

DiNa-Sonderausgabe

TAGUNGSBAND MINT SYMPOSIUM 2023

MINT-Lehre
gemeinsam gestalten –
Lehre erforschen,
Wissen teilen



TAGUNGSBAND ZUM 5. SYMPOSIUM
zur Hochschullehre in den MINT-Fächern

Informatik-Grundlagenmodul mit den Methoden Inverted Classroom und Scrum

Karsten Morisse, Christian Heidemann, Sebastian Möller
Hochschule Osnabrück

Zusammenfassung

Im Modul Algorithmen und Datenstrukturen ist das Inverted Classroom Modell mit der Scrum-Methodik kombiniert. Die Studierenden erarbeiten die Inhalte des Moduls im Lernmanagementsystem mithilfe von Videoaufzeichnungen, digitalem Skript und interaktiven Übungseinheiten. Der Wegfall der klassischen Vorlesung ermöglicht mehr Zeit zur Beantwortung von Fragen, Diskussionen sowie der Reflexion des Erlernten durch Hörsaal-Quizze. Die Themen der Veranstaltung werden vorgegeben, aber die Bearbeitung erfolgt individuell und die Studierenden gestalten ihre eigenen Lernprozesse.

Theorie und Praxis der Veranstaltung werden analog zur Scrum-Methodik in mehrwöchigen Sprints im Team bearbeitet. Die Aufgaben sind in den Kontext einer virtuellen Betriebssystemumgebung eingebettet und bauen aufeinander auf. Das Softwareprojekt wird hierzu als GitLab-Repository zur Verfügung gestellt. Die Verwendung von Git und integrierten Test-Routinen entsprechen einer realitätsnahen Vorgehensweise, wie sie in der Softwareentwicklung allgemein gängige Praxis ist.

1. Einleitung und Veranstaltungskontext

Das Inverted Classroom Modell (ICM) (Lage, Platt, und Treglia 2000), (Bergmann und Sams 2012) erfreut sich seit einiger Zeit großer Beliebtheit in der Hochschullehre. Hierbei werden die Phasen der Wissensvermittlung und der individuellen Vertiefung umgekehrt: Was bisher während der gemeinsamen Veranstaltungszeit präsentiert wurde, wird nun in eine vorgelagerte Selbstlernphase ausgelagert. Die Präsenzzeit zwischen Studierenden und Lehrenden wird für vertiefende Gruppenarbeitsphasen, Diskussion oder andere aktivierende Formate genutzt. Das ICM-Konzept wird disziplin- und veranstaltungsübergreifend in der Lehre eingesetzt. Zahlreiche Einsatzbeispiele sind in den Tagungsbänden der jährlich stattfindenden Konferenz ICM Beyond (<https://www.icmbeyond.net>) beschrieben.

Aus Studierendensicht ist die Teilnahme an einer ICM-Veranstaltung eine Herausforderung. Für ein erfolgreiches Lernen sind insbesondere personale Kompetenzen wie Selbstmotivation, Durchhaltevermögen und Selbstregulationskompetenz erforderlich (Pöpel & Morisse, 2019). Die Vorbereitung der Präsenztermine, also die individuelle Wissensaneignung vor dem gemeinsamen Termin mit der Lehrperson, wird oftmals nicht mit der Konsequenz durchgeführt, wie es für den idealtypischen Einsatz des ICM wünschenswert wäre. Um diesem Umstand entgegenzuwirken, wurde für das Modul Algorithmen & Datenstrukturen das ICM mit Ideen aus Scrum kombiniert (siehe auch Morisse & Heidemann, 2021).

Das Modul Algorithmen & Datenstrukturen ist ein Pflichtmodul im zweiten Fachsemester zweier Informatik-Studiengänge an der Hochschule Osnabrück und besteht aus einer Plenumsveranstaltung (3 SWS) und einem Praktikumstermin (1 SWS) in Kleingruppen. Die Modulinhalte sind Effiziente Algorithmen, Algorithmisches Denken, Elementare Datenstrukturen sowie das Zusammenspiel von Algorithmen und Datenstrukturen und haben in einer zunehmend von Algorithmen durchsetzten Welt hohe berufspraktische Relevanz. Aus Sicht der Informatik sind dies klassische Inhalte, die auf Bachelorniveau nur geringen Änderungen unterworfen sind. Der theoretische, wie auch der praktische Teil stellt für die Studierenden mit noch wenig Programmierkenntnissen regelmäßig eine Schwierigkeit dar. Das Modul schließt gemäß Prüfungsordnung mit einer Klausur und einem unbenoteten Leistungsnachweis für das Praktikum ab.

2. Scrum

Die u.a. von Sutherland und Schwaber (2021) entwickelte Scrum-Methodik ist ein etabliertes Vorgehensmodell in der Software-Entwicklung, bei dem eine iterative Produktentwicklung durch ein kleines Entwicklungsteam in Arbeitszyklen (Sprints) betrieben wird. Scrum basiert auf den Haltungen Commitment, respektvoller Umgang, Mut, Offenheit sowie Fokus und es gibt klar definierte Rollen: Das Scrum-Team ist das Entwicklungsteam, welches im Kontext der Lehre von einer Studierendengruppe gebildet wird. Die Leitung des Teams erfolgt durch den Scrum-Master. Diese Rolle sorgt dafür, dass die Teammitglieder die Prinzipien und Praktiken von Scrum einhalten. Aufgabe des Product Owners wiederum ist die Festlegung der Produktziele und -anforderungen mit dem Ziel, das bestmögliche Produkt zu erstellen. Die spezifischen

Produktfunktionen werden im sogenannten Produkt-Backlog festgehalten. Hierin ist der Gesamumfang des zu erstellenden Produktes beschrieben und es wird eine Priorisierung der zu erstellenden Funktionen festgelegt.

Die Arbeit des Scrum-Teams an dem Produkt erfolgt in kurzen Arbeitszyklen, den sogenannten Sprints. Ein Sprint startet mit einer Planung (Sprint Planning), bei der festgelegt wird, welche Elemente aus dem Produkt Backlog in diesem Sprint bearbeitet werden sollen. Das Ergebnis der Planung wird im Sprint-Backlog dokumentiert und am Ende eines Sprints sollte als Ergebnis ein Zwischenprodukt (Product Increment) erstellt sein. Für die enge Abstimmung des Scrum-Teams dienen kurze tägliche Arbeitstreffen (Daily Standup). Am Ende eines Sprints, der typischerweise ein bis vier Wochen dauert, steht das Review. Hierin wird das erstellte Ergebnis des Sprints begutachtet. Was sollte erreicht werden und was wurde erreicht? Aus dem Sprint-Backlog offen gebliebene Elemente werden im Rahmen der Planung des nächsten Sprints neu priorisiert. Den Abschluss des Sprints bildet die Retrospektive. Hierin wird die Zusammenarbeit im Team in den Blick genommen: Was hat gut funktioniert und was sollte im nächsten Sprint besser gemacht werden? Dieser reflektierende Blick auf das Team stellt die Verbesserung der Zusammenarbeit im Team in den Fokus.

Scrum definiert damit den Rahmen für eine inkrementelle Arbeitsweise, bei der die permanente Verbesserung sowohl des Produktes als insbesondere auch der Zusammenarbeit im Team im Fokus stehen. Die in einem Sprint gesammelten Erfahrungen werden in der Retrospektive kritisch beleuchtet und es werden Verbesserungen für den nächsten Arbeitszyklus entwickelt.

Mit eduScrum (Wijnands & Stolze, 2019) oder Scrum4Schools (Gloger, o. J.) werden die Ideen von Scrum in die Schule übertragen. Bei beiden werden die zentralen Scrum-Elemente beibehalten, jedoch mit anderen Namen und mit kindgerechten Werkzeugen für den Einsatz an Schulen versehen. In den Arbeiten von Lutter (2019) und Sturm und Rundnagel (2021) sind Einsätze von eduScrum in der Hochschullehre beschrieben. Zwei Aspekte erschweren diesen Einsatz jedoch: Einerseits die Durchführung der Daily StandUp-Meetings und andererseits die fehlende Anwesenheitspflicht. Das Daily StandUp-Meeting ist ein wichtiges Element der Methodik, es erfolgt hier in sehr enger Taktung eine Abstimmung der Arbeit innerhalb des Teams. Da es in der Hochschule in der Regel wöchentliche Veranstaltungstermine gibt, muss die Abstimmung im Kontext der Veranstaltung als Weekly Standup durchgeführt werden und die tägliche Abstimmung führt das Team eigenverantwortlich durch. Die fehlende Anwesenheitspflicht in Modulen kann dazu führen, dass die studentischen Teams im Laufe eines Semesters auseinanderfallen. Wenn sich Studierende im Laufe eines Semesters dazu entschließen, ein Modul doch nicht zu absolvieren und ihr Team verlassen, müssen die Teams möglicherweise reorganisiert werden.

In der Regel sind die Abläufe und Elemente von Scrum den Studierenden nicht bekannt, so dass mit der Nutzung im Lehrkontext die Gefahr der Überforderung droht, da neben dem Erlernen der Fachinhalte auch der Scrum-Prozess organisiert werden muss. Obwohl die eigenverantwortliche und selbstständige Arbeit des Teams eines der Paradigmen von Scrum ist, ist bei der Einführung von Scrum im Lehrkontext eine anfängliche Begleitung der Scrum-Teams durch die Lehrperson ratsam. Weniger im Sinne einer Kontrolle und Überwachung, sondern in beratender, ermutigender und unterstützender Funktion.

3. ICM + Scrum = ICMScrum

Beim Inverted Classroom entfällt die Phase der Wissensvermittlung durch eine klassische Vorlesung und wird durch eine selbstgesteuerte Lernphase ersetzt. Ein bekanntes Phänomen im ICM ist die fehlende oder gering ausgeprägte Selbstregulationskompetenz, die eine defizitäre Vorbereitung der Präsenztermine zur vertieften Zusammenarbeit zur Folge hat. Aus diesem Grund wurde ein an Scrum angelehnter stützender organisatorischer Rahmen eingeführt, der als eine Möglichkeit der in Fallmann und Reinthaler (2016) sowie Pöpel und Morisse (2019) beschriebenen Scaffolding-Maßnahmen zu verstehen ist. Hier fließen ICM und Scrum also konkret zusammen. Zu Beginn des Semesters teilen sich die Studierenden in Lernteams ein, um die Lernsprints gemeinsam zu bearbeiten. Bei der Bildung der Teams kann es hilfreich sein, Studierenden mit unterschiedlichen Lern-Charakteristika oder Stärken in einem Team zu vereinen, so dass sich die Studierenden im Sinne eines Peer Learning gegenseitig unterstützen können (Fallmann & Reinthaler, 2016). Im ersten Termin wird der Scrum-Zyklus bestehend aus Planung, Produktion, Review und Retrospektive in Form eines agilen Spiels erlebt. Hierfür bietet sich beispielsweise das Ball Point Game oder das Taschenrechner-Spiel an (Böhmer, 2022). Dabei werden alle Elemente von Scrum in einem engen Time-Boxing durchlaufen, um die Wirkung der empirischen Prozesskontrolle erlebbar zu machen.

Die Lehrperson gestaltet als Product Owner den inhaltlichen Rahmen der einzelnen Sprints und gibt die zu erlernenden Inhalte vor. Im Sinne von Scrum bleibt die interne Organisation der Arbeit vollständig dem Team überlassen. Eine Person aus dem Scrum-Team übernimmt zusätzlich die Rolle des Scrum-Masters, um den Scrum-Prozess im Blick zu haben. Als Scaffolding-Maßnahme kann die Lehrperson hier anfänglich

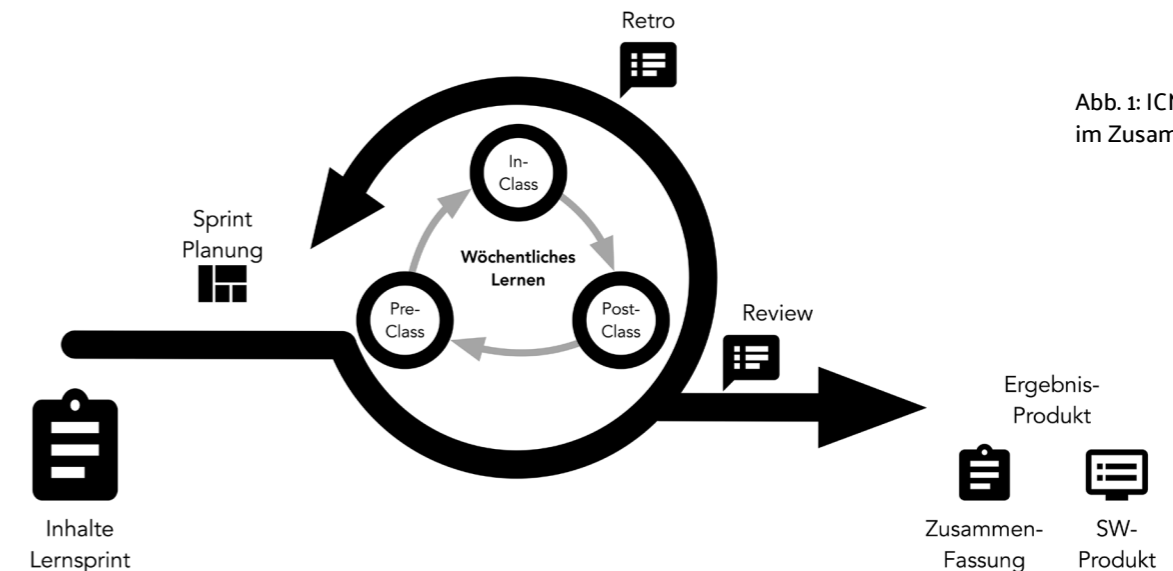


Abb. 1: ICM und Scrum im Zusammenhang

unterstützend mitwirken, diese Unterstützung aber im Laufe der Bearbeitung zurücknehmen. Die Verantwortung für das erfolgreiche eigenständige Lernen geht also schrittweise auf die Studierenden über. Die Erarbeitung der Theorie wird im Sinne des Inverted Classroom anhand der bereitgestellten Materialien von den Studierenden selbstständig vollzogen. Als Material werden im zentralen LMS Ilias (<https://lms.hs-osnabrueck.de>) Texte, Videos und Übungen bereitgestellt. Die laut Stundenplanung regelmäßigen (wöchentlichen) Lehrveranstaltungstermine werden wie beim regulären ICM genutzt, um Fragen zu klären und die Möglichkeit zur Diskussion zu geben. Die Termine werden aber auch für das wöchentliche StandUp-Meeting und für ein regelmäßiges Review des Lernstands genutzt, welches als kurzes Hörsaal-Quiz angeboten wird. Die ICM-Methodik ist also in die Scrum-Zyklen eingebettet. In Abb. 1 ist das Zusammenspiel der einzelnen Elemente im Zusammenhang dargestellt.

Die Wirkung von Scrum kann sich nur entfalten, wenn die Arbeit in einzelne Sprints aufgeteilt wird und bei der Bearbeitung der Sprints auch die Team-Zusammenarbeit im Rahmen einer Retrospektive kritisch reflektiert wird. Zu diesem Zweck wird der gesamte Inhalt der Lehrveranstaltung in drei thematische Blöcke aufgeteilt und für jeden Themenblock wird ein zu entwickelndes Produkt definiert. Im Kontext einer Informatik-Veranstaltung bietet sich hierbei die Entwicklung eines Software-Produktes an. Dies birgt jedoch die Gefahr der Vernachlässigung der Theorie durch die Studierenden. Daher werden je Sprint zwei Produkte festgelegt: eine kurze Zusammenfassung der Theorie, die als Hilfsmittel für die Klausur verwendet werden darf, sowie ein Programmierprojekt. Die Durchführung des Reviews und der Retrospektive schließen bei Scrum einen Sprint ab. Genau dies lässt sich auch im Hochschulkontext sehr gut als Feedback-Instrument nutzen. Das Review der theoretischen Inhalte wird neben

wöchentlicher Quizze im Hörsaal auf der elektronischen Lernplattform als freiwilliges Assessment angeboten. Im Review des praktischen Anteils stellt das Team die entwickelte Software vor und diskutiert diese sowie aufgetretene Schwierigkeiten bei der Erstellung mit der Lehrperson. Mit der Retrospektive wird die Zusammenarbeit im Team kritisch reflektiert. Für die konkrete Durchführung gibt es vielfältige Methoden (Derby, Larsen & Köster, 2018). In der einfachsten Form führen die Studierenden ein Brainwriting an einem Whiteboard durch anhand der Fragen: Was ist gut gelaufen? Was ist nicht gut gelaufen? Was wollen wir im nächsten Sprint besser machen?

4. Praxisnahe Werkzeuge im Praktikum

Für den praktischen Teil der Veranstaltung wird wie oben beschrieben ein Software-Produkt definiert, welches von den Studierenden in den Lernsprints begleitend zur Erlernung der Theorie entwickelt wird. Die üblicherweise in einem Grundlagenmodul behandelten Algorithmen lassen sich schnell in gängigen Foren finden oder sind durch generative KI erzeugbar. Daher sind die Aufgaben in den komplexeren, aber praxisnahen Projektkontext hsOS eingebettet, eine virtuelle Betriebssystemumgebung mit einer Konsole als Bedien-schnittstelle. Die Umsetzung erfolgt mittels professioneller Werkzeuge, die im ersten Fachsemester noch nicht behandelt wurden, für die spätere Berufspraxis aber hohe Relevanz besitzen. Als Entwicklungsumgebung wird IntelliJ (<https://www.jetbrains.com/de-de/idea/>) im Zusammenspiel mit dem Build-Tool Maven (<https://maven.apache.org>) verwendet. Für die Unterstützung der kollaborativen Software-Entwicklung im Team kommt das Versionierungssystem GitLab (<https://about.gitlab.com>) zum Einsatz. Für alle Lernsprints werden über GitLab die zu bearbeitenden Entwicklungsprojekte bereitgestellt und

müssen von den Studierenden nach Fertigstellung wieder in GitLab zurückgespielt werden. Die Aufgaben sind erfolgreich bearbeitet, wenn die in GitLab hinterlegten Tests erfolgreich durchlaufen wurden. Die Aufgabenplanung der Studierenden und Hilfestellungen werden über das Issue-Tracking-System von GitLab abgebildet. Neben den klassischen Informatik-Inhalten des Moduls werden somit auch relevante praktische Kompetenzen im Tool-Umgang vermittelt, die sowohl im weiteren Verlauf des Studiums als auch für die spätere Berufspraxis sehr hilfreich sind.

5. Fazit

Die selbstständige Ausgestaltung des eigenen Lernprozesses ist für die Studierenden im zweiten Fachsemester der Informatik-Studiengänge eine neue Erfahrung und zugleich auch eine Herausforderung. Die Veranstaltungen im ersten Semester sind in der Regel sehr stark instruktionsorientiert, so dass die eingesetzte und für die Studierenden neuartige Methodik intensiv kommunikativ begleitet werden muss. Insbesondere der Einsatz von GitLab zur Unterstützung der kollaborativen Software-Entwicklung ist eine zusätzliche Herausforderung und der Sinn und Zweck der im Zusammenspiel eingesetzten Methoden und Werkzeuge muss wiederholt deutlich gemacht werden. Wenn dieser Transfer aber gelingt, erkennen die Studierenden einen deutlichen Mehrwert, insbesondere auch für die Arbeit in späteren Semestern. Dies ist ein Ergebnis regelmäßiger informeller Feedback-Gespräche. Da die Phase der Wissensvermittlung in Form der klassischen Vorlesung entfällt, gewinnt man in der Präsenzphase Zeit für den Dialog mit den Studierenden. Diese bessere Ansprechbarkeit der Lehrenden, sowohl für die theoretischen Inhalte wie auch für den Praxisteil der Veranstaltung wurde positiv hervorgehoben. Gelobt wurden die hervorragende Organisation und

die Abstimmung zwischen dem Theorie- und dem Praxisteil der Veranstaltung. Da der Lehrende die freigewordene Zeit während des regulären Vorlesungstermins in den individuellen Lernsupport investieren konnte, haben die Studierenden eine sehr gute Betreuung wahrgenommen und dies auch entsprechend zurückgespiegelt. Ein besonderes Augenmerk verdient auch der kommunikative Stil zwischen Lehrenden und Studierenden. Bei einem Veranstaltungskonzept, welches getragen wird von der Interaktion und der Diskussion, kommt einer empathischen und wertschätzenden Kommunikation eine besondere Bedeutung zu. Wenn der Lehrende den Rollenwandel vom Wissensvermittler zum Lernbegleiter vollzieht, ist eine offene und wertschätzende Kommunikationskultur unabdingbare Voraussetzung.

Eine weitere Erkenntnis aus den Feedback-Gesprächen ist die Länge der Sprints. Hier erscheinen vier Wochen sehr lang und die Studierenden wünschen sich eine kürzere Taktung. Dies wurde mit Blick auf die Theorie dadurch abgefangen, dass eine wöchentliche Themenplanung eingeführt wurde. In den wöchentlichen Präsenzterminen wurde dann jeweils ein Hörsaal-Quiz mit einem Audience Response System zum entsprechenden Thema durchgeführt.

Die zuvor erwähnten auseinanderbröckelnden Teams lassen sich systembedingt nicht verhindern. Die fehlende Anwesenheitspflicht und die individuelle Studiensituation führen dazu, dass Studierende sich anfänglich zu einem Modul anmelden, dieses aber nicht vollständig absolvieren und dann im Laufe des Semesters nicht weiter teilnehmen. Hier muss mit dem verbliebenem Team überlegt werden, ob es das Modul mit reduzierter Teamstärke durchführen möchte oder ob man Teams zusammenlegt.

Die intensive Zusammenarbeit der Studierenden im Rahmen der Lernsprints mit der regelmäßigen Durchführung von Retrospektiven zur Reflexion adressiert personale Kompetenzen, so dass der Methodenmix in besondere Weise fachliche wie auch personale Kompetenzen fördern kann. Studierende mit anfänglichen Schwierigkeiten können durch Scaffolding-Maßnahmen auf den richtigen Weg gebracht werden bzw. können innerhalb des Teams durch Peer-Learning mitgenommen werden. Eine im Sommersemester 2023 begleitende Befragung der Studierenden zur Entwicklung personaler Kompetenzen auf Basis einer Selbsteinschätzung konnte diese Wirkung des eingesetzten Methodenmixes noch nicht bestätigen. Hier ist noch genauer zu untersuchen, ob die zahlreichen Angebote der Reflexion über das eigene Arbeitsverhalten eher hinderlich wirken, da sich die Studierenden häufiger über ihre eigene Unkenntnis Gedanken machen. Ein zweiter hinderlicher Punkt kann die Koexistenz des neuen Lernformates mit den anderen klassischen Lernformaten sein.

Beide Seiten – Studierende wie auch Lehrende – sollten sich also immer wieder die erforderlichen Grundhaltungen Offenheit, Mut und respektvoller Umgang für den erfolgreichen Einsatz von Scrum in Erinnerung rufen. Das dies in der vorgestellten Veranstaltung scheinbar gelungen ist, zeigt das Zitat eines der teilnehmenden Studierenden „Ich freue mich immer auf die Review-Termine“. Zur Erinnerung: Das Review ist der Termin zur Vorstellung des erreichten Ergebnisses, also eher ein Prüfungs-Setting, bei dem Studierende ihre Arbeit präsentieren. Weitere positive Resonanzmerkmale durch die Studierenden waren die hoffnungsvollen Fragen, ob denn die Veranstaltungen im kommenden Semester auch nach ICM-Scrum organisiert sind.

Literatur

Bergmann, J. & Sams, A. (2012). Flip your Classroom. International Society for Technology in Education.

Böhmer, C. (2022). Agile Games. Göttingen: BusinessVillage GmbH.

Derby, E., Larsen, D. & Köster, L. (2018). Agile Retrospektiven. München: Verlag Franz Vahlen GmbH.

Fallmann, I., & Reinthaler, P. (2016). „Bedeutung und Förderung von selbstreguliertem Lernen im Inverted Classroom“. In J. Haag & C. Freisleben-Teutscher (Hrsg.), Das Inverted Classroom Modell. Begleitband zur 5. Konferenz Inverted Classroom and Beyond 2016 (S. 45-54). FH St. Pölten: ikon Verlagsgesellschaft.

Gloger, B. (o. J.). „Scrum4Schools I borisgloger consulting“. Abgerufen am 24. November 2021 von: <https://www.borisgloger.com/ueber-uns/csr/scrum-4-schools>

Lage, M., Platt, G., & Treglia, M. (2000). „Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment“. Journal of Economic Education 31(1), 30-43. DOI: <https://doi.org/10.2307/1183338>.

Lutter, A. (2019). „Eduscrum - Eduscrum“. EduScrum. Abgerufen am 25. November 2021 von: <https://www.eduscrum.hs-mannheim.de/>

Morisse, K. & Heidemann, C. (2021). „Inverted Classroom kombiniert mit Scrum für die Informatik-Lehre“. 9. Fachtagung Hochschuldidaktik Informatik (HDI) 2021, 133-138.

Pöpel, N. & Morisse, K. (2019). „Inverted Classroom: Wer profitiert – wer verliert? Die Rolle der Selbstregulationskompetenzen beim Lernen im umgedrehten MINT-Klassenraum“. die hochschullehre 5, 55-74.

Schwaber, K. & Sutherland, J. (2021). „Scrum Guides“. Scrum Guides. Abgerufen am 24. November 2021 von: <https://scrumguides.org/>.

Sturm, N., und Rundnagel, H. (2021). „Agiles Lernen digital gestützt: Die Methode eduScrum in der Hochschullehre“. In: Hochschulforum Digitalisierung (Hrsg.). Digitalisierung in Studium und Lehre gemeinsam gestalten. Innovative Formate, Strategien und Netzwerke (S. 577-598). Wiesbaden: Springer VS.

Wijnands, W. & Stolze, A. (2019). „Transforming Education with EduScrum“. In D. Parsons & K. MacCallum (Hrsg.). Agile and Lean Concepts for Teaching and Learning: Bringing Methodologies from Industry to the Classroom (S. 95-114). Singapore: Springer.

Angaben zu den AutorInnen**Karsten Morisse**

Studium der Informatik und Promotion in Mathematik an der Universität Paderborn, industrielle Tätigkeiten als Entwicklungsleiter im Bereich von Streaming Media Technologie. Nach einer Professur für Multimedia-Technologie an der Fachhochschule Trier seit 2000 Professor für Medieninformatik an der Hochschule Osnabrück.

Christian Heidemann

Bachelorstudium der Medieninformatik an der Hochschule Osnabrück mit anschließender Tätigkeit als wissenschaftlicher Mitarbeiter in einem Forschungsprojekt zur Eutergesundheit bei Milchvieh.

Sebastian Möller

Ausbildung zum Fachinformatiker und Studium Medieninformatik an der Hochschule Osnabrück. Danach verschiedene Tätigkeiten an der Hochschule Osnabrück und Fachhochschule Münster. Derzeitig Projektmitarbeiter und eingeschrieben im Masterstudiengang.