer Design-Based Research-Modellentwurf für die Hochschuldidaktik. *Educational Design Research*, 4(2). <https://doi.org/10.15460/eder.4.2.1554>

Sälzle, S., Blank, J., Vogt, L. & Bleicher, A. (2021). Möglichkeitsräume an Hochschulen post Corona experimentell gestalten. *Zeitschrift für Hochschulentwicklung*, 16(3), 149–160. <https://doi.org/10.3217/zfhe-16-03/09>

Schnekenburger, C. C. (2009). *E-Learning an der Universität Rostock: Eine explorative, quantitative Online-Trenderhebung zum tatsächlichen Einsatz von Stud.IP*. Unveröffentlichte Dissertation. Universität Rostock. https://rosdok.uni-rostock.de/resolve/id/rosdok_disshab_0000000447

Schwaber, K. & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. <https://scrumguides.org/scrum-guide.html>

Sturm, N. & Rundnagel, H. (2021). Agiles Lernen digital gestützt: Die Methode eduScrum in der Hochschullehre. In *Digitalisierung in Studium und Lehre gemeinsam gestalten* (S. 577–598). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-32849-8_32

Vogel, F., Wecker, C., Kollar, I. & Fischer, F. (2017). Socio-cognitive scaffolding with computer-supported collaboration scripts: A meta-analysis. *Educational Psychology Review*, 29, 477–511. <https://doi.org/10.1007/s10648-016-9361-7>

Wekerle, C., Daumiller, M. & Kollar, I. (2022). Using digital technology to promote higher education learning: The importance of different learning activities and their relations to learning outcomes. *Journal of Research on Technology in Education*, 54(1), 1–17. <https://doi.org/10.1080/15391523.2020.1799455>

Wissenschaftsrat. (2022). *Empfehlungen zur Digitalisierung in Lehre und Studium*. <https://doi.org/10.57674/SG3E-WM53>

Autor:innen



Dr. Carl-Christian FEY || Universität Augsburg, Zentrum für digitales Lehren und Lernen (DigiLLab) || Universitätsstraße 10, D-86159 Augsburg

<https://digillab.uni-augsburg.de/>

carl-christian.fey@uni-a.de



Dr. Christina WEKERLE || Universität Augsburg, Lehrstuhl für Psychologie m.B.B.d. Pädagogischen Psychologie || Universitätsstraße 10, D-86159 Augsburg

<https://www.uni-augsburg.de/de/fakultaet/philsoz/fakultat/padagogische-psychologie/team/>

christina.wekerle@uni-a.de



Dr. Inke BECKMANN || data-quest Suchi & Berg GmbH || Friedländer Weg 20a, D-37085 Göttingen

<https://www.data-quest.de/>

Beckmann@data-quest.de



Arne SCHRÖDER || data-quest Suchi & Berg GmbH || Friedländer Weg 20a, D-37085 Göttingen

<https://www.data-quest.de/>

schroeder@data-quest.de



Prof. Dr. Ingo KOLLAR || Universität Augsburg, Lehrstuhl für Psychologie m.B.B.d. Pädagogischen Psychologie || Universitätsstraße 10, D-86159 Augsburg

<https://www.uni-augsburg.de/de/fakultaet/philsoz/fakultat/padagogische-psychologie/team/prof-dr-ingo-kollar/>

ingo.kollar@uni-a.de