



Agile Schule –

Weiterentwicklung des Projektunterrichts in der Informatik und darüber hinaus

PETRA KASTL – PETER BRICHZIN – RALF ROMEIKE

In Start-ups geht heute fast nichts mehr ohne Agile Softwareentwicklung. Im Folgenden geben wir einen Überblick über einen an die Schule angepassten agilen Prozess und sein pädagogisches Potenzial. Praktiken für den Einstieg werden detaillierter erläutert und um Varianten schulischer Umsetzung und praxisrelevante Hinweise ergänzt. Exemplarisch wird auf das Zusammenspiel von agilen Werten und Praktiken eingegangen. Die Erfahrungen zeigen, dass ein Übertragen der agilen Schule aus der Informatik in andere Schulfächer sinnvoll und vielversprechend ist.

1 *Agiler Projektunterricht – Praxis und Forschungsstand*

Traditionell verlaufen Schulprojekte, ähnlich wie professionelle Projekte, in sequentiell aufeinanderfolgenden Phasen (FREY, 1983), (SCHUBERT & SCHWILL, 2011). Am Anfang steht die Analyse der gestellten Aufgabe, aus der zunächst eine Anforderungsdefinition entwickelt wird. Anschließend folgt die Entwurfsphase und erst dann erfolgt die Umsetzung, die am Ende gegen die Anforderungen getestet wird.

In schulischen Projekten stellen sich sequentiell organisierte Projekte jedoch oft als schwierig heraus, da die Schüler/innen zu Beginn in der Regel nicht über die nötigen Erfahrungen und „Softskills“ verfügen, die Voraussetzung für eine erfolgreiche Projektarbeit sind (HARTMANN, NÄF & REICHERT, 2007), (MEERBAUM-SALANT & HAZZAN, 2010), (SCHUBERT & SCHWILL, 2011), (WEIGEND, 2005).

Konkret zeigen sich in Schulprojekten in der Informatik ähnliche Probleme wie in professionellen Softwareprojekten: die Projektlaufzeiten müssen oft verlängert werden, Tests finden dann meist unter großem Zeitdruck statt und Qualitätsmängel oder unfertige Produkte sind häufig die Folge. Auch die anfangs an das Produkt gestellten Anforderungen erfüllen trotz sorgfältiger Analyse am Ende oft nicht (mehr) die Bedürfnisse der Kunden. Die Projektbeteiligten beschreiben die Situation immer wieder als demotivierend und zu bürokratisch.

Als Alternative zu sequenziellen Vorgehensweisen haben sich in der Arbeitswelt seit den 2000er Jahren agile Methoden in kleinen wie in großen Unternehmen rasch verbreitet. So setzten nach einer jährlich zum Stand agiler Softwareentwicklung durchgeführte Umfrage im Jahr 2018 bereits 97 % aller Unternehmen in der IT agile Methoden ein (VersionOne, 2020). Oft sind agile Teams besonders motiviert, arbeiten fokussiert, pflegen einen wertschätzenden Umgang miteinander und sehen Fehler als eine wichtige Möglichkeit etwas zu lernen. Sie bestimmen ihren Weg zum Ziel selbst, reflektieren diesen regelmäßig und begreifen Veränderung als Chance. So verändern agiles Denken und Handeln die Unternehmenskultur, da mit ihnen eine Reihe zentraler Werte, wie offene Kommunika-

tion auf Augenhöhe und Selbstverantwortung im Team verbunden sind. Deshalb wollen Teams und Kunden, die einmal agile Luft geschnuppert haben, in der Regel nicht mehr zurück.

Das agile Manifest

Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge.

Funktionierende Software ist wichtiger als umfassende Dokumentation.

Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlungen.

Reagieren auf Veränderungen ist wichtiger als das Befolgen eines Plans.

Abb. 1. Manifest für agile Softwareentwicklung (2001)

In agilen Projekten entwickeln Teams iterativ und inkrementell Prototypen. Sie erproben und überprüfen diese Zwischenergebnisse und holen sich regelmäßig Feedback. Konkrete Praktiken wie User-Stories und Project-Boards oder Stand-up-Meetings und Team-Retrospektiven unterstützen die Teams bei der Projektorganisation und -durchführung. Ein Teil des spezifischen Methodenrepertoires kann auch den Unterricht bereichern. Bei Lehrkräften und Schüler/innen sorgt agiles Vorgehen fast immer rasch für viele positive und motivierende Effekte, denn es gelingt den Schüler/innen in der Regel mit Hilfe der Praktiken deutlich besser, sich selbst zu organisieren, in gemeinsamer Verantwortung tolle Produkte zu erzeugen und erfreuliche fachliche und soziale Lernerfolge zu erzielen.

Betrachtet man als Lehrkraft die agilen Methoden genauer, offenbart sich ihr didaktisch-pädagogisches Potenzial für die Unterrichtsgestaltung schnell: Die Praktiken und Techniken fördern und fordern Kommunikation im Team, unterstützen die Strukturierung der inhaltlichen Arbeit in gut bearbeitbare Teilmuster und erleichtern es den Schüler/innen, ihren Lernprozess selbst mitzugestalten. Kurze Zyklen sorgen schnell für erste motivierende Erfolgserlebnisse. Das Ende eines Zyklus regt aber auch dazu an, innezuhalten und nachzudenken über das Produkt und die Zusammenarbeit: Was haben wir bis jetzt erreicht? Sind wir damit auf dem richtigen Weg? Was sind unsere nächsten Schritte? Hierdurch wird Lernen auch zu einem wichtigen Prozessbestandteil.

Verschiedene praxisorientierte Untersuchungen im Schulkontext beschreiben positive Erfahrungen bei der Anwendung einzelner agiler Elemente. Zwei theoretisch sehr gut fundierte Studien stellen adaptierte Vorgehensmodelle vor: MEERBAUM-SALANT & HAZZAN (2010) entwickelten eine „agile constructionist mentoring methodology“ in der sie agile Werte nutzen, um Lehrkräfte bei der Betreuung von Softwareprojekten zu unterstützen. ROMEIKE & GÖTTEL (2012) wiederum stellen in ihrem adaptierten agilen Vorgehensmodell die Perspektive der Lernenden ins Zentrum. Auf der Fachtagung „Informatik und

Schule – INFOS 2015“ der Gesellschaft für Informatik e. V. in Darmstadt wurden Erfahrungen mit dem schülerzentrierten Modell durch verschiedenen Berichte aus der Praxis in unterschiedlichsten Unterrichtsettings illustriert (BRICHZIN, 2015), (BRICHZIN & RAU, 2015), (DIETRICH, GRAMM, KASTL & ROMEIKE, 2015), (KASTL, MÄRZ & ROMEIKE, 2015), (KERBER, KASTL & ROMEIKE, 2015), (KIESMÜLLER, KASTL & ROMEIKE, 2015). Seitdem wurden das zugrundeliegende Prozessmodell sowie die Unterrichtstechniken und Praktiken weiterentwickelt und hinsichtlich ihrer Praxiswirksamkeit evaluiert (KASTL, KIESMÜLLER & ROMEIKE, 2016), (KASTL & ROMEIKE, 2018). Hintergrundwissen zu agilen Methoden, Berichten aus der Praxis sowie grundlegende Methoden und Methoden für Fortgeschrittene mit detaillierten Ausführungen zum professionellen Hintergrund, Varianten schulischer Umsetzung, mögliche Stolpersteine Tipps und Tricks ist bei BRICHZIN, KASTL & ROMEIKE (2019) zu finden.

Im Folgenden geben wir einen Überblick über den agilen Prozess und erläutern dann einige Praktiken detaillierter, die für den Einstieg in agile Projektarbeit empfehlenswert sind. Anschließend gehen wir exemplarisch auf das Zusammenspiel von Werten und Praktiken ein, das der gemeinsame Kern aller agilen Methoden ist. Abschließend skizzieren wir zwei konkrete Beispiele zu agilen Methoden außerhalb der Informatik.

2 *Einstieg in einen iterativen Prozess*

Statt die Projektphasen nur einmal sequenziell zu durchlaufen, werden diese in agilen Projekten zyklisch in kurzen, festen Zeitfenstern, den sogenannten Iterationen oder Sprints wiederholt bearbeitet. Die Iterationen folgen direkt aufeinander, sind alle gleich lang und werden jeweils mit einem lauffähigen, inkrementell wachsenden Zwischenergebnis abgeschlossen. Im Prozess verankert sind auch klare Kommunikationsstrukturen sowie eine Visualisierung, die für Transparenz sorgen.

Das iterative Vorgehen ermöglicht es den Lernenden, sich mit dem Projekt weiterzuentwickeln und Selbstvertrauen in der Projektdurchführung zu gewinnen. So wird die erste Iteration erfahrungsgemäß von Schüler/inne/n ebenso wie von Lehrkräften auch mal als „chaotisch“ beschrieben. Die Struktur und die Praktiken unterstützen die Schüler/innen jedoch dabei, Selbstorganisation und Kollaboration zu erlernen. Erste Erfolge sind rasch sichtbar, motivieren die Schüler/innen und helfen ihnen, die Zielvorstellungen regelmäßig zu überprüfen. Auch die Lehrkraft kann nun regelmäßig Feedback geben und, falls gewünscht, neue Lerninhalte passend zur Produktentwicklung gestaffelt in die Projektarbeit integrieren.

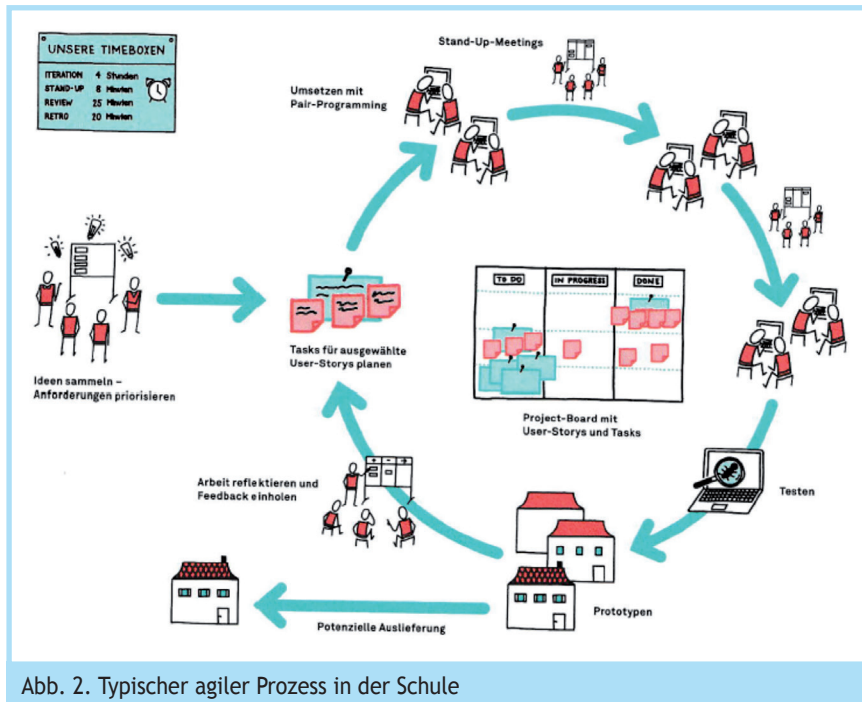


Abb. 2. Typischer agiler Prozess in der Schule

Agile Projekte beginnen mit einer vorangestellten Planungsphase, in der die Produktvision in den sogenannten User-Stories konkretisiert wird. Anschließend werden die User-Stories priorisiert. Falls es für den Projekterfolg nötig ist, können User-Stories und ihre durch die Priorität festgelegte Arbeitsreihenfolge jeweils nach einer Iteration geändert und ergänzt werden. Nach der vorangestellten Planung wird in Iterationen gearbeitet, die in Schulprojekten meist zwei bis vier Stunden lang sind. Die Iterationen laufen nun wie folgt ab: In der Iterationsplanung legt das Team die zu bearbeitenden User-Stories der Iteration fest und plant dazu die für ihre Umsetzung zu erledigenden Arbeitspakete, die sogenannten Tasks. Während der Iteration arbeitet das Team daran ungestört, fokussiert und selbstbestimmt. Das Project-Board gibt dabei jederzeit einen Überblick über den Arbeitsstand. Gegen Ende der Iteration führt das Team die arbeitsteilig erstellten Umsetzungen zusammen und testet das Zwischenergebnis. Am Ende kann sich das Team Feedback zum aktuellen Prototyp holen und den Arbeitsprozess reflektieren (Abb. 2).

In professionellen Projekten beginnt während der Arbeitsphase jeder Arbeitstag mit einem zeitlich in der Regel auf 10 Minuten begrenzten Stand-up-Meeting. Hier tauscht sich das Team über den aktuellen Projektstand aus. In der Schule beginnt entsprechende jede Unterrichtseinheit mit einem Stand-up-Meeting der Teams von fünf bis acht Minuten.

Üblicherweise wird der agile Prozess an das individuelle Projekt und die Schülergruppe angepasst. Das betrifft die Iterationslänge, den Ablauf der Iteration, die verwendeten Praktiken sowie deren konkrete Umsetzung. So hat eine Lehrkraft beispielsweise für Programmieranfänger/innen den Ablauf einer Iteration geändert. Die Teams planen zweimal Tasks jeweils für eine Arbeitsphase von ca. 15 bis 20 Minuten. Die Integration

der Arbeitsergebnisse erfolgte stets nur am Ende der Doppelstunde. Einzelne Praktiken können auch zeitversetzt eingeführt und Praktiken, die die Schüler/innen nicht (mehr) optimal unterstützen können verändert oder weggelassen werden.

In der praktischen Umsetzung bieten sich agile Spiele (z.B. das Ball Point Game, die Marshmallow-Challenge etc.) als methodischer Einstieg in agiles Handeln und Denken an. Beim Ball Point Game beispielsweise hat das Team die Aufgabe, so viele Bälle wie möglich in einer bestimmten Zeit durch die „Ballmaschine“ zu schleusen. Das Team spielt nach zwei Minuten Planung vier Runden, die jeweils aus 2 min Spielzeit und 1 min Reflexion bestehen. Dabei befolgt es vier einfache Regeln und entscheidet die übrige Spielgestaltung selbst. Danach wertet das Team das Spiel und die erreichten Leistungen aus. Diese agilen Spiele sprechen das Soziale und Emotionale stark an. Die positiven Gefühle bieten für die gesamte Projektphase Anknüpfungspunkte und erzeugen Motivation. Um den Einstieg zu erleichtern können einzelne Praktiken wie User-Stories oder Pair-Programming schon im Unterricht eingeführt werden. Bei Zeitproblemen werden nicht erledigte User-Stories in die nächste Iteration geschoben und die aktuelle Iteration mit einem getesteten und lauffähigen Prototyp abgeschlossen.

User-Story – Zentrales Planungselement auch didaktisch nutzen

User-Stories beschreiben Anforderungen an eine Software aus Sicht des Kunden. Sie bestehen aus wenigen Sätzen und sind in Alltagssprache formuliert.

Sie helfen den Schüler/innen, die umfangreichen fachlichen Funktionalitäten des Gesamtsystems in überschaubare und damit gut handhabbare Teile zu gliedern und geben damit dem Softwareentwicklungsprozess eine klare Struktur. Hier können sich auch fachlich schwächere Schüler/innen einbringen. So werden User-Stories zur Basis für die Kommunikation über die Teilziele, Lehrkräfte können mit den Teams darüber ins Gespräch kommen und können ggf. frühzeitig lenkend eingreifen.

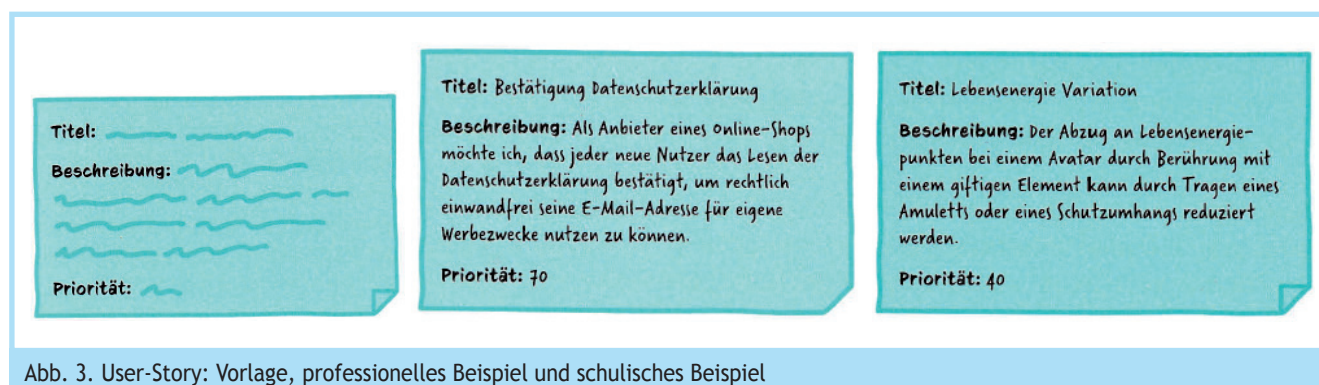


Abb. 3. User-Story: Vorlage, professionelles Beispiel und schulisches Beispiel

Eine User-Story beantwortet die Frage, wer sich was wozu wünscht. In der Schule genügt in der Regel das „Was“ zu formulieren. Dabei soll eine User-Story so klein sein, dass das Team später in einer Iteration mehrere davon umsetzen kann. Die Gesamtheit der User-Stories beschreiben das Produkt aus Kun-

densicht. Sie helfen den Aufwand und den Projektfortschritt abzuschätzen und zu kontrollieren. Die Bearbeitungsreihenfolge der User-Stories legt der Kunde zusammen mit dem Team durch eine Priorisierung fest. In agilen Projekten stellt das Reagieren auf Veränderungen einen wesentlichen Wert dar, sodass User-Stories oder ihre Priorisierung nach jeder Iteration abgeändert werden können. Dies geschieht in einer offenen und wertschätzenden Absprache. In der Schule sind die Schüler/innen in der Regel ihre eigenen Kunden. Sie bestimmen das grobe Ziel ihres Projekts selbst und differenzieren es dann so weit aus, dass sie User-Stories auf Karteikarten festhalten können. Auch die Priorisierung sowie spätere Änderungen entscheidet und verantwortet das Team selbst.

In Form von didaktischen Varianten können User-Stories auch eingesetzt werden, um den Lernprozess zu strukturieren, indem diese Varianten über eine entsprechende Priorisierung in den Projektablauf eingeflochten werden: Sie erhalten einen Zeitrahmen und eine Visualisierung innerhalb der Iterationsplanung und ihre erfolgreiche Bearbeitung wird gefeiert.

Als Varianten verwenden wir einerseits *Modeling-Stories*, die einen konkreten Modellierungsauftrag z.B. die Erstellung eines Klassendiagramms enthalten. *Modeling-Stories* können von den Schüler/innen eines Teams bei Bedarf in ihre Planung aufgenommen werden. Ebenso können sie von der Lehrkraft vorgegeben werden, um eine Reflexion über Strukturen anzustoßen.

Eine weitere Variante sind *Student-Stories*, die Lernaufgaben darstellen. In der Regel streut die Lehrkraft *Student-Stories* in die Planung ein. Abhängig von der Lerngruppe oder der Binnendifferenzierung kann die Lehrkraft die Bearbeitung der *Student-Story* komplett in die Verantwortung des Teams geben oder über konkrete Tasks und kleinschrittiges Lernmaterial stark unterstützen.

Erfahrungsgemäß werden User-Stories leicht zu groß und mitunter sieht man der Formulierung nicht die Intention der Schüler/innen an. Daher ist es sinnvoll als Lehrkraft oder als fiktiver Kunde während der Planung immer wieder mit den Teams ins Gespräch zu kommen. Bei längeren Projekten kann man die Schüler/innen User-Story für ein erstes Etappenziel schreiben lassen.

3 *Transparenz, Kommunikation, Zielstrebigkeit – Wertorientierung in der selbstorganisierten Arbeitsphase*

Agile Methoden sind vielfältig in Zahl und Ausgestaltung und auch wenn die Praktiken am sichtbarsten sind, etwa die vielen bunten Klebezettel am Board, so definiert sich agiles Vorgehen nicht durch das Verwenden von Praktiken oder den iterativen Verlauf. Vielmehr ist der gemeinsame Kern eine Reihe an Werten, an denen sich agile Teams – und auch die Lehrkraft – orientieren, wenn sie Entscheidungen treffen auf ihrem selbstgestalteten Weg zum gesetzten Ziel. Die Praktiken wiederum helfen den Teams, nach den Werten zu handeln. Zu den zentra-

len Werten gehören Kommunikation und Einfachheit, aber auch Transparenz, Selbstorganisation und Feedback sind wichtig. Daneben kann je nach Schwerpunktsetzung auch Fokus (widme deine Aufmerksamkeit ungeteilt einer konkreten Aufgabe), Mut oder Commitment (verpflichte dich als Teil deines Teams, die verabredeten Aufgaben zu erfüllen), betont werden. Exemplarisch wird dies an drei Praktiken erläutert.

3.1 Das Project Board

Das Project-Board visualisiert nicht nur die Ziele und Aufgabenpakete, sondern auch den aktuellen Arbeitsstand (und damit auch den Projektfortschritt) sowie die Arbeitsverteilung. Die Visualisierung sorgt für Transparenz, unterstützt die Selbstorganisation, zeigt das Commitment des Teams bzw. der einzelnen Teammitglieder, fordert Einfachheit bei der Planung ein und sorgt für ein fokussiertes Arbeiten. Es ist neben den Prototypen eine Grundlage für Feedbackgespräche. Agil arbeitende Projektbeteiligte berichten dazu: „Wir arbeiten jetzt an weniger Sachen, aber die werden fertig.“ und „Da weiß man immer gleich, wo man momentan steht und was als Nächstes drankommt. Da muss man nicht so viel im Kopf haben und verzettelt sich nicht so leicht.“

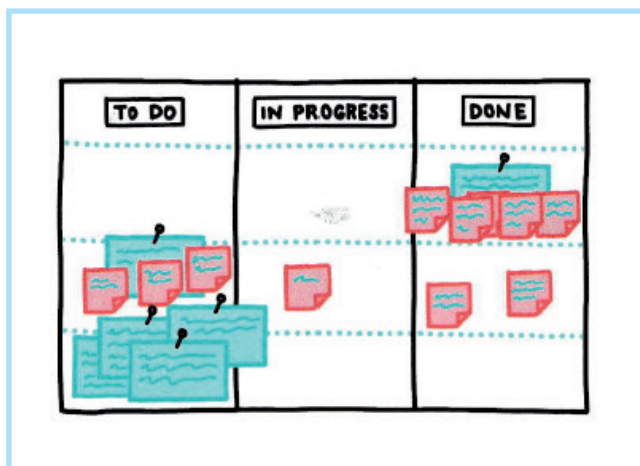


Abb. 4. Einfaches Project-Board

Im einfachsten Fall besteht das Projekt-Board aus drei Spalten. Links in der Spalte „To Do“ hängen nach Priorität absteigend geordnet die User-Stories (Karteikarten) sowie die dazu geplanten Tasks (Klebezettel), falls die User-Story schon spezifiziert wurde. Während der Arbeitsphase wählt jedes Teammitglied bzw. jedes Paar einen Task aus, schreibt den Namen darauf und hängt den Task in die mittlere Spalte „In Progress“ und nachdem er erledigt ist, in die rechte Spalte „Done“. So ist auf einen Blick erkennbar, wer gerade für welches Arbeitspaket verantwortlich ist und welche Tasks und User-Stories bereits erledigt sind. Das aktive Umhängen erledigter Aufgaben motiviert die Schüler/innen, ist Anlass das Erreichte zu feiern und bietet den Lehrkräften eine Grundlage für Hilfestellungen.

Je nach Bedarf kann man zusätzliche Bereiche einrichten, um beispielsweise das Ende einer Iteration sichtbar zu machen. Sinnvoll können auch Bereiche sein für konkrete Fragen der Schüler/innen, für spontane Ideen, für zurückgestellte

User-Stories, bei denen es noch Probleme gibt oder auch für Fehler, die noch zu lokalisieren und zu korrigieren sind. Die dort abgelegten Themen können nach der Iteration ggf. im Team oder mit der Lehrkraft besprochen werden. Ebenso möglich sind zusätzliche Spalten etwa für das Testen. Aber auch hier gilt nach dem „KISS“-Prinzip (keep it small and simple): „Führe nur das ein, was einen Mehrwert hat“.

Die Freude am Umhängen von Zetteln, der einfache Umgang und die Übersichtlichkeit bewegen auch professionelle agile Teams oft dazu ein physisches Board zu installieren. Aus denselben Gründen eignen sich physische Boards mit echten Zetteln auch in der Schule besonders gut. Dennoch mag ein digitales Board z.B. in Oberstufenprojekten sinnvoll sein, damit die Schüler/innen auch zu Hause darauf zugreifen und digitale Dokumente direkt mit einbinden können.

3.2 Das Stand-up-Meeting

Stand-up Meetings helfen der Gruppe bei der Selbstorganisation. Wie alle anderen Besprechungsformen setzt das Stand-up-Meeting neben der Fokussierung auf Offenheit, Respekt und den Mut, etwa Fehler oder eine schwache Arbeitsleistung anzusprechen. Es fördert und erfordert eine fachlich korrekte Kommunikation, bringt das Team auf einen gemeinsamen Informationsstand und gibt der Lehrkraft einen Einblick in die Teamarbeit. Das tägliche Stand-up-Meeting der Profis halten die Teams in der Schule jeweils zum Stundenbeginn ab. Es ersetzt die übliche Rekapitulation im Klassenverband. Die Teammitglieder beantworten reihum drei Fragen: „Welche Tasks habe ich seit dem letzten Treffen erledigt? Welche Tasks will ich als nächstes bearbeiten? Gab es Probleme und womit?“ Damit dabei nur die wesentlichen Informationen ausgetauscht werden wird das Meeting im Stehen und vor dem Project-Board abgehalten. Probleme werden nur benannt und ggf. wird Hilfe angefordert. Zu Hause erledigte Arbeiten werden nur mit Zustimmung des Teams integriert.

3.3 Pair-Programmierung – Strukturierte Paararbeit

Strukturierte Paararbeit hilft den Schüler/innen kollaboratives Arbeiten ohne aktiv-passiv-Teilung zu organisieren, sich gegenseitig zu unterstützen und Wissen und Information über konkrete Umsetzungen auszutauschen. Die Praktik sorgt dadurch für Transparenz, unmittelbares Feedback, planvolles und fokussiertes Vorgehen und einfache Lösungen. Im Informatikunterricht ist Pair-Programmierung eine Möglichkeit, strukturierte Paararbeit umzusetzen.

Die Partner/innen übernehmen definierten Rollen, die regelmäßig getauscht werden. Der Driver bearbeitet den Task und teilt dem Navigator seine Absichten und sein Vorgehen mit. Der Navigator überprüft die Bearbeitung, überlegt, ob es alternative, einfachere Lösungswege gibt, achtet darauf, dass der Driver bei der eigentlichen Aufgabe bleibt, und spricht mögliche Fehlinterpretationen an. In der Schule erhöht die Paararbeit die Selbstsicherheit, fordert ein Beschreiben des Vorgehens ein, initiiert Diskussionen über Lösungsansätze und Lösungsstrategien, verhindert Fehler und unterstützt so den Lernprozess. Es bietet sich an, dem Navigator konkrete, zur

Schülergruppe und dem Projektkontext passende Aufgaben an die Hand zu geben. Dennoch fällt den Schüler/innen der regelmäßige Rollenwechsel und die fachliche Kommunikation schwer und sie brauchen gelegentlich einen Hinweis der Lehrkraft.

4 Reflexion – Produkt- und Lernfortschritte durch Review und Retrospektive

An sich in Lehr- Lernprozessen eine Selbstverständlichkeit, bleibt die Reflexion aus Zeitgründen doch oft auf der Strecke, oder die Lernanstöße versanden, weil das nächste Projekt zu spät stattfindet. Hier ist das iterative Vorgehen ein klarer Pluspunkt – und das Ball-Point-Game, das fast alle Teams als Warm-up spielen verdeutlicht das besonders deutlich. Regelmäßig nach jeder Arbeitsphase innezuhalten und im sogenannten Review Feedback zum Produkt einzuholen, bringt das Team voran, muss aber auch gelernt werden. Indem dabei gesetzte Ziele mit dem Erreichten verglichen werden, fördert es die Selbstregulationsfähigkeiten. Zugleich ist es ein Anlass, das Erreichte zu feiern. Mögliche Probleme im Arbeitsprozess oder im Team können in einer Retrospektive offen angesprochen und angegangen werden. Dazu braucht es Mut, Respekt und Offenheit. Die wiederkehrende Gelegenheit, das zu üben, unterstützt den Ausbau und die Stärkung der Teamfähigkeit und verhindert, dass sich Frustration aufstaut.

5 Agile Projekte nicht nur im Informatikunterricht

Die Erfahrungen zeigen, dass Schüler/innen, die Erfahrungen mit agilen Projekten im Informatikunterricht gesammelt haben, ihre Kompetenzen zur Arbeits- und Selbstorganisation auch in anderen Fächern einsetzen und so die ganze Schule davon profitieren kann. Gleichfalls liegen inzwischen Erfahrungen mit agilen Unterrichtsprojekten im Physik- und Mathematikunterricht vor und selbst in Grundschulen werden Projekte agil organisiert.

Das iterative Vorgehen eignet sich beispielsweise auch hervorragend für naturwissenschaftlicher Forschungsprojekte und die Entwicklung von technischen Produkten. So wurden agile Methoden beispielsweise auch in einem Physik-Projektseminar mit dem Titel „Erfinderwerkstatt“, in dem Oberstufenschüler/innen ein Jahr Zeit hatten, Produkte aus einer gemeinsamen Idee zu entwickeln, mit Erfolg eingesetzt. Hierbei halfen die agilen Methoden den Schüler/innen den herausfordernd langen Zeitrahmen selbständig zu gestalten. Die Prototypen halfen, die generelle Machbarkeit zu prüfen, Kosten- und Aufwandsanalysen zu konkretisieren und schrittweise ein praxistaugliches Produkt zu entwickeln.

Im Regelunterricht können agile Methoden kooperatives Lernen unterstützen. So kann beispielsweise eine ausreichend komplexe Aufgabe aus der Analysis oder der analytischen Geometrie in der Oberstufe bearbeitet werden. Die Time-Boxen für die folgenden Aufgaben sind sehr kurz: im Team Teilprobleme

notieren, in strukturierter Paararbeit arbeitsteilig Lösungsstrategien entwickeln und sie im Team prüfen und ggf. ergänzen, in strukturierter Paararbeit arbeitsteilig relativ konkrete Lösungswege skizzieren und im Team besprechen. In einer anschließenden längeren Arbeitsphase am Platz erfolgt die konkrete Lösung der Teilprobleme in strukturierter Paararbeit und soweit möglich arbeitsteilig. Danach werden die Teillösungen im Team zur Gesamtlösung zusammengefügt. Abschließend besprechen die Teams wesentliche Aspekte und Erkenntnisse. Eine detailliertere Beschreibung der beiden Beispiele findet sich bei BRICHZIN KASTL & ROMEIKE (2019).

Leser/innen, die nicht aus der Informatik kommen, haben sicher inzwischen schon einige Ideen, agile Praktiken in ihren Unterricht zu integrieren und agile Projekte in ihrem Fach durchzuführen. Diesen Kolleg/inn/en sei noch ein Rat agil arbeitender Teams mit auf den Weg gegeben: „Haben Sie Mut, adaptieren Sie den Prozess und die Praktiken und sprinten Sie los. Nutzen Sie die regelmäßigen Reflexionspunkte, um aus den Erfahrungen zu lernen. Lassen Sie sich durch die ersten ein, zwei Iterationen nicht aus der Ruhe bringen – sie verlaufen meist chaotisch. Verändern Sie den Prozess in kleinen Schritten, um die Auswirkungen möglichst sicher der vorangegangenen Änderung zuschreiben zu können. Nutzen Sie die agilen Werte, wenn Sie sich unsicher sind. Stellen Sie sich immer wieder die Frage, ob die Auswahl der Praktiken das Team noch optimal unterstützt, ob am Project-Board jederzeit alle relevanten Informationen auf einen Blick erfassbar sind und ob die gewählte Iterationsdauer für das Team passt. Überdenken Sie als Team den Arbeitsablauf und haben Sie den Mut, Änderungen, die im Einklang mit den agilen Werten stehen, auszuprobieren.“ „Es ist“, resümierte eine Lehrkraft ihr erstes agiles Projekt, „erstaunlich, was die Schüler/innen leisten, wenn man sie nur machen lässt.“

Literatur

BRICHZIN, P. (2015). Agile Softwareentwicklung – Erfahrungsbericht eines Oberstufenprojekts im Wahlpflichtunterricht. In J. GALLENBACHER (Hg.), *Informatik Allgemeinbildend Begreifen* (S. 63–72). Bonn: Lecture Notes in Informatics (LNI) der Gesellschaft für Informatik.

BRICHZIN, P.; KASTL, P.; & ROMEIKE, R. (2019). *Agile Schule Methoden für den Projektunterricht in der Informatik und darüber hinaus*. Bern: hep Verlag.

BRICHZIN, P. & RAU, T. (2015). Repositories zur Unterstützung von kollaborativen Arbeiten in Softwareprojekten. In J. GALLENBACHER (Hg.), *Informatik Allgemeinbildend Begreifen* (S. 73–82). Bonn: Lecture Notes in Informatics (LNI) der Gesellschaft für Informatik.

DIETRICH, L.; GRAMM, A.; KASTL, P. & ROMEIKE, R. (2015). An image of the essence of software development: Experiences from two agile projects. In J. GALLENBACHER (Hg.), *Informatik Allgemeinbildend Begreifen* (S. 83–92). Bonn: Lecture Notes in Informatics (LNI) der Gesellschaft für Informatik.

FREY, K. (1983). Die sieben Komponenten der Projektmethode – mit Beispielen aus dem Schulfach Informatik. *Log In*, 3(2), 16–20.

HARTMANN, W.; NÄF, M. & REICHERT, R. (2007). *Informatikunterricht planen und durchführen*. Berlin, Heidelberg: Springer.

KASTL, P.; KIESMÜLLER, U.; & ROMEIKE, R. (2016). Starting out with projects – Experiences with agile software development in high schools. In *ACM International Conference Proceeding Series* (Vol. 13-15-Octo).

KASTL, P.; MÄRZ, S. & ROMEIKE, R. (2015). Agile Softwareentwicklung im Informatikunterricht – Ein Best-Practice-Beispiel am Spiel „Pengu“. In J. GALLENBACHER (Hg.), *Informatik Allgemeinbildend Begreifen* (S. 191–200). Bonn: Lecture Notes in Informatics (LNI) der Gesellschaft für Informatik.

KASTL, P. & ROMEIKE, R. (2018). Agile projects to foster cooperative learning in heterogeneous classes. In *IEEE Global Engineering Education Conference, EDUCON* (1182–1191).

KERBER, L.; KASTL, P.; & ROMEIKE, R. (2015). Agile Information Technology lessons as elementary instruction. In J. GALLENBACHER (Hg.), *Informatik Allgemeinbildend Begreifen* (S. 211–218). Bonn: Lecture Notes in Informatics (LNI) der Gesellschaft für Informatik.

KIESMÜLLER, U.; KASTL, P. & ROMEIKE, R. (2015). Project lessons all year round with agile framework. In J. GALLENBACHER (Hg.), *Informatik Allgemeinbildend Begreifen* (S. 219–228). Bonn: Lecture Notes in Informatics (LNI) der Gesellschaft für Informatik.

MEERBAUM-SALANT, O. & HAZZAN, O. (2010). An Agile Constructionist Mentoring Methodology for Software Projects in the High School. *ACM Transactions on Computing Education*, 9(4), 1–29.

ROMEIKE, R. & GÖTTEL, T. (2012). Agile Projects in High School Computing Education – Emphasizing a Learners' Perspective. In *Proceedings of the 7th WiPSCE'12* (pp. 48–57). ACM New York, NY, USA.

SCHUBERT, S. & SCHWILL, A. (2011). *Didaktik der Informatik*. Berlin, Heidelberg: Springer.

VersionOne (2020). *14th annual State of Agile Report*. <https://explore.versionone.com/state-of-agile> (06.08.2021)

WEIGEND, M. (2005). Extreme Programming im Klassenraum. In S. FRIEDRICH (Hg.), *INFOS 2005 -11. GI-Fachtagung Informatik und Schule*. Dresden, Germany. Bonn: Lecture Notes in Informatics (LNI) der Gesellschaft für Informatik.

PETRA KASTL, p.kastl@gmx.de, ist Lehrerin für die Fächer Informatik, Mathematik und Physik. Von 1995 bis 2005 entwickelte sie Software in Großprojekten in der IT. Seit 2013 begleitet sie das Projekt „Agile Methoden für den Informatikunterricht“ wissenschaftlich.

PETER BRICHZIN, brichzin@tcs.ifi.lmu.de, unterrichtete 20 Jahre als Lehrer für Informatik, Mathematik und Physik an der Medienschule Gymnasium Ottobrunn und ist jetzt Seminarlehrer für Informatik am Erasmus-Grasser-Gymnasium München. Er ist Autor der erfolgreichsten Informatik-Schulbuchreihe in Bayern und tätig in der Didaktik der Informatik an der LMU München.

RALF ROMEIKE, ralf.romeike@fu-berlin.de, ist Professor für Didaktik der Informatik an der Freien Universität Berlin. Ziel seiner Aktivitäten in Forschung und Lehre ist die Ausgestaltung informatischer Bildung, die Kinder und Jugendliche dazu befähigt, die digitale Gesellschaft zu verstehen und mitzugestalten. ■□