D. Demmler, D. Krupka, H. Federrath. (Hrsg.): INFORMATIK 2022,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2022 1067

Workshop zum Einsatz agiler Methoden im Informatikunterricht

Ein Minigolf-Spiel als erstes Softwareprojekt

Tobias Bahr[1] und Bernd Zinn[2]

Abstract: In diesem Bericht zum Workshop für Informatiklehrpersonen wird eine Unterrichtseinheit zum Themenbereich Algorithmen der Klassenstufe 8 für ein Programmierprojekt mit Scratch vorgestellt. Die teilnehmenden Informatiklehrpersonen sollen in dem Workshop die Grundlagen des agilen Prozesses lernen und reflektieren unterrichtsrelevante Aspekte, indem diese die Schülerperspektive einnehmen sowie strukturgebend die Vor- und Nachteile zum Einsatz von agilen Methoden im Informatikunterricht diskutieren. Abschließend werden Erkenntnisse aus der Unterrichtspraxis vorgestellt sowie methodisch-didaktische Aspekte zur Umsetzung von Agilen Methoden diskutiert.

Keywords: Informatikunterricht; Agile Methoden; Projekt; Softwareentwicklung; kollaboratives Arbeiten

1 Motivation

Agile Methoden werden bereits seit mehreren Dekaden im wirtschaftlichen Sektor und insbesondere auch in der Softwareentwicklung eingesetzt [Co05, Wi03]. Über die Jahre haben sie auch ihren Weg in den Informatikunterricht der Schulen gefunden [Br15, Br19, Di15 Ka15, Ri09]. Anknüpfend an die pädagogische und fachdidaktische Forschung zu der Projektmethode [Sc13] wurden praxisnahe Handlungsanweisungen für die unterrichtliche Umsetzung formuliert [Wa09, Sc13, Tr11]. Wenngleich sich die vorliegenden Veröffentlichungen zu agilem Informatikunterricht bislang nur mit der unterrichtlichen Umsetzung und kontextuellen Einkleidung beschäftigen, nach den uns vorliegenden Daten keine Evidenz zu den Wirkungseffekten der Umsetzung der Methode vorliegt, wird übergreifend unterstellt, dass die Nutzung agiler Methoden im Unterricht das Potenzial hat, um die personalen und sozialen Kompetenzen (u. a. Kommunikationsfähigkeit, Selbstorganisationfähigkeit, Teamfähigkeit) von Schüler*innen zu fördern [Br19, Ri09]. Im hier thematisierten Workshop wird eine Unterrichtseinheit vorgestellt, mit der im Anfangsunterricht Informatik anhand eines Softwareprojekts für die 8. Klassenstufe mit der visuellen Programmierumgebung Scratch agile Methoden eingeführt werden. Dem Vorgehen wird

[1] Universität Stuttgart, Abteilung Berufspädagogik mit Schwerpunkt Technikdidaktik (BPT), Azenbergstraße 12, Stuttgart 70174, tobias.bahr@ife.uni-stuttgart.de
[2] Universität Stuttgart, Abteilung Berufspädagogik mit Schwerpunkt Technikdidaktik (BPT), Azenbergstraße 12, Stuttgart 70174, bernd.zinn@ife.uni-stuttgart.de

doi:10.18420/inf2022_89

unterstellt, dass das niedrigschwellige Einstiegsniveau von Scratch [Ma10] den Vorteil für die Schüler*innen und der Lehrperson bietet, den Fokus auf den methodischen Unterrichtsablauf im Kontext des agilen Projektmanagements zu richten.

Auf agile Methoden kann in späteren Klassenstufen und bei komplexeren Projektarbeiten immer wieder zurückgegriffen werden, wenn die grundlegenden Vorgehensweisen der Methode den Lehrenden und Lernenden bekannt sind. Auch die Nutzung von einzelnen agilen Methoden im Unterricht kann von den Beteiligten strukturgeben und positiv wahrgenommen werden [Br19]. Gerade bei Informatikprojekten in der Oberstufe bzw. Kursstufe oder im fächerverbindenden Profilfach Informatik, Mathematik, Physik (IMP), welches an allgemeinbildenden Gymnasien in Baden-Württemberg unterrichtet wird und bei denen die Lerninhalte spiralcurricular zunehmend komplexer werden und sich Projektarbeiten über mehrere Wochen erstrecken, bietet ein strukturiertes Vorgehen offensichtliche Vorteile. Positive Aspekte werden sowohl für den Lernprozess als auch für die Projektleistung der Schüler*innen gesehen [Br15]. Lehrpersonen können davon profitieren, dass sich durch das Project Board transparent darstellen lässt, welche Schüler*innen an welcher Aufgabe gearbeitet haben und wie der Projektfortschritt in den einzelnen Teams ist. Zudem kann es die Lehrpersonen im Hinblick auf eine transparente Leistungsbewertung der Schüler*innen unterstützen.

2 Der agile Prozess

Da an anderen Stellen der agile Prozess bereits ausführlich beschrieben und dokumentiert ist [Br19], wird der grundlegende Prozess hier nur kurz erklärt. Für weitergehende Beschreibungen und Erläuterungen wird auf das Lehrbuch *Agile Schule* verwiesen, auf das sich die hier dargestellten Prozessabläufe ebenfalls beziehen [Br19]. Schwerpunkt des vorliegenden Beitrags soll daher vor allem das umgesetzte Softwareprojekt in Scratch und die Umsetzung der Unterrichtseinheit sein. Das Pflichtenheft für die Software sieht u.a. vor, dass das Minigolf-Spiel von einem User (Spieler*in) auf drei Bahnen gespielt werden kann. Im Hauptmenü hat der User die Möglichkeit sich einen Spielernamen zu geben. Am Ende des Spiels nach der letzten Bahn, oder im Hauptmenü wählbar, soll der Highscore angezeigt werden. Mehrere User sollen das Spiel nacheinander nutzen können.

Nachfolgend wird auf die Unterrichtseinheit und dessen Ablauf eingegangen. Zu Beginn des Minigolf-Projektes wird mit einem fragend-entwickelnden Unterricht eine Anwendungsfalldiagramm erstellt (vgl. Abbildung 1), bei dem möglichst alle Anwendungsfälle des Users für den Spielverlauf visualisiert werden. Sollten die Schüler*innen bereits in Teams gut zusammengearbeitet haben, so bietet es sich an, dass die Erstellung des Anwendungsfalldiagrammes in den einzelnen Teams arbeitsgleich vorgenommen wird.

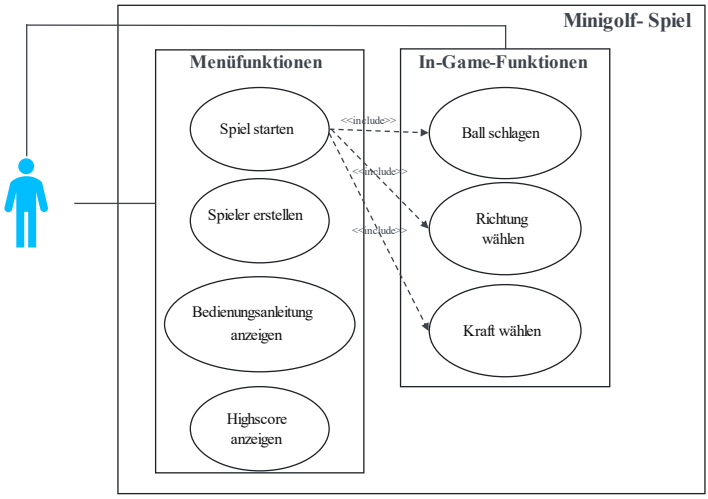

Abbildung 1: Anwendungsfalldiagramm (Use-Case-Diagramm)

Anschließend daran formulieren die Teams in Einzelarbeit oder teamübergreifend die einzelnen sogenannten Tasks[3], womit sie dann den Backlog ihres Project Boards (vgl. Abbildung 2) füllen. Tasks sind einer User Story[4] zugeordnet. Jedes Team spielt danach Planning Poker[5] um die User Stories und Tasks zu priorisieren.

[3] Tasks sind kleinstmögliche Aufgaben, die nicht mehr in Teilaufgaben zerlegbar sind. Tasks werden von einzelnen Teammitglieder bearbeitet [vgl. BR19].

[4] Eine User Story hat immer einen Namen und beschreibt in einem Satz was der Nutzer, „as a“ der Software machen möchte „I want to“, um etwas zu bekommen „so that“ [vgl. BR19].

[5] Beim Planning Poker werden Karten mit einer Einschätzung des Aufwandes von den Teammitgliedern ausgespielt. Anschließend wird eine User Story mit einem Aufwand bewertet [BR19].

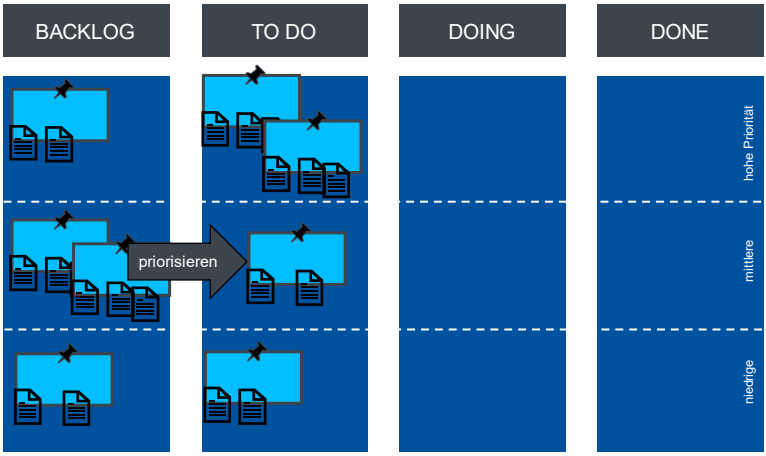

Abbildung 2: Project Board mit Backlog (vgl. Abb. 6 aus [Ka15], Abbildung 3.4 aus [Br19])

In den nächsten Unterrichtsstunden ist der Ablauf fast immer identisch. Zu Beginn finden sich die Teams in Stand-Up-Meetings vor dem Project Board zusammen, besprechen kurz welche Tasks erledigt wurden, welche als nächstes bearbeitet werden und ob Probleme aufgetreten sind. Die restliche Zeit der Unterrichtsstunde wird zur inhaltlichen Bearbeitung der einzelnen Tasks genutzt.

Reflexionsphasen stellen ein zentrales Element des agilen Prozesses dar. Daher wird am Ende der Stunde innerhalb des einzelnen Teams kurz besprochen (1.) was gut lief, (2.) was optimiert werden kann und (3.) was das Team im nächsten Schritt konkret verändern will [Br19]. Das agile Prozessvorgehen sieht auch vor, dass während der Unterrichtsstunde, bedarfs- und anlassbezogen, Stand-Up-Meetings eingefordert werden können. Somit wird ermöglicht, dass sich die Teams direkt und kurzfristig besprechen und sich über im Projektprozess aufgetreten Probleme unmittelbar und zeitnah austauschen können.

Ein weiterer wichtiger Teil der Softwareentwicklung ist das Testen der produzierten Software. Zur methodischen Umsetzung dessen ist es beispielsweise möglich, dass sich Teams gegenseitig und orientiert am Anfang aufgestellten Pflichtenheft testen. So können die Schüler*innen untereinander einen kritisch-konstruktiven Austausch üben. Basis des Feedbacks kann ein Fragebogen zu den einzelnen Funktionalitäten des Minigolf-Spiels sein. Optional steht den Schüler*innen ein didaktisch reduzierter Fragebogen, der in Usabilitystudien für Softwareprodukte eingesetzt wird, für die Testung und ggf. Bewertung zur Verfügung. Hierdurch können die Schüler*innen auch vertiefte Einblicke in die Entwicklung und Forschung im Bezugsfeld der Softwareentwicklung gewinnen. Auch hier ist zu beachten, dass den Schüler*innen die Kriterien des Usability-Fragebogens bereits zu Projektbeginn bekannt sein müssen. Andernfalls ist das Feedback zur Usability von der

Lehrperson in den entsprechenden Kontext zu setzen, da die Evaluationskriterien vorher bekannt und transparent gemacht werden müssen.

Nachdem bei den Programmen der jeweiligen Teams die Basisfunktionen vorhanden sind, findet ein Steakholdergespräch mit der Lehrperson, die weitere leistungsdifferenzierende Anforderungen an die Software stellen kann, statt. Das Minigolf-Spiel kann beispielsweise um einen Multiplayermodus für mindestens drei Spieler*innen erweitert werden. Dabei spielen die Spieler*innen immer nacheinander eine Bahn. Zusätzlich können je nach den zu erwerbenden inhaltsbezogenen Kompetenzen in der jeweiligen Klassenstufe weitere Elemente hinzugefügt werden. Beispielsweise können die inhaltsbezogenen Kompetenzen im Bereich Algorithmen im Informatik-Teil von IMP zusätzlich im Projekt fokussiert werden [Au18]. Die drei Bahnen müssen dann zusätzlich pseudozufällig ausgewählt werden, sodass sich die Reihenfolge beim Spielen beliebig ändern kann. Die eben genannten Anforderungen implizieren weitere Änderungen bzw. Erweiterungen der bisherigen Funktionen. Durch dieses Vorgehen lernen die Schüler*innen einzelne Elemente des Software Development Life Cycle (SDLC) und erhalten einen Einblick in den beruflichen Alltag einer/s Programmentwickler*in (vgl. Abbildung 3). Hierbei wurde die vereinfachte Darstellung des SDLC mit fünf Phasen gewählt. Die Darstellung gibt einen Überblick zur Unterrichtseinheit, die sich über 13 Doppelstunden erstreckt und die Entwicklung des Softwareproduktes allgemein visualisiert. Beim agilen Vorgehen wird dieser Zyklus mehrmals iteriert. Als letzten Punkt dieser Kurzbeschreibung ist bei der Durchführung das möglicherweise wichtigste Kriterium für die Umsetzung von agilen Methoden im Unterricht zu nennen, nämlich die konsequente Durchführung des Projekts. Das Einarbeiten in agile Methoden erfordert ein wenig Unterrichtszeit bei der Einführung und der Vorbereitung des Projekts. Wenn es in den ersten Wochen, oder später im Softwareprojekt zu möglicherweise unerwünschten Situationen kommt und die agile Umsetzung von den Schüler*innen in Frage gestellt wird, sollte die Umsetzung trotzdem bis zum Ende durchgeführt werden. Andernfalls können Probleme bei der klassischen Umsetzung von Softwareprojekten, wie ein nicht lauffähiges Programm, hoher Arbeitsaufwand bei den Teams und Frustration bei einzelnen Schüler*innen auftreten [Br15, Ke15].

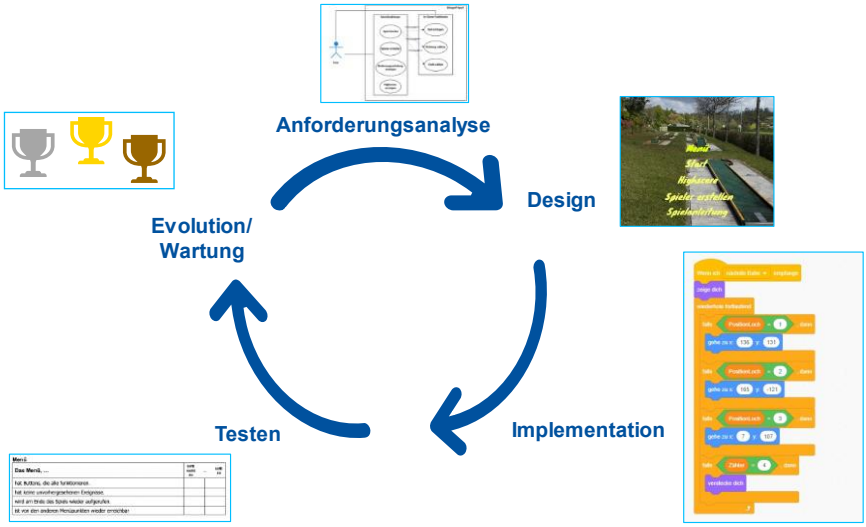

Abbildung 3: Software Development Life Cycle des Minigolf-Spiels (vgl. [Al19])

3 Kreativität im agilen Informatikunterricht

Die Erstellung eines Softwareprodukts wird vielfach als eine der Voraussetzungen für kreativen Informatikunterricht genannt [Ro08] und kann zugleich eine kompetenzförderliche Plattform, sowohl für fachliche als auch überfachliche Kompetenzen bieten. Das Commitment [Br19] der Teammitglieder am „eigenen“ Projekt, im selbstgewählten Team und in einem der Lerngruppe angepassten Projektanforderungsniveau zu arbeiten erfüllt die drei universellen Grundbedürfnisse nach der Selbstbestimmungstheorie: Streben nach Kompetenz, sozialer Eingebundenheit und Autonomie kann damit motivationsfördernd wirken [De00]. Dieses Commitment kann als höher angenommen werden, wenn die Schüler*innen nicht auf eine vorgefertigte funktionale Lösung durch das Ausführen kleinschrittiger Arbeitsaufträge hinarbeiten, sondern in einem kreativen (agilen) Prozess eigene Ideen einbringen können. Durch Differenzierungsmöglichkeiten und Offenheit in späteren Aufgabenstellungen ist es den Schüler*innen beim Minigolf-Spiel als Softwareprojekt möglich, die Aufgaben aktiv mitzugestalten. Ob sie z. B. Sounds, Wind, Hindernisse oder andere Spielelemente und -funktionen hinzufügen, können sie selbstständig entscheiden. Um neue Ideen für weitere Funktionen des Programms zu generieren, können zusätzlich Kreativitätstechniken wie Brainstorming, Mindmaps oder die Alphabet-Technik Eingang in den agilen Projektunterricht finden. Den Schüler*innen keinen Lösungsweg oder Muster

vorzugeben, ist damit eine zentrale Anforderung in der Aufgabenstellungen und soll das kreative Arbeiten von Schüler*innen sowie eigene Freiräume für Kreativität und Agilität ermöglichen [Ro08]. Als letzten Punkt sind die Gestaltungsmöglichkeiten für Schüler*innen zu nennen. Bis auf die Limitationen durch Anwendungsfalldiagramme und den curricularen Vorgaben (z. B. Bildungsplan IMP [Au18]) zu den inhaltsbezogenen Kompetenzen, sowie der vorgegebenen Programmierumgebung Scratch, können die Schüler*innen das Minigolf-Spiel, einschließlich Menübedienung, individuell gestalten.

Agile Methoden können in vielfältigen informationstechnischen Projekten und fächerübergreifenden Kontexten genutzt werden. Aspekte des kreativen Informatikunterrichts wie z. B. die Ideenanregung durch komplexe Probleme oder ein Anknüpfen an die (über)fachliche Erfahrungswelt der Schüler*innen [Ro08] kann in eine agile Umsetzung eingebunden werden. Die offene und trotzdem, durch die einzelnen agilen Methoden gut strukturierte, Bearbeitung des Projekts kann, so die Annahme, die Kreativität der Schüler*innen und ihre grundlegenden Einstellungen im Informatikunterricht wünschenswert fördern. Wünschenswert dahingehend, dass sie arbeitsfähige Strukturen, die u.a. auch in Unternehmen umgesetzt werden, lernen, als auch die konstruktiven und innovativen Aspekte der Gestaltung der Informatik applizieren können.

Heterogene Produktergebnisse sind beim methodischen Einsatz agiler Methoden in Softwareprojekten eine entscheidende Facette. Beispielsweise entscheidet sich jedes Team individuell für die Visualisierung der einzelnen Hintergründe und Kostüme. Die Lehrperson hat die Möglichkeit das Anwendungsfalldiagramm in einer Plenumsphase mit den Schüler*innen zusammen zu erstellen, oder bei entsprechendem Fachwissen und prozessbezogenen Kompetenzen, die Erstellung des Diagramms den Schüler*innen zu überlassen. Für den letztgenannten Fall ermöglicht es die Transparenz des Project Boards aus der Perspektive der Lehrperson, sicherzustellen, dass durch das Projekt alle fachlichen Kompetenzen abgedeckt werden. Das weitere Vorgehen, wie die Formulierung, das Bewerten bzw. Einstufen, das Abarbeiten, Abändern, oder Hinzufügen der einzelnen Tasks nach der Besprechung des Anwendungsfalldiagramms befähigt das Team ihr Softwareprodukt individuell zu gestalten. Zusammengefasst mit einem Satz aus dem agilen Schulmanifest: „Kreativität und Reagieren auf Gelerntes sind wichtiger als das Befolgen eines starren Plans“ [Br18]. Wichtig beim Einstufen der einzelnen Tasks ist damit, dass in der Unterrichtsverlaufsplanung ausreichend Zeit für Kreativität bei der Gestaltung eingeplant wird.

Der Projektunterricht mit agilen Methoden ist schüler- und nicht lehrerorientiert ausgerichtet. Die Lehrperson fungiert, wie bei den Anforderungen an die Lehrperson für kreativen Informatikunterricht als Coach oder Moderator [Ro08]. Im Vordergrund der Lehrperson stehen, neben dem üblichen Classroom-Management, das bedarfsgerechte Helfen und Unterstützen bei größeren anhaltenden Problemen und ggf. das Motivieren und Inspirieren der Schüler*innen. Um einer möglichen Leistungsheterogenität innerhalb der Klasse gerecht zu werden, bietet die agile Projektmethode damit generell gute Voraussetzungen, um die Individualisierung im Unterricht zu fördern. Strukturgebende Aufgaben der Lehrperson sind zudem die agile Umsetzung, das Monitoring der Arbeit und die Ak-

tualisierung der Project Boards. Steakholdergespräche, oder Vorträge, die zusätzlich eingefordert werden können. Außerdem kann die Lehrperson an Stand-Up-Meetings und Reflexionen als Beobachtende oder im Extremfall intervenierende Person (z.B. durch das Hinweisen auf bestimmte agile Werte wie bspw. Respekt) teilnehmen, um Eindrücke von der Arbeit im einzelnen Team zu bekommen.

4 Ziele des Workshops

Die Ziele des bereits mehrfach in Baden-Württemberg umgesetzten Workshops für Lehrpersonen im Profilfach IMP sind, dass die Teilnehmenden das agile Schulmanifest [Br19], Werte in agilen Projekten, den gesamten agilen Prozess, sowie die verschiedenen agilen Methoden kennenlernen [ebd.]. Darauf aufbauend sollen die Teilnehmenden ein handlungsbezogenes Wissen zum Zyklus des agilen Prozesses in reduzierter Form in der Schülerperspektive erwerben. Anschließend daran, werden die Vor- und Nachteile beim Einsatz von agilen Methoden und ihrer digitalen Umsetzung im Unterricht und den individuellen schulischen Voraussetzungen der Teilnehmenden diskutiert und reflektiert.

5 Pilotierung und Erkenntnisse

Im Rahmen einer ersten Pilotierungsstudie im Jahr 2021 nahmen N = 37 Informatik-Lehrpersonen teil. Die Lehrerweiterbildung fokussierte *Agile Methoden* und *digitale Tools zur unterrichtlichen Umsetzung*. In einer Vorabumfrage zur Weiterbildungsmaßnahme mit dem Ziel, die Weiterbildung eng an den individuellen fachlichen Voraussetzungen und Interessen der Lehrpersonen zu orientieren, gaben 85 % der Befragten an, zuvor nichts über Agile Schule bzw. agile Unterrichtsmethoden gehört zu haben. Auch wenn die Befragungsergebnisse nach der Lehrerweiterbildung auf Selbsteinschätzungen der Teilnehmenden beruhen und mögliche Verzerrungen (z. B. soziale Erwünschtheit) beinhalten, so stimmen diese positiv im Hinblick auf einen tatsächlichen Einsatz der Methode im Informatikunterricht durch die Schulungsteilnehmer*innen. Demnach gaben 80,8% der Befragten haben an, dass sie sich zutrauen, die Unterrichtseinheit Minigolf-Spiel mit Scratch durchzuführen und dabei agile Methoden anzuwenden. 65,4% haben in der Weiterbildungseinheit, entsprechend ihren Angaben, viel Neues im Bereich Agile Schule dazu gelernt (vgl. Abbildung 4). Die Tatsache, dass 85 % der befragten Lehrpersonen vor der Weiterbildung angeben, die Methode bisher nicht im Unterricht verwendet zu haben, zeigt zudem, dass agile Methoden noch keinen flächendeckenden Einzug in den IMP Unterricht an Gymnasien in Baden-Württemberg gehalten haben.

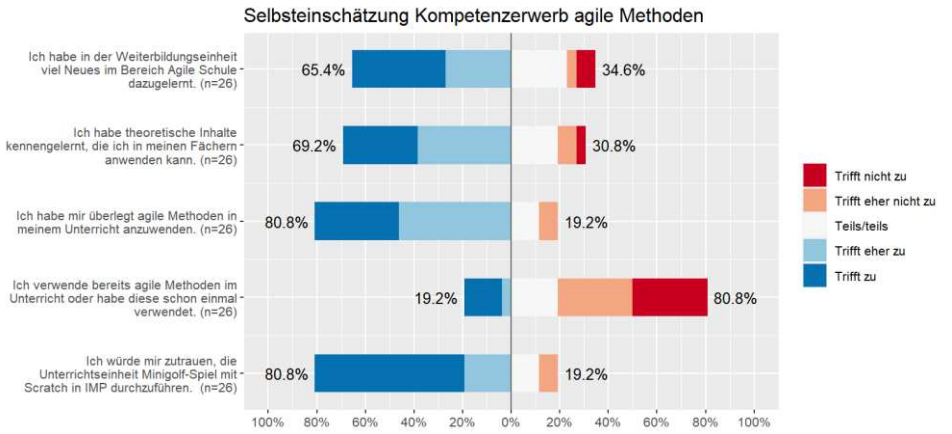

Abbildung 4: Auszug zu den Evaluationsergebnissen der Weiterbildungseinheit „Agile Schule“

6 Fazit und Limitationen

Zusammengefasst bietet der Einsatz von agilen Methoden im Informatikunterricht das Potential zur Bereitstellung adaptiver Lehr- und Lernmöglichkeiten, um im Informatikunterricht der Heterogenität der Schülerschaft gerecht zu werden und fokussiert schülerorientiertes Lernen zu unterstützen, sowie Tandem-Programmierung und individualisiertes Lernen zu adressieren. Durch das Prototyping, ein in Teilen lauffähiges Produkt zu erhalten und dieses in einem inkrementellen Prozess stetig weiterzuentwickeln, eine positive Fehlerkultur im Unterricht zu pflegen sowie fachübergreifende Kompetenzen wie Kommunikationsfähigkeit, Selbstorganisation und Kooperation prozessbezogen zu fördern, sind weitere mögliche Vorteile. Weitere Chancen bieten die hohe Unterrichtstransparenz durch das Project Board, die der Lehrperson auch eine angemessene Bewertung einzelner Schüler*innenleistungen ermöglicht. Eine Fokussierung auf agile Werte, wie Commitment, Respekt, Fokussierung und strukturiertes Feedback können förderlich für den Unterrichtsverlauf sein, indem typische Probleme im Rahmen von Gruppenarbeiten, wie beispielsweise soziales Faulenzen, Trittbrettfahren oder Gimbel Effekte geringgehalten werden [Ha75]. Wie Eingangs bereits berichtet, haben agile Methoden ihren Ursprung in unternehmerischen und wirtschaftlichen Kontexten. Vor dem Hintergrund der Digitalisierung der Arbeits- und Berufswelt [Zi17] sowie der Forderung nach Agilität von Personen, Organisationen, Strukturen und Prozessen in ebendieser Arbeits- und Berufswelt, erscheint es wünschenswert entsprechend hierfür notwendige Kompetenzen bereits in der Schule zu fördern.

Flankiert mit den Erfahrungen aus Praxisberichten [Br15, Br19, Ka15, Gö11, Ke15] ist der Einsatz von agilen Methoden kein Patentrezept für einen guten Projektunterricht, kann

jedoch dessen Umsetzung, insbesondere durch seine strukturgebende und gleichzeitig flexible anpassungsfähige Methodik, unterstützen. Die Evidenz zu den tatsächlichen Wirkungseffekten des methodischen Vorgehens im Unterricht und Kontext der agilen Projektmethode im Hinblick auf die Schüler*innen, ist nach den uns vorliegenden Daten gering und umfasst mehrere Forschungsdesiderata. Sowohl im Hinblick auf die kognitiven, motivationalen als auch affektiven Merkmale von Lernenden, wären im Bezugsfeld Interventionsstudien interessant, um evidenzbasierte Aussagen zur tatsächlichen Wirkung treffen zu können.

Literatur

[Al19] Alpar, P., Alt, R., Bensberg, F., Weimann, P. (2019). Phasenmodelle in der Softwareentwicklung. In: Anwendungsorientierte Wirtschaftsinformatik. Springer Vieweg, Wiesbaden. https://doi.org/10.1007/978-3-658-25581-7_13

[Au18] Autorengruppe des Ministeriums für Kultus, Jugend und Sport Baden-Württemberg (2018). Bildungsplan zum Profilfach Informatik, Mathematik, Physik (IMP). BILDUNGSPLAN DES GYMNASIUMS. http://bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_GYM_IMP.pdf

[Br15] Brichzin, P. (2015). Agile Softwareentwicklung. Erfahrungsbericht eines Oberstufenprojekts im Wahlpflichtunterricht. In: Gallenbacher, J. (Hrsg.), Informatik allgemeinbildend begreifen. Bonn: Gesellschaft für Informatik e.V.. 63-72.

[Br19] Brichzin, P., Kastl, P., & Romeike, R. (2019). Agile Schule: Methoden für den Projektunterricht in der Informatik und darüber hinaus. Hep Verlag.

[Co05] Coram, M. & Bohner, S. (2005). The impact of agile methods on software project management. 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05). 363-370.

[De00] Deci, Edward. L. & Ryan, Richard. M. (2000). The „What“ and „Why“ of Goal Pursuits: Human Needs and the Self-Determination of Behavior. In: Psychological Inquiry 11(4). 227-268.

[Di15] Dietrich, L., Gramm, A., Kastl, P. & Romeike, R. (2015). Ein Bild vom Wesen der Softwareentwicklung: Erfahrungen aus zwei agilen Projekten. In: Gallenbacher, J. (Hrsg.), Informatik allgemeinbildend begreifen. Bonn: Gesellschaft für Informatik e.V.. 83-92

[Gö11] Göttel, T., (2011). Agiler Informatikunterricht: Soziale Aspekte der professionellen Softwareentwicklung im Schulunterricht erfolgreich erfahrbar machen. In: Thomas, M. (Hrsg.), Informatik in Bildung und Beruf – INFOS 2011 – 14. GI-Fachtagung Informatik und Schule. Bonn: Gesellschaft für Informatik e.V.. (S. 37-46).

[Ha75] Hackman, J.R., & Morris, C.G. (1975). Group tasks, group interaction process, and group performance effectiveness: A review and proposed integration. Advances in Experimental Social Psychology, 8, 45-99.

[Ka15] Kastl, P., März, S. & Romeike, R., (2015). Agile Softwareentwicklung im Informatikunterricht. Ein Best-Practice-Beispiel am Spiel “Pengu'. In: Gallenbacher, J. (Hrsg.), Informatik allgemeinbildend begreifen. Bonn: Gesellschaft für Informatik e.V.191-200.

[Ke15] Kerber, L., Kastl, P. & Romeike, R., (2015). Agiler Informatikunterricht als Anfangsunterricht. In: Gallenbacher, J. (Hrsg.), Informatik allgemeinbildend begreifen. Bonn: Gesellschaft für Informatik e.V.. (S. 211-218).

[Ma10] Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E.(2010). The Scratch Programming Language and Environment. ACM Trans. Comput. Educ. 10(4), 15 pages. https://doi.org/10.1145/1868358.1868363

[Ri09] Rico, D. F. & Sayani H. H. (2009). Use of Agile Methods in Software Engineering Education. 2009 Agile Conference.174-179.

[Ro08] Romeike, R. (2008). Kreativität im Informatikunterricht. Potsdam, Universität Potsdam (Doctoral dissertation, Dissertation).

[Sc13] Schumacher, C., Rengstorf, F., & Thomas, C. (Eds.). (2013). Projekt: Unterricht: Projektunterricht und Professionalisierung in Lehrerbildung und Schulpraxis. Vandenhoeck & Ruprecht.

[Tr11] Traub, S. (2011). Selbstgesteuert lernen im Projekt? Anspruch an Projektunterricht und dessen Bewertung aus Sicht von Lehrenden und Lernenden. Zeitschrift für Pädagogik, 57 (1), 93–113.

[Wa09] Wasmann-Frahm, A. (2009). Kompetenzentwicklung durch Projektunterricht. Unterrichtswissenschaft, 37 (1), 77-96.

[Wi03] Williams, L. & Cockburn, A. (2003). Agile software development: it's about feedback and change. in Computer, 36 (6), 39-43.

[Zi17] Zinn, B. (2017). Digitalisierung der Arbeit – Kompetenzerwartungen des Beschäftigungssystems und didaktische Implikationen. In: Bonz, B., Schanz, H. & Seifried, J. (Hrsg.): Berufsbildung vor neuen Herausforderungen – Wandel von Arbeit und Wirtschaft, Berufsbildung konkret, Band 13, Baltmannsweiler: Schneider Verlag Hohengeheren GmbH, S. 163-176.