

Bahr, Tobias

Algorithmen in Softwareprojekten. Eine Unterrichtsidee für den Informatikunterricht im Profilfach IMP

Lehren & Lernen 49 (2023) 10, S. 31-35



Quellenangabe/ Reference:

Bahr, Tobias: Algorithmen in Softwareprojekten. Eine Unterrichtsidee für den Informatikunterricht im Profilfach IMP - In: Lehren & Lernen 49 (2023) 10, S. 31-35 - URN: urn:nbn:de:0111-pedocs-340969 - DOI: 10.25656/01:34096

<https://nbn-resolving.org/urn:nbn:de:0111-pedocs-340969>

<https://doi.org/10.25656/01:34096>

in Kooperation mit / in cooperation with:



<https://neckar-verlag.de>

Nutzungsbedingungen

Gewährt wird ein nicht exklusives, nicht übertragbares, persönliches und beschränktes Recht auf Nutzung dieses Dokuments. Dieses Dokument ist ausschließlich für den persönlichen, nicht-kommerziellen Gebrauch bestimmt. Die Nutzung stellt keine Übertragung des Eigentumsrechts an diesem Dokument dar und gilt vorbehaltlich der folgenden Einschränkungen: Auf sämtlichen Kopien dieses Dokuments müssen alle Urheberrechtshinweise und sonstigen Hinweise auf gesetzlichen Schutz beibehalten werden. Sie dürfen dieses Dokument nicht in irgendeiner Weise abändern, noch dürfen Sie dieses Dokument für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, aufführen, vertreiben oder anderweitig nutzen.

Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

Terms of use

We grant a non-exclusive, non-transferable, individual and limited right to using this document.

This document is solely intended for your personal, non-commercial use. Use of this document does not include any transfer of property rights and it is conditional to the following limitations: All of the copies of this documents must retain all copyright information and other information regarding legal protection. You are not allowed to alter this document in any way, to copy it for public or commercial purposes, to exhibit the document in public, to perform, distribute or otherwise use the document in public.

By using this particular document, you accept the above-stated conditions of use.

Kontakt / Contact:

peDOCS
DIPF | Leibniz-Institut für Bildungsforschung und Bildungsinformation
Informationszentrum (IZ) Bildung
E-Mail: pedocs@dipf.de
Internet: www.pedocs.de

Mitglied der


Leibniz-Gemeinschaft

Dieser Beitrag ist in der Ausgabe 10-2023 erschienen.





Hinweise zur Redaktion
und zu unserer
Zeitschrift finden
Sie auch unter
www.lehrenundlernen.eu

Lehren & Lernen

Zeitschrift für Schule und Innovation
aus Baden-Württemberg

Impressum

Herausgeber

Neckar-Verlag GmbH, Villingen-Schwenningen

Redaktionsleitung

Dr. Johannes Zylka (V. i. S. d. P.)

Prof. Dr. Helmut Frommer † bis 2014

OStD Johannes Baumann bis 2019

Redaktion

OStD Johannes Baumann, Wilhelmsdorf

Prof. Rolf Dürr, Reutlingen

Prof. Dr. Eva-Kristina Franz, Brühl

Dr. Joachim Friedrichsdorf, Grünstadt

Jun. Prof. Dr. Axinja Hachfeld, Konstanz

Carmen Huber, Salem

Prof. Dr. Kathrin Müller, Zürich

Prof. Dr. Volker Reinhardt, Freiburg

Dr. Helmut Wehr, Malsch

Dr. Johannes Zylka, Weingarten

Zuständig für das Thema dieses Heftes:

Dr. Johannes Zylka, Johannes Baumann

Manuskripte an den Verlag erbeten.

Über die Annahme entscheidet die Redaktion.

Rücksendung unverlangt eingesandter Manu-

skripte, Bücher und Arbeitsmittel erfolgt nicht.

Für unverlangt eingesandte Manuskripte wird

keine Haftung übernommen.

Verlag

Neckar-Verlag GmbH

Klosterring 1, 78050 Villingen-Schwenningen

Tel: +49 (0)7721/8987-0, Fax: +49 (0)7721/8987-50

info@neckar-verlag.de; www.neckar-verlag.de

Die Datenschutzbestimmungen der Neckar-Verlag GmbH
können Sie unter www.neckar-verlag.de einsehen.

Marketing/Anzeigenleitung:

Rita Riedmüller, Tel: +49 (0)7721/8987-44

E-Mail: werbung@neckar-verlag.de

Anzeigenverkauf:

Alexandra Beha, Tel: +49 (0)7721/8987-42

E-Mail: anzeigen@neckar-verlag.de

Es gilt die Anzeigenpreisliste Nr. 7 vom 1.1.2021

Bestellungen: beim Verlag

E-Mail: bestellungen@neckar-verlag.de

Lehren & Lernen erscheint monatlich und ist
in gedruckter sowie in digitaler Form erhältlich.

print oder digital: Einzelheft: 7,95 €,

Doppelheft: 15,90 €, Jahresabonnement: 50,- €

print & digital: Jahresabonnement: 54,80 €

Printausgaben jeweils zzgl. Versandkosten

Kündigungen nur schriftlich, spätestens 8 Wochen
vor Ende des aktuellen Bezugszeitraumes (nach
Ablauf der Mindestvertragslaufzeit).

Für Verbraucher/innen gilt: Nach Ablauf der
Mindestvertragslaufzeit verlängert sich das
Abonnement bis auf Widerruf und kann dann mit
Frist von 4 Wochen jederzeit gekündigt werden.

© 2023 Neckar-Verlag GmbH

Druck

jetoprint GmbH, 97080 Würzburg

INHALT

Unerledigte Aufgaben der Schulpädagogik – ein Blick zurück

Editorial 3

Doris Knab

Schulqualität: Unerledigte Aufgaben der Schulpädagogik 4

Renate Valtin

Grundschule – die Schule der Nation

Überlegungen zum Bildungsauftrag der Grundschule 10

Helmut Frommer †

Die Zeit ist reif

Die Neustrukturierung unserer Schullandschaft ist überfällig 17

Diethelm Wahl

Lernen heute 24

Zwischenruf

Manfred Schnitzler

Schöne neue KI-Welt: ein Zwischenruf 28

Panorama

Tobias Bahr

Algorithmen in Softwareprojekten.

Eine Unterrichtsidee für den Informatikunterricht im Profilfach IMP 31

Jürgen Belgrad, Carmen Huber

Leseförderung! Aber wie?

Tipps konkret: zur Beilage in LEHREN & LERNEN im Schuljahr 2023/24 36

Für Sie gelesen

Johannes Zylka zu:

Lemov: Unterricht im digitalen Klassenzimmer 37

Helmut Wehr zu:

Baumann/Götz: Schulleitung und Schulentwicklung für Fortgeschrittene 38

Themen der nächsten Hefte:

■ Sexuelle Bildung in der Schule

■ Grundsätzliches zur Gestaltung von Lehrerarbeitszeit an Schulen

■ Arbeitsfelder und Herausforderungen der Schulverwaltung



Einzelartikel downloaden unter: www.neckar-verlag.de

Dieser Ausgabe liegt ein Prospekt der Pädagogischen Hochschule Weingarten in
Zusammenarbeit mit der Neckar-Verlag GmbH bei. Wir bitten um freundliche Beachtung.

Algorithmen in Softwareprojekten. Eine Unterrichtsidee für den Informatikunterricht im Profilfach IMP

Immer mehr Gymnasien und andere Schularten bieten das Profilfach „Informatik – Mathematik – Physik“ (IMP) an. Hierfür, und für den Informatikunterricht insgesamt, liegen Unterrichtsideen und Materialien nicht im gleichen Maß vor wie in anderen Fächern. Der Beitrag stellt eine Unterrichtsidee zum Themenbereich „Algorithmen“ in Klasse 8 vor, die mit agilen Methoden konzipiert wurde. Am Beispiel der Unterrichtseinheit werden agile Methoden erklärt.

► Stichwörter: Profilfach „Informatik, Mathematik, Physik (IMP)“, Informatik, Agile Methoden, Tandemprogrammierung, Scrum

2018 startete das [Profilfach Informatik, Mathematik, Physik \(IMP\)](#) an allgemeinbildenden Gymnasien in Baden-Württemberg. Das neue vierstündige Fach IMP beginnt in Klassenstufe 8 und vereint Fachanteile aus Informatik, Mathematik und Physik. Mittlerweile ist das neue Profilfach an immer mehr Gymnasien etabliert und auch Gemeinschaftsschulen können sich seit dem Schuljahr 2019/20 dafür entscheiden. Einen Durchgang von Klassenstufe 8 bis 10 konnten die ersten Gemeinschaftsschulen mittlerweile abschließen, andere sind im ersten Durchgang oder wollen mit dem neuen Profilfach beginnen.

Für IMP liegt ausgearbeitetes [Unterrichtsmaterial](#) vor, bspw. auf dem Lehrerfortbildungsserver des Zentrums für Schulqualität und Lehrerbildung Baden-Württemberg (ZSL) (Eisenmann 2018). Insbesondere für den Unterricht an Gemeinschaftsschulen oder Lerngruppen mit großer Leistungsheterogenität scheint es jedoch erforderlich, das vorliegende Unterrichtsmaterial zu modifizieren und anzupassen. Dies ist bspw. in Bezug auf Differenzierungen oder auf methodische Überlegungen erforderlich, die auf ein individualisiertes und kooperatives Arbeiten zielen, wie dies im Schulgesetz für die Gemeinschaftsschulen festgelegt ist.

Der vorliegende Beitrag skizziert darauf bezogen eine Unterrichtsidee zum [Thema Algorithmen](#) des IMP-Bildungsplans des Gymnasiums der 8. Klassenstufe. Ziel der Unterrichtseinheit ist, dass die Schüler/innen ein erstes kleines Softwareprojekt in der visuellen Programmiersprache Scratch unter Anleitung durchführen (MKJS 2018). Dafür sind 13 Doppelstunden des Informatikteils in IMP vorgesehen (Eisenmann 2018). Die Einheit zielt sowohl auf inhaltsbezogene als auch prozessbezogene Kompetenzen des Bildungsplans: Einerseits sollen die bisher von den Schüler/innen erworbenen inhaltsbezogenen Kompetenzen (Variable, Schleife, Verzweigung, Zufallszahlen, indexbasierte

Datenstrukturen etc.) wiederholt werden, andererseits sollen sie kommunizieren und kooperieren, um damit die anvisierten prozessbezogenen Kompetenzen zu stärken (MKJS 2018). Möglichkeiten zur Differenzierung verschiedener Niveaustufen werden angedeutet.

Zur Durchführung der nachfolgend skizzierten Unterrichtsidee sollte ein Computerraum mit der aktuellen Version von Scratch, ein Austauschordner auf dem Laufwerk oder eine andere Infrastruktur zum Teilen von Informationen (z. B. Programmcode) zur Verfügung stehen. Weiterhin benötigen die Schüler/innen Accounts bei einer datenschutzrechtlich konformen Plattform zum Austausch von Informationen und/oder um den Projektfortschritt zu dokumentieren. Hier bieten sich bspw. die in der Basisversion kostenlosen Plattformen [factro®](#), [meistertask](#) oder [conceptboard](#) an ([factro®](#), 2023; [MeisterLabs GmbH](#), 2023; [Conceptboard Cloud GmbH](#), 2021). Die Einheit wird mit agilen Methoden geplant, die in der Softwareentwicklung ihren Ursprung haben. Sie werden nachfolgend erläutert.

Zur Zielsetzung der Unterrichtseinheit und zum Begriff agiler Methoden

Inhaltlich zielt die Einheit auf Algorithmen. Die inhalts- und prozessorientierten Kompetenzen sollen von den Schüler/innen am Beispiel eines [Minigolfspiels](#) erworben werden, das zur Verwirklichung der Kompetenzen sinnvoll erscheint und den Lernenden als Gegenstand bekannt ist. Für die Kreativität und die Motivation ist es förderlich, wenn die einzelnen Gruppen ihre individuellen Spiele in der Unterrichtseinheit selbst entwickeln und immer wieder die erfolgten Planungen mit dem Ist-Stand abgleichen. Ebenfalls versucht die Einheit, sich an das Vorgehen großer Softwarekonzerne anzulehnen, in denen agile Methoden erstmals entwickelt wurden.

Agile Methoden stammen aus der Softwareentwicklung. Sie wurden Anfang der 1990er Jahre für das effizientere und flexiblere Zusammenarbeiten in Softwareprojekten erstmals entwickelt und firmieren heute unter Begriffen wie z. B. Kanban, Lean, Scrum oder Design Thinking. Ihnen gemeinsam ist ein schrittweises Vorgehen in Lernschleifen, den sogenannten Iterationen (Frei/Koch 2023, 7). Sie gleichen sich darin, dass in ihnen jeweils der nächstfolgende Schritt geplant wird und dann eine Evaluation durch die Arbeitsgruppe folgt, welche die Grundlage des nächsten Schrittes darstellt. Damit trägt die Methode den sich verändernden Kontexten und Ungewissheiten Rechnung, die sich aus der längerfristigen Planung arbeitsteiliger Projekte ergeben. Mittlerweile wurde die Vorgehensweise auch auf Entwicklungen in anderen Feldern und auf die Schulpraxis übertragen (Brichzin u. a. 2019).

Agile Methoden im Informatikunterricht sind keineswegs neu (Brichzin u. a. 2019). Es existiert bereits ein Lehrbuch *Agile Schule* (Brichzin u. a. 2019), in dem die Methoden ausführlich beschrieben und durch Erfahrungsberichte ergänzt werden. Die nachfolgende Unterrichtseinheit will hier beispielhaft das Vorgehen mit agilen Methoden in einem konkreten Projekt beschreiben.

Zur Durchführung der Unterrichtseinheit

Die erste Doppelstunde beginnt mit einem motivierenden Einstieg, der zugleich das von den Schülergruppen anzustrebende Endprodukt vorstellt. Hierzu bietet sich die Präsentation eines fertigen Spiels an, aus welchem die qualitativen und quantitativen Anforderungen ersichtlich werden. Anschließend wird im gemeinsamen Gespräch ein **Anwendungsfalldiagramm** erarbeitet (Bahr u. a. 2022). Dieses visualisiert alle möglichen Funktionen des finalen Programms (ebd.). Es steht am Ende der ersten Stunde und wird mit den Schüler/innen gemeinsam erarbeitet (vgl. Abb. 1). Ein weiteres Ziel der ersten Einheit kann es sein, dass die Lernenden den Arbeitsalltag von Informatiker/innen kennenlernen. Um ihren Alltag zu teilen und um die Idee der agilen Methode zu verwirklichen, werden die aus der Sitzung resultierenden To-dos in Tasks (oder Aufgaben in Deutsch) umbenannt. Ein **Task** ist dadurch gekennzeichnet, dass er nicht mehr in kleinere Aufgaben geteilt werden kann. Dies kann beispielsweise sein, eine Minigolfbahn in Scratch zu zeichnen oder ihr einen Namen zu geben.

In der Softwareentwicklung werden der Einsatz und die Verwendungsmöglichkeiten von zu entwickelnden Programmen von einem imaginären Nutzer (User) her gedacht. Hier ergeben sich verschiedene Perspektiven auf das anvisierte Ziel, die von der Arbeitsgruppe (Team) in sogenannten User-Stories antizipiert werden müssen. Die **User-Stories** formulieren aus der Perspektive des gedachten Nutzers die potenziellen Verwendungsmöglichkeiten, die der User mit einem zu entwickelnden Minigolfspiel verbindet. Bspw. kann die User-Story sein: „Ich möchte eine Minigolfbahn sehen, um sie spielen zu können!“

Aus den antizipierten User-Stories und den daraus resultierenden Tasks erstellt das Team eine **To-do-Liste**, die die Grundlage der Arbeit in den nachfolgenden Einheiten bildet. Die Arbeit an diesen Aufgaben im Team wird nun in den nächsten Stunden über das sogenannte Projekt Board konzipiert (vgl. Abb. 2). Für den Einsatz des **Projekt Boards** sind verschiedene Möglichkeiten vorhanden (z. B. factro, conceptboard, analog). Das Projekt Board kann digital mit conceptboard, einer strukturgebenden Plattform wie factro oder analog auf einer Pinnwand mit Post-its entstehen. Welche Form des Projekt Boards für den Unterricht insgesamt oder von den Arbeitsgruppen gewählt wird, ist nicht entscheidend, aber es ist bedeutsam, dass durch die visuelle Anordnung der Aufgaben ihre Priorisierung ersichtlich wird und deutlich ersichtlich ist, welche Schritte aktuell bearbeitet werden und welche nachfolgend anstehen.

Die Arbeitsphasen, die mit der zweiten Doppelstunde beginnen, können beispielsweise mittels **Tandem-Programmierung** umgesetzt werden, bei der am Computer zwei Schüler/innen kooperativ, aber in unterschiedlichen Rollen, zusammenarbeiten und in bestimmten Zeitabständen die Rollen wechseln. Hier ist die Rolle des Drivers und des Navigators zu unterscheiden. Der

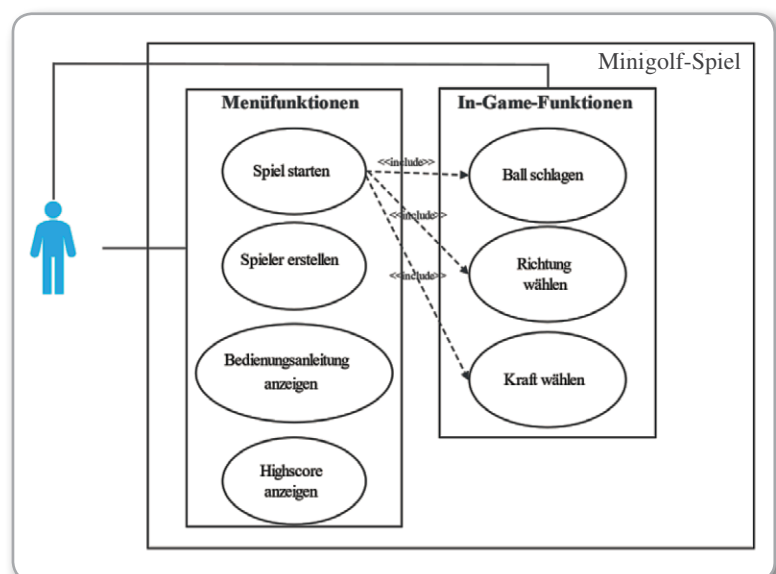


Abb. 1: Anwendungsfalldiagramm
(Original von Bahr u. a. 2022 lizenziert unter CC BY-SA 4.0)

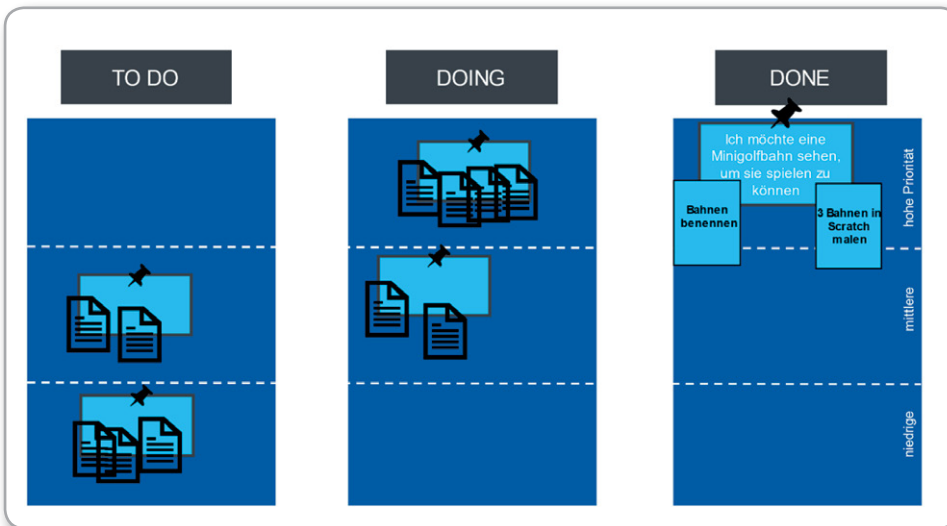


Abb. 2: Project Board (adaptiert, Original von Bahr u. a. 2022 lizenziert unter CC BY-SA 4.0)

Driver (d.h. der/die Fahrer/in) ist für die Implementierung des Programmcodes zuständig (bei diesem Projekt in Scratch). Der/die **Navigator/in** trägt als Tandempartner dagegen die Verantwortung dafür, dass an der aktuellen Aufgabe gearbeitet wird. Außerdem ist seine Aufgabe, die Verknüpfungen zwischen den Funktionen des Spiels im Blick zu behalten, und bei fortgeschrittenen Projekten auch, auf die Kommentierung des Codes hinzuweisen. Die Programmierung im Tandem mit verteilten Rollen bietet die Möglichkeit, den entstehenden Programmcode zu testen und zu hinterfragen.

Sobald den Tandempartnern die unterschiedlichen Rollen geläufig sind, kann jedes Tandem mit der Programmierung beginnen. Es empfiehlt sich, nach 20 Minuten einen Rollen- und Sitzplatzwechsel vorzunehmen, damit alle Schüler/innen am Ende ungefähr dieselbe Programmierzeit bekommen. Zu Beginn der Unterrichtseinheit übernimmt die Lehrperson die Funktion des Zeitwächters, nach einer gewissen Einarbeitungszeit soll der Wechsel von den Tandems selbstständig vorgenommen werden (z. B. durch einen Wecker).

Zu Beginn der zweiten Doppelstunde zeichnen die Gruppen neben die Spalte für die To-do-Liste eine weitere Spalte: Doing (bedeutet: wird gerade bearbeitet). Bearbeiten Gruppenmitglieder eine User-Story, dann verschieben sie diese in die zweite Spalte. Somit verfügen alle Gruppenmitglieder und die Lehrperson über eine aktuelle Übersicht, an welchen To-dos gerade gearbeitet wird und welche weiteren Aufgaben geplant sind. Sind die Gruppen mit einer User-Story fertig, dann verschieben sie diese in die dritte Spalte: Done (fertig). Somit ergibt sich ein Steckbrett, welches auf einen Blick den Stand des Projekts zeigt, das sogenannte Project Board (vgl. Abb. 2).

Die ersten User-Stories dienen dazu, dass die Teams zügig visuell vorzeigbare Ergebnisse erstellen kön-

nen. Hierzu gehören die Grundfunktionen des Minigolfspiels, bspw. dass der Minigolfball rollt, von der Wand abprallt usw. Agil wird dieses Vorgehen dadurch, dass User-Stories keine feste und unveränderliche Zielgröße darstellen, sondern nach jedem Task modifiziert und angepasst werden können. Möglicherweise stellt sich im Arbeitsprozess heraus, dass eine User-Story viel mehr Arbeit als ursprünglich gedacht umfasst und deshalb zu

umfangreich geraten ist. Das agile Setting bietet hier die Möglichkeit, dass sich das Team kurz und unmittelbar vor dem Project Board trifft und die User-Story in mehrere Teilbereiche aufteilt. Sollten von der Arbeitsgruppe keine spontanen Meetings eingefordert werden, sind trotzdem feste Treffen vorgesehen, um sicherzustellen, dass über auftretende Probleme gesprochen werden kann. Diese festinstallierten Meetings firmieren unter dem Namen **Reflexion**. Zu ihnen trifft sich das Team vor dem Project Board und jedes Mitglied beantwortet die drei folgenden Fragen (Brichzin u. a. 2019, 143f.):

1. Was lief gut?
2. Was hätte besser laufen können?
3. Was wollen wir im nächsten Schritt konkret ändern?

Dadurch werden die Teammitglieder aufgefordert, konstruktiv kritisch über ihre Zusammenarbeit nachzudenken und den weiteren Prozess zu optimieren. Analog dazu kann zu Beginn eines Zyklus, bspw. jede Woche, das Stand-up-Meeting durchgeführt werden. Hier werden vor dem Project Board die nachfolgenden drei Fragen geklärt (adaptiert nach Brichzin u. a. 2019, 118f.):

1. Welche Tasks wurden seit dem letzten Treffen erledigt?
2. Welche Tasks werden im nächsten Schritt bearbeitet?
3. Falls Probleme aufgetreten sind, welche waren dies?

Meetings (Stand-up-Meeting, Review) beziehen sich hauptsächlich auf den Arbeitsprozess. In ihnen stehen die Prozesssteuerung und die Qualität der Arbeit im Vordergrund. Die feststehenden und spontanen Treffen der Arbeitsgruppen kennzeichnen das agile Vorgehen, das damit jederzeit die Möglichkeiten zur flexiblen Veränderung der Arbeitsprozesse beinhaltet. Inhaltliche Fragen können direkt bilateral oder kurz in der Gruppe

geklärt werden, bspw. „Welche Farbe verwenden wir für die Ränder einer Minigolfbahn? Oder benutzen wir verschiedene Farben für die Ränder der Minigolfbahnen?“

Nachdem die ersten User-Stories bearbeitet sind und getestet wurden, entsteht ein erster Prototyp des Programms. Dieses Vorgehen wird in den nächsten Doppelstunden der folgenden Wochen wiederholt (sogenannte Iterationen). Als Ergänzung zum Abschluss eines jeden Zyklus, d.h. nach dem Testen des Programmes, erfolgt eine gemeinsame Reflexion.

Die Ergebnissicherung findet in der Regel durch die Präsentation der iterativ erstellten Prototypen statt. Jedes Gruppenmitglied soll dabei Teile des Codes erklären. Darüber hinaus können während der Projektarbeit mündliche Noten zum Arbeitsprozess eingeholt werden. Die Rolle der Lehrperson ist die des Coaches, welche Schüler/innen berät, sie motiviert und positiv auf Fehler reagiert. Eine Darstellung des Stundenablaufs mit Stand-up-Meetings, Arbeitsphasen und Reflexion gibt Abbildung 3.

Möglichkeiten zur Differenzierung der Unterrichtseinheit

Differenzierungen der Unterrichtseinheit sind sowohl in Bezug auf die inhaltlichen Anforderungen an das Minigolfspiel als auch im Hinblick auf das soziale Arrangement möglich. Bei 13 vorgesehenen Unterrichtsstunden ist ausreichend Zeit vorhanden, dass einzelne Gruppen ein grundlegendes Niveau (G-Niveau) erreichen. Beispielsweise bietet sich hierzu an, dass nur drei Minigolfbahnen gespielt werden können. Für das M- und E-Niveau kann darüber hinaus ein Mehrspielermodus programmiert werden. Ebenfalls könnten Sounds, Wind und weitere Extras von zügig arbeitenden Gruppen hier additiv hinzugefügt werden. Weiter ist denkbar, dass die Schüler/innen mit viel Programmiererfahrung länger in der Rolle des Navigators verbleiben, um leistungsschwächere Schüler/innen zu unterstützen (Brichzin u.a. 2019). Optional können auch Arbeitsblätter zum Minigolfspiel eingesetzt werden, die strukturgebende und kreative Aufgaben enthalten, die als Hilfestellung für schwächere Schüler/innen dienen. Zusätzlich können die Aufgabenblätter Fragen zur Überprüfung für die Schüler/innen beinhalten, damit die Lernenden im Team ihre bis dahin erstellten Programme selbsttätig auf mögliche Fehler hin überprüfen.

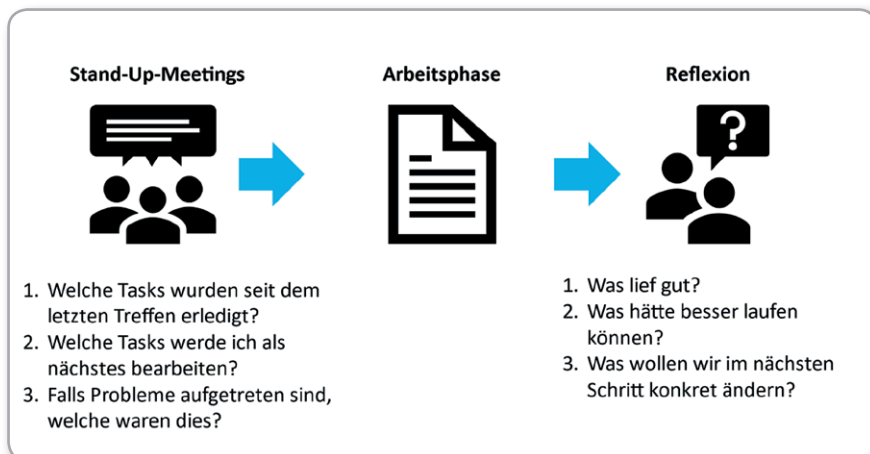


Abb. 3: Darstellung eines Stundenablaufs

Merkkasten zu den Begriffen der Unterrichtseinheit:	
Anwendungsfall-diagramm	Das Anwendungsfalldiagramm beschreibt aus Nutzer/innensicht, welche Anwendungen das Minigolfspiel enthalten sollte (z. B. Spiel starten, Richtung des Schlags auswählen, Minigolfball schlagen, ...)
Teams	Gruppen aus 3 bis 5 Schüler/innen
Project Board	Sortiertes Steckbrett mit den Spalten To-do, in Bearbeitung und abgeschlossen (digital oder analog)
User-Story	Benutzer/innengeschichte, die beschreibt, was der/die Spieler/in tun kann. (Z. B. der/die Spieler/in kann die Richtung auswählen, um den Ball in eine Richtung zu schlagen.)
Tasks	Aufgaben, die nicht mehr in weitere Unteraufgaben geteilt werden können
Stand-up Meeting	Ein 5- bis maximal 15-minütiges Treffen vor dem Project Board, indem jedes Teammitglied die 3 Fragen beantwortet: Welche Tasks wurden seit dem letzten Treffen erledigt? Welche Tasks werde ich als Nächstes bearbeiten? Falls Probleme aufgetreten sind, welche waren dies?
Reflexion	Ein 5- bis maximal 15-minütiges Treffen vor dem Project Board, indem jedes Teammitglied die 3 Fragen beantwortet: Was lief gut? Was hätte besser laufen können? Was wollen wir im nächsten Schritt konkret ändern?

Tandem-Programmierung	2 Schüler/innen sitzen vor dem Computer. Eine/r programmiert (Driver), der andere sitzt daneben und überprüft/kommentiert das Arbeiten (Navigator). Nach 20 Minuten werden die Rollen gewechselt.
Prototyp	Teilprodukt aus dem Arbeitsprozess mit einzelnen Funktionalitäten. Zum Beispiel ein Minigolfspiel mit einer Bahn.
Zyklus	Arbeitsprozess bei agilen Methoden mit den Schritten: (1) Tasks beschreiben, (2) Stand-up-Meeting, (3) Tandem-Programmieren, (4) Programm testen und (5) Reflexion

[tps://lehrerfortbildung-bw.de/u_matnatech/imp/gym/bp2016/fb1/2_i2_alg/1_hintergrund/2_verlauf/01_alg_unterricht_hintergrund.pdf](https://lehrerfortbildung-bw.de/u_matnatech/imp/gym/bp2016/fb1/2_i2_alg/1_hintergrund/2_verlauf/01_alg_unterricht_hintergrund.pdf) (aufgerufen am 24.03.2023).

Factro: Homepage 2022. Online verfügbar unter <https://www.factro.de/impressum/> (aufgerufen am 24.03.2023).

Frei, C./Koch, F.: Agiles Arbeiten macht Schule. In: Pädagogik 75 (2023), H. 4, 6-10.

MeisterLabs GmbH: Task Management für Teams – MeisterTask 2023. Online unter <https://www.meistertask.com/de/imprint> (aufgerufen am 24.03.2023).

Ministerium für Kultus, Jugend und Sport Baden-Württemberg: Bildungsplan zum Profulfach Informatik, Mathematik, Physik (IMP). Villingen-Schwenningen 2018. Online unter http://bildungsplaene-bw.de/site/bildungsplan/get/documents/lsw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_GYM_IMP.pdf (aufgerufen am 24.03.2023).

Romeike, R.: Kreativität im Informatikunterricht. Dissertationsschrift. Potsdam 2008. Online verfügbar unter <http://ddi.cs.uni-potsdam.de/Forschung/Schriften/RomeikeDiss2008.pdf> (aufgerufen am 12.05.2023).

Fazit

Informatikunterricht hat das Potenzial, die Kreativität von Lernenden zu fördern. Dies kann beispielsweise durch Kreativitätstechniken, offene Aufgabenstellungen und die Möglichkeit, dass Schüler/innen sich selbst in ihrer Lösung wiederfinden, ermöglicht werden (Romeike 2008, 80f.). In der Praxis ist es oft schwierig, diese Anforderungen an kreativen und selbstorganisierten Informatikunterricht zu erfüllen. Der Einsatz agiler Methoden bietet für den Projektunterricht einen Rahmen, der den Fokus auf die genannten Herausforderungen setzt. Im Beispiel der skizzierten Unterrichtseinheit wird der agile Prozess über den Zyklus aus (1) Tasks beschreiben, (2) Stand-up-Meeting, (3) Tandem-Programmieren, (4) Testen und (5) Reflexion (auch Review genannt) realisiert (Brichzin u. a. 2019, 93f.).

Weitere Informationen

Für weitere Informationen zum Unterrichtsmaterial (Arbeitsblätter, Folien, Scratch Projekt) oder Weiterbildungen zu agilen Methoden wenden Sie sich bitte an: bahr@ife.uni-stuttgart.de.

Literatur

Bahr, T./Zinn, B.: Workshop zum Einsatz agiler Methoden im Informatikunterricht. In: Demmler, D./Krupka, D./Federrath, H. (Hrsg.): INFORMATIK 2022, Gesellschaft für Informatik. Bonn 2022, 1067-1077.

Brichzin, P./Kastl, P./Romeike, R.: Agile Schule – Methoden für den Projektunterricht in Informatik und darüber hinaus. Bern 2019.

Conceptboard Cloud Service GmbH: Homepage 2023. Online unter <https://conceptboard.com/de/impressum/> (aufgerufen am 24.03.2023).

Eisenmann, M.: Vertiefung Algorithmen – Unterrichtsgang und Hintergrundinformationen. 2018. Online verfügbar unter <https://www.eisenmann.de/>

Tobias Bahr

Akademischer Mitarbeiter am Institut für
Erziehungswissenschaft der Universität Stuttgart
bahr@ife.uni-stuttgart.de