

Der Begriff ‚Kreativität‘ wird in der Informatikdidaktik häufig und in unterschiedlicher Weise verwendet. In diesem Artikel werden zunächst Berührungspunkte von Informatikdidaktik und Kreativität identifiziert. Die vier Perspektiven der Kreativitätsforschung nach Rhodes (1961) – Person, Produkt, Prozess und Press (Umfeld) – dienen als Grundlage für eine Einordnung. Die anschließende Diskussion deckt Desiderate auf und gibt Impulse, wie Kreativität in der Informatikdidaktik weitergedacht werden kann.

The term ‘creativity’ is used frequently and in different ways in computer science education. In this article, we first identify points of contact between computer science education and creativity. The four perspectives of creativity research according to Rhodes (1961) – person, product, process, and press – serve as a basis for classification. The following discussion reveals desiderata and gives impulses how creativity can be further thought in computer science education.

KREATIVITÄT IN DER INFORMATISCHEN BILDUNG

BESTANDSAUFNAHME UND ENTWICKLUNGSPERSPEKTIVEN

Thomas Schmalfeldt und Björn Maurer

1 EINLEITUNG

Seit Ende der 1960er Jahre ist Kreativität im deutschsprachigen Raum zu einem zentralen Bildungsziel avanciert (vgl. Veit 2017, S. 13). Die OECD deklariert Kreativität im Jahr 2000 als Kernelement der Wissensgesellschaft und fordert explizit die Förderung kreativer Fähigkeiten (vgl.

Haager 2019, S. 197). Die Agenda EU2020 empfiehlt im Jahr 2010 die Aufnahme kreativer und unternehmerischer Kompetenzen in die schulischen Curricula. Nebst Kollaboration, Kommunikation und kritischem Denken zählt Kreativität zu den 4Ks, (engl. 4Cs), die unter anderem im Framework der 21st Century Skills der wirtschaftsnahen *Partnership for 21st Century Skills* unter

der Bezeichnung „Learning and Innovationskills“ zusammengefasst werden (vgl. Kay 2010). Schulen stehen zunehmend vor der Aufgabe, kreativitätsfördernde Lerngelegenheiten und Rahmenbedingungen zu schaffen und die Schüler*innen zu kreativem Handeln zu ermutigen. In der Schweizer Volksschule (Kindergarten bis Klasse 9) ist im Schuljahr 2018/19 in der Mehrzahl der Kantone das Fach *Medien und Informatik* eingeführt worden, das medienpädagogische Anliegen mit informatischer Bildung verbinden soll. Bestandteil der Entwicklung der neuen Fachbereichsdidaktik ist der Versuch, bildungsrelevante Schnittfelder zwischen der interdisziplinär ausgerichteten Medienpädagogik und der Ingenieurwissenschaft Informatik auszuloten und didaktisch nutzbar zu machen. Kreativität könnte ein solches Schnittfeld sein. In medienpädagogischen Praxisfeldern haben Kreativität und kreativer Selbstaussdruck – insbesondere in der aktiven Medienarbeit – traditionell eine große Bedeutung (vgl. z. B. Niesyto 2006; Schmoelz et al. 2017). In der Informatikdidaktik wird der Begriff ebenfalls diskutiert, allerdings unter anderen Vorzeichen. In diesem Beitrag wird diesbezüglich eine Bestandsaufnahme vorgenommen: Von welchem Kreativitätsverständnis wird im informatikdidaktischen Diskurs ausgegangen? Welche kreativen Spielräume bieten schulische Informatiklernanlässe? Wo und wie zeigt sich informatische Kreativität? Welche Anknüpfungspunkte ergeben sich zu medienpädagogischen Perspektiven? Der Literaturkorpus, der dieser Bestandsaufnahme zugrunde liegt, setzt sich zum einen aus Standardwerken der Informatikdidaktik im deutschsprachigen Raum zusammen. Zum anderen werden Texte von Autor*innen einbezogen, die einschlägig im Bereich

Kreativität als Gegenstand oder Perspektive des Informatikunterrichts publiziert haben. Ergänzend werden relevante Publikationen aus der Digital Library der *Association for Computing Machinery* (ACM) herangezogen. Grundlagentexte zur Kreativitätsforschung und Kreativitätstheorie runden den Textkorpus ab.

In Handbüchern der deutschsprachigen Informatikdidaktik wird das Thema Kreativität aufgegriffen, spielt insgesamt aber eine eher untergeordnete Rolle (vgl. Hartmann et al. 2007; Hubwieser 2007; Humbert 2006). Modrow/Strecker (2016) erwähnen die Relevanz von kreativem Informatikunterricht, nehmen aber keine Ausdifferenzierung des Begriffs Kreativität vor. Schubert/Schwill (2011) widmen Kreativität ein eigenes Kapitel – beigesteuert von Romeike, der den Diskurs um Kreativität im deutschsprachigen Raum angestoßen hat (vgl. Romeike 2007, 2008, 2011). Mit den Entwicklungen in den Bereichen Robotik, Physical Computing und Making erfährt dieser Diskurs in jüngerer Zeit eine Renaissance (vgl. Schelhowe 2018, S. 3; Dittert et al. 2016 und 2019).

2 INFORMATIK – EINE KREATIVE DISZIPLIN?

Während Informatik in der Öffentlichkeit als eher schematisch, un kreativ und anwendungsfern wahrgenommen wird (vgl. Romeike 2008a, S. 356; Knobelsdorf/Romeike 2008, S. 290; Dittert et al. 2016), betonen einige Vertreter*innen der Informatikdidaktik das kreative Potenzial (vgl. Mishra/Yadav 2013, S. 11; Romeike 2008a, S. 130; Romeike 2011, S. 355). So zeichnet die Informatik für bahnbrechende Innovationen verantwortlich.

Die Computermaus (erster Prototyp 1963) beispielsweise wurde zu einer Zeit erfunden, als es keine grafischen Benutzeroberflächen gab, was wiederum zur Entwicklung von grafischen Benutzeroberflächen beigetragen hat. Der Algorithmus von *Google* (veröffentlicht 1997) hat die Suche im Internet revolutioniert und seinerzeit etablierte Suchmaschinen wie *Yahoo* oder *Alta-Vista Lycos* vom Markt verdrängt. Im Jahr 2011 wurden einer breiten Öffentlichkeit erstmals die Möglichkeiten der künstlichen Intelligenz bewusst gemacht. In der Quizshow *Jeopardy!* müssen die Teilnehmenden nicht die Antwort auf eine Frage geben, sondern zu einer Antwort eine Frage formulieren. Das von *IBM* entwickelte KI-Programm *Watson* spielte gegen reale, zuvor sehr erfolgreiche Teilnehmer*innen und gewann. *Watson* folgt nicht nur einem Regelwerk, sondern ‚versteht‘ die Frage, sucht die geeignete Antwort und formuliert daraus eine korrekte Frage. Innovationen dieser Tragweite bezeichnen Kaufman/Beghetto (2009) als „Big-C-Creativity“. Sie werden von Expert*innen hervorgebracht, die sich jahrelang in einer Domäne engagiert und Wissen und Erfahrung aufgebaut haben. Sie sind zum jeweiligen Zeitpunkt neuartig, bzw. originell, entsprechen aber auch den Bedürfnissen der Nutzer*innen, lösen Probleme adäquat, gelten je nach Kontext als effektiv, zuverlässig oder praxistauglich. Damit erfüllen sie die klassischen Kriterien für kreative Produkte, wie sie unter anderem von Stein (1953) und Jackson und Messick (1965) formuliert wurden (vgl. Barnett/Romeike 2017, S. 302). „Originality is not alone sufficient for creativity. Original things must be effective to be creative“ (Runco 2004, S. 92).

3 DAS KREATIVITÄTS- VERSTÄNDNIS IN DER INFORMATIKDIDAKTIK

Von den ersten Gehversuchen im Programmieren in der Schule bis hin zu informatischen Innovationen auf *Big-C-Level* ist es ein weiter Weg, den zu verfolgen nicht das Ziel schulischer Bildung sein kann. Von welchem Kreativitätsverständnis und -anspruch wird in der Informatikdidaktik ausgegangen? „We call a phenomenon ‚creative‘ (...) when it leads to original, adaptive, and useful ideas, solutions, or insights“ (Knobelsdorf/Romeike 2008, S. 287). Es herrscht Konsens, dass Kreativität im Informatikunterricht nicht an der ‚Grossen Kreativität‘ bemessen, sondern subjektorientiert konzeptualisiert werden muss. So stellen Dagiené et al. (2019, S. 384) fest: „Creativity for learners is not only the process of creating something novel. From the viewpoint of learners, it is already creativity when they develop or discover something that is new to them although someone else already knew the idea“. Ähnlich positioniert sich Romeike (2007). Er gehört zu den Autor*innen im deutschsprachigen Raum, die sich explizit und längerfristig mit der Bedeutung von Kreativität im Informatikunterricht auseinandergesetzt haben.

Romeike strebt für den Informatikunterricht die Förderung von „P-Kreativität“ (Personenkreativität) im Gegensatz zur ‚großen‘ historischen „H-Kreativität“ an. *P-Kreativität* entspricht dem, was Kaufman/Beghetto (2009) als *Mini-C* bzw. *Little-C Creativity* bezeichnen. *Little-C Creativity* beschreibt kreativen Ausdruck im Alltag. Damit sind Problemlösungen gemeint, die auch von Personen ohne Fachexpertise entwi-

ckelt werden können (vgl. Kaufman/Beghetto 2009, S. 95). Als *Mini-C-Creativity* bezeichnen sie subjektiv neue und bedeutsame Interpretationen von Erfahrungen, Aktivitäten und Ereignissen im Alltag (vgl. ebd., S. 98). Dieses Kreativitätsverständnis betont die Parallelen mit dem Lernbegriff sowie die Verbindung von Kreativität mit konstruktivistischen Lernmodellen und Deweys Ansatz ‚Learning by Doing‘ (vgl. Dagienė et al. 2019, S. 384). Raum für Kreativität im Informatikunterricht korrespondiert mit einem emanzipativen Bildungsanliegen. In der Auseinandersetzung mit Digitalität, Informatiksystemen und Algorithmen wechseln die Schüler*innen von der Konsumierenden- in die Produzierendenperspektive (vgl. Mishra/Yadav 2013, S. 11). Sie sind dabei nicht nur Anwender*innen von digitalen Angeboten, sondern sie gestalten diese Angebote selbst mit. Ferner werden kreative Ansätze als motivationale Türöffner genutzt, um Schüler*innen für informatische Themen zu interessieren (vgl. Dittert et al. 2016, S. 16). Diese Zugänge tragen zu Bereitschaft, Konzentration und Ausdauer der Schüler*innen bei (vgl. Romeike 2008a, S. 356; Romeike 2011, S. 304). Kreative Freiräume sensibilisieren für den Umstand, dass Informatik neben der technischen noch weitere spannende Facetten beinhaltet (vgl. Katterfeldt et al. 2019). So weisen Dagienė et al. daraufhin: „Our ambition is to go beyond technology and open more space for creativity“ (2019, S. 382). Romeike (2008a, S. 355) sieht den Beitrag der Informatik allerdings nicht primär in einer weiteren kreativitätsfördernden Maßnahme.

4 VIER PERSPEKTIVEN AUF KREATIVITÄT IM FACH INFORMATIK

Kreativität gilt als multidimensionales Konstrukt, dessen Elemente in wechselseitiger Abhängigkeit stehen. Rhodes (1961) hat in seinem *4Ps of Creativity* Modell die Perspektiven Prozess, Person, Produkt und Umfeld (engl. Press) differenziert. In Anlehnung an Jourdanous (2016, S. 210) dient das Modell von Rhodes (1961) als konzeptionelle Leitplanke, um herauszuarbeiten, inwieweit die vier Perspektiven im informatikdidaktischen Diskurs zu Kreativität Berücksichtigung finden.

4.1 KREATIVE PERSÖNLICHKEITSEIGENSCHAFTEN

Romeike (2011, S. 355) geht davon aus, dass alle Schüler*innen in der Lage sind, kreativ zu sein. „Aufgabe des Informatikunterrichts muss es sein, dieses Potenzial herauszufordern, zu fördern und für die Unterrichtsziele (...) zu nutzen.“ Dass Motivation und Kreativität in einem engen Wechselverhältnis zueinander stehen, ist auch die Erfahrung in der Informatikdidaktik. Zur Frage, wieviel informatische Expertise nötig ist, um tatsächlich informatisch kreativ werden zu können, finden sich in der Literatur dagegen nur wenige Hinweise. Für Romeike (2007, S. 62) ist ein solides Grundwissen im jeweiligen informatischen Tätigkeitsfeld eine Voraussetzung. Informatische Modellierungs- und Problemlösungsentscheidungen, so Romeike, basieren auf diesem Grundwissen.

4.2 KREATIVE PROZESSE IM INFORMATIKUNTERRICHT

Romeike (2011, S. 355) geht davon aus, dass dem Unterrichtsfach Informatik per se kreative Prozesse immanent sind. Diese schlagen sich insbesondere in Modellierungs- und Programmieraktivitäten nieder (vgl. auch Modrow/Strecker 2016). Hierfür werden kreative Fähigkeiten wie Problemlösen, Reflexion und Computational Thinking benötigt (vgl. Barnett/Romeike 2017, S. 311). Brennan et al. (2020, S. 3) beschreiben Programmieren als kreative und kreativitätsfördernde Handlung. „We think about the creative power of programming, how programming can be used in many different ways to express ideas and solve problems“. Kreativität beim Programmieren bedeutet, so Romeike (2011, S. 367), „mit Ideen zu experimentieren, Heuristiken anzuwenden und verschiedene Lösungsmöglichkeiten zu testen. (...) Experimentieren schließt hierbei nicht das ‚Nach-Experimentieren‘ gemäß vorgegebener Versuchsanleitungen mit ein ...“. Stattdessen sollen eigene Ideen und Hypothesen überprüft werden (vgl. ebd.). Yadav/Cooper (2017, S. 32) sehen Parallelen zwischen kreativen Denkkoperationen bzw. -fertigkeiten (set of thinking tools) und den Grundbestandteilen des informatischen Denkens. Demnach seien Beobachtung, Imagination, Abstraktion, Dekomposition und Mustererkennung für Kreativität und Problemlösen zentral. Gleichzeitig handele es sich dabei um Teilaspekte von Computational Thinking (vgl. Wing 2006).

Im Informatikunterricht werden drei, sich teils überschneidende kreative Prozessmodelle eingesetzt. Sie differenzieren unterschiedliche Prozessphasen und ordnen diesen Tätigkeiten zu, die für die Produktentwicklung relevant sind. „Creativity

is seen as essential in designing, constructing, testing, and implementing, as well as in the phases of maintaining the product and in project management. (Barnett/Romeike 2017, S. 305).

Die (1) *Creativity Spiral* (Resnick 2008) beschreibt einen möglichen individuellen Programmierprozess. Sie beginnt im Idealfall mit einer eigenen Idee des Kindes (imagine), die es im nächsten Schritt umsetzen möchte (create). Während ein Kind ein Produkt entwickelt, spielt es mit dem verfügbaren Material (play) und entdeckt dabei selbst Verbesserungsmöglichkeiten. Beim Vorstellen und Teilen (share) bekommt es Anregungen, die dazu beitragen, das Produkt und den Entwicklungsprozess zu reflektieren (reflect). Dies kann zu neuen Ideen führen, die wiederum einen neuen Zyklus einleiten.

Das aus der Design-Entwicklung stammende (2) *Design Thinking* Modell (vgl. Kerguenne 2017) ist in zwei Hauptphasen eingeteilt – den im Folgenden beschriebenen Problemraum und den Lösungsraum. Beim *Design Thinking* ist *Empathie* ein essenzieller Bestandteil des Gestaltungsprozesses (vgl. Meinel et al. 2011). Ausgangspunkt ist das Entwickeln einer passenden Lösung für ein klar umrissenes Problem. Zu Beginn wird viel Zeit aufgewendet, um das Problem zu verstehen. Dabei erleichtern sogenannte ‚Persona‘ – fiktive idealtypische Vertreter*innen einer Zielgruppe –, sich in die Rolle der Betroffenen zu versetzen (Problemraum). Anschließend werden – in der Phase des sogenannten ‚Lösungsraums‘ – mögliche Prototypen bzw. Lösungen in mehreren Iterationen erarbeitet. *Design Thinking* fördert die intensive Auseinandersetzung der Schüler*innen mit Problemstellungen, für die mit informatischen Mitteln Lösungen entwickelt

werden können (vgl. O'Callaghan/Connolly 2020). Das Verfahren erhöht die Chance auf ein originelles und adäquates Produkt, da bereits vorhandene Lösungsansätze für den konkreten Fall häufig entweder zu unspezifisch oder unpassend sind.

Auch im (3) *Challenge Cycle* Modell (vgl. Romeike 2008b, S. 129) lösen die Schüler*innen ihre Probleme (beziehungsweise ‚Challenges‘) in zyklisch aufeinander folgenden Phasen. Idealerweise definieren sie ihre Challenge selbst (Challengephase), wobei die Lehrperson durch didaktisch flankierende Maßnahmen (zum Beispiel Vorgabe eines Rahmenthemas, Bereitstellung fachlicher Grundlagen, geeignete Lernumgebung) die Voraussetzungen für die erfolgreiche Bewältigung der Challenge schafft. In der darauffolgenden Problemmanagementphase werden mithilfe von Kreativitätstechniken Ideen und Problemlösestrategien entwickelt, die in der Implementierungsphase auf ihre Funktion getestet und bei Bedarf optimiert werden. Das Produkt wird abschließend den Peers vorgestellt und diskutiert.

4.3 KREATIVE PRODUKTE

Produkte sind in der Kreativitätsforschung ein wichtiger Analysegegenstand (vgl. Forthmann 2019, S. 80; Schubert/Loderer 2019, S. 54). Was aber charakterisiert kreative Produkte aus informatikdidaktischer Perspektive? Auf der Basis eines subjektorientierten Verständnisses von Kreativität geht die divergente Komponente ‚Originalität‘ nach Romeike (2007, S. 62) im subjektiven Originalitätsanspruch der Schüler*innen auf, die ihrem Produkt eine persönliche Note verleihen. Der konvergente Gegenpol – ‚Adäquatheit‘ – manifestiert sich dagegen in der *Lauffähigkeit* des

Produkts/Programms im jeweils vorgesehenen Anwendungskontext (vgl. Romeike 2007, S. 58).

Lernprodukte im Informatikunterricht sind meist Softwareprodukte. Anders als im Hardwarebereich können sich Schüler*innen, so Romeike (2008a, S. 132), im Softwarebereich eigenständig informatisch kreativ ausdrücken. Zudem vollziehen sich ein Großteil der informatischen Innovationen ohnehin softwareseitig. Softwareprodukte haben die Eigenschaft, nicht abgeschlossen und somit offen zu sein für Weiterentwicklungen und Optimierungen (vgl. Romeike 2008a, S. 133). Design-Entscheidungen beim Programmieren können getestet, bei Bedarf neu ausgerichtet oder rückgängig gemacht werden. Diese Eigenschaft von Softwareprodukten kommt iterativen Entwicklungsverfahren (wie Design Thinking bzw. Challenge Cycle) entgegen. Programmierumgebungen wie *Scratch* eröffnen einen niederschweligen Zugang zur Softwareentwicklung. Die Option, Figuren zu erstellen, mit Kostümen zu bestücken und sie auf virtuellen Bühnen agieren zu lassen, bietet Raum für lustvolles und kreatives Storytelling (vgl. Resnick et al. 2009), was wiederum die Aneignung von Programmierkompetenzen unterstützt (vgl. Vinayakumar et al. 2018).

Auch im Bereich Robotik und Physical Computing entstehen in informatischen Lernsettings kreative Produkte (vgl. Tengler 2020). Guven et al. (2020, S. 3) haben festgestellt, dass Schüler*innen beim Programmieren von Robotikanwendungen kreativ werden können. Physical Computing und Robotik bieten die Chance, ausgehend von Problemen des täglichen Lebens kreative Lösungen zu entwickeln. Dabei werden Algorithmen mit technischen Werkstoffen wie Zahnrädern,

Motoren und Sensoren verbunden und eigene datengetriebene Produkte entwickelt, die mit ihrer Umgebung interagieren. Schulische Making-Projekte bieten Schüler*innen zusätzlichen Raum, um eigene Projekte mit analogen und digitalen Technologien umzusetzen, Code unmittelbar produktbezogen zu entwickeln, zu testen und zu optimieren (vgl. Ingold et al. 2019).

Kreativität im Informatikunterricht hat mitunter auch künstlerische Facetten. In diesem Zusammenhang wird von ‚Creative Coding‘ (Levin/Brain 2021) oder ‚Creative Computing‘ (Brennan et al. 2014) gesprochen. Eine künstlerische Idee wird mithilfe der Entwicklung eines Algorithmus umgesetzt, der ein Artefakt erzeugt. Dieses Vorgehen wird ‚generative Gestaltung‘ genannt. In den Algorithmus werden häufig Zufallselemente eingeschrieben, so dass sich ein Spannungsfeld von Beliebigkeit und Steuerung ergibt. Ein Anwendungsbeispiel im Informatikunterricht sind *Art-Robots*, die einer Programmierung und einer mechanischen Konstruktion folgend Stifte über Papier bewegen und dort Spuren hinterlassen. Die Programmiersprache *Logo* geht in eine ähnliche Richtung. Hier werden Parameter und Algorithmen unter anderem genutzt, um eine Schildkröte – in diesem Fall ohne Zufallskomponente – so anzusteuern, dass sie ornamentale Figuren auf dem Bildschirm erzeugt. Diese Möglichkeiten werden auch in die physische Welt mit Zeichenrobotern (vgl. Baccaglini-Frank 2020) und Stickmaschinen (zum Beispiel *TurtleStitch*, vgl. Wolz 2019) übertragen.

4.4 KREATIVES UMFELD (PRESS)

Einigkeit besteht in der Annahme, dass Projekte für die Schüler*innen eine persönliche Relevanz

haben und lebensweltnah sein sollten (vgl. Brennan et al. 2020, S. 3, vgl. Romeike 2007, S. 59). Im besten Fall werden eigene Ideen umgesetzt oder Aufgabenstellungen selbst entwickelt (vgl. Romeike 2011, S. 369). Mork et al. (2020, S. 253) formulieren als Qualitätskriterien für kreativitätsfördernde informatische Lernaufträge unter anderem den Freiraum für eigene Entscheidungen sowie das Angebot von sogenannten ‚Ill-defined Problems‘, das heißt Problemen, für die es weder naheliegende Lösungen noch etablierte Lösungswege gibt. Komplexere Aufgabenstellungen verlangen einen multiperspektivischen Blick, was zur Stärkung kollaborativer Arbeitsformen in kreativen Teams führen kann (vgl. Bornemann 2011, S. 224). Kreativitätsfördernde Lernumgebungen unterstützen explorative Arbeitsformen durch ein niederschwelliges und risikoarmes Umfeld (‚Low-Risk Environment‘). Sie ermöglichen Erfolgserlebnisse nicht nur durch zielgerichtetes Arbeiten auf der Basis von systematischem Hintergrundwissen, sondern auch durch ein Herantasten im Sinne von Versuch und Irrtum.

In diesem Zusammenhang wird immer wieder auf die Online-Programmierungsumgebung *Scratch* verwiesen. „Children can start by simply tinkering with the bricks, snapping them together in different sequences and combinations to see what happens“ (Resnick et al. 2009, S. 63). Als griffige Bezeichnung für derartige informatische Lernumgebungen haben sich ‚Low Floor‘, ‚Wide Walls‘ und ‚High Ceiling‘ etabliert (vgl. Resnick/Silverman 2005, S. 117). Romeike sieht in den Programmierwerkzeugen wie *Scratch* auch die technische Grundlage für eine soziale Dimension von Kreativität. Sie bieten Schüler*innen Kollaborationsmöglichkeiten bei der Ideenfindung und -umset-

zung und ermöglichen eine einfache Verbreitung ihrer Werke in der Peergruppe (vgl. Romeike 2011, S. 364; vgl. auch Resnick et al. 2009).

Die Offenheit aller Akteur*innen, auch für unkonventionelle und ineffiziente Ideen, eine wertschätzende Atmosphäre und eine positive Fehlerkultur gelten unter anderem als Grundlage für die Entwicklung eines kreativen Selbstkonzepts im Informatikunterricht (vgl. Barnett/Romeike 2017, S. 302). Umgekehrt kann ein hoher, von der Lehrperson angelegter Funktionalitätsanspruch bewirken, dass die Schüler*innen bei der Produktentwicklung risikoarme und bewährte Wege einschlagen (vgl. Romeike 2007, S. 58). Kreativitätshemmendes Problemvermeidungsverhalten ist ferner im Kontext von Wettbewerben oder von benoteten Leistungsnachweisen zu erwarten (vgl. Knobelsdorf/Romeike 2008, S. 290).

5 DISKUSSION

Nach der Durchsicht ausgewählter informatikdidaktischer Texte kann festgehalten werden, dass Kreativität durchaus mehrperspektivisch – im Sinne des *4Ps of Creativity* Modells von Rhodes (1961) – diskutiert wird. Dabei wird Kreativität als eine Art Möglichkeitsraum konzeptualisiert. Schüler*innen bekommen die Gelegenheit aktiv zu werden, selbst zu gestalten, explorativ Software zu entwickeln, zu testen, zu optimieren und zu implementieren. Aus bildungstheoretischer Sicht erfreulich ist der Subjektbezug im Kreativitätsverständnis sowie die Haltung, jedem Menschen kreatives Potenzial zuzuschreiben. Programmierumgebungen wie *Scratch* sowie Lernanlässe im Bereich Physical Computing und Making schaffen Voraussetzungen für kreativen

Ausdruck (Schmalfeldt 2020). Der Fokus liegt dabei häufig auf narrativer Kreativität (Storytelling), Gamification (Game-Entwicklung) und auf künstlerischem Ausdruck (Animationen, Zufallskunst). Kreative Zugänge schaffen den motivationalen und sinngebenden Rahmen (eigene Idee für eine Geschichte, für ein Spielziel ...) für die Auseinandersetzung und den aktiv-gestalterischen Umgang mit informatischen Lerngegenständen. Kreative Prozessmodelle wie *Creativity Spiral*, *Design Thinking* und *Challenge Cycle* bieten Orientierung bei der Entfaltung von Kreativität. Neben zahlreichen Gemeinsamkeiten (Phasen, Agilität durch Iteration, zyklische Struktur) weisen die drei Prozessmodelle einige Unterschiede auf, die in Bezug auf Kreativität didaktisch reflektiert werden sollten. Resnick beschreibt mit der *Creativity Spiral* den individuellen Prozess einer Person im Rahmen einer außerschulischen Aktivität (zum Beispiel in einem communityorientierten MakerSpace). Er betont spielerische Formen von Kreativität und inkludiert das Scheitern als wichtigen Bestandteil des Lernprozesses. Damit grenzt er sich von der Position Hubwiesers (2007, S. 18) ab, der „gelegentliche Fehlschläge“ im Kontext von Kreativität lediglich „in Kauf nehmen“ möchte. Resnick leitet aus seinem Prozessmodell keine didaktischen Konsequenzen ab, sondern geht davon aus, dass bereits die Abfolge der Phasen in Verbindung mit einer hohen intrinsischen Motivation Raum für Kreativität öffnet. Schüler*innen, die sich mit der Entwicklung eigener Ideen (anfangs) eher schwertun, könnten vom *Design Thinking* Ansatz profitieren. Der *Design Thinking* Ansatz hat durch das gemeinschaftliche Ausgangsproblem und durch den Empathiefokus zu Beginn einen höheren Strukturierungsgrad

aber auch einen weniger starken Subjektbezug. Die Entwicklung von Persona wird in der Regel von der Lehrperson initiiert. Reflexionsanlässe ergeben sich nicht in erster Linie im individuellen Spiel mit Materialien oder durch das Testen von Prototypen, sondern durch den kontinuierlichen Abgleich mit den Bedürfnissen der Zielgruppe. Die Orientierung an realen Problemen bietet die Chance, informatische Kreationen in einen sinnhaften Anwendungskontext zu stellen und etwas zu entwickeln, das für Dritte nützlich ist. Dies ermöglicht insbesondere sozial interessierten Schüler*innen Selbstwirksamkeit.

Der für den schulischen Kontext entwickelte *Challenge Cycle* zielt auf die Förderung informatisch-technischer Kreativität. Die Lehrperson greift steuernd ein, indem sie den Fokus auf die Realisierbarkeit einer Challenge auf der Basis der vorhandenen informatischen Kompetenzen legt – freilich ohne den Anspruch aufzugeben, dass die Schüler*innen ihre Challenge selbst entwickeln. Nach Möglichkeit wäre die Wahl des Prozessmodells nicht der Lehrperson zu überlassen, sondern den Schüler*innen, die sich auf der Grundlage ihrer Interessen und Voraussetzungen für ein Ausgangsproblem und eine Vorgehensweise entscheiden. In diesem Zusammenhang gilt es zu reflektieren, bei welchen Schüler*innen ein bestimmtes Modell Kreativität ermöglicht, fördert oder auch voraussetzt. Unter Umständen können iterative Prozesse stark repetitiv sein, indem die Schüler*innen von ihnen selbst bereits entwickelte Produkte reproduzieren, was in der Maker Education unter dem Schlagwort ‚Keychain-Effekt‘ (vgl. Blikstein 2013) bekannt ist.

Der Fokus auf Kreativität in der Informatikdidaktik liegt nicht nur auf dem Prozess, sondern

auch auf dem Produkt (vgl. Barnett/Romeike 2017, S. 305). In einigen Artikeln wird anhand von Beispielen aufgezeigt, welche ‚kreativen‘ Produkte die Schüler*innen im Unterricht erstellen können (vgl. Wolz 2019; Komm 2020). Inwieweit maschinell gestickte Ornamente, Entwicklungen mit Physical Computing oder mit Robotern tatsächlich kreativ sind – und somit den Kriterien der (subjektiven) Originalität *und* der Adäquatheit entsprechen –, wird jeweils nicht ausgeführt. Zwar wird auf die ‚Lauffähigkeit von Programmen‘ (adäquat) und auf die ‚individuelle Note‘ (originell) verwiesen, es gibt aber weder Instrumente, um diese Form von Kreativität sichtbar zu machen, noch wird ausgewiesen, wie sich Kreativität – in Relation zu den individuellen Voraussetzungen der Schüler*innen – in den Produkten konkret niederschlägt. So erscheint Kreativität weniger Ziel, sondern vielmehr Mittel zum Zweck. Aus fachdidaktischer Sicht ist diese Haltung nachvollziehbar und prinzipiell spricht nichts gegen einen lernwirksamen und attraktiven Informatikunterricht mit Freiräumen für narrative oder gestalterische Kreativität. Informatikdidaktische Anliegen ließen sich aber noch stärker mit Kreativitätsförderung verbinden, wenn die Schüler*innen zur Entwicklung kreativer informatisch-technischer Lösungen angeregt würden. Dazu müsste allerdings klar sein, was „P-Kreativität“ (Romeike 2008a) oder „Little-C-Creativity“ (Kaufman/Beghetto 2009) bezogen auf die Entwicklung von Algorithmen oder Physical Computing Anwendungen bedeuten kann.

In der Software-Entwicklung wird das Rad selten neu erfunden. Codefragmente werden oftmals nicht selbst programmiert, sondern aus be-

stehenden Bibliotheken entnommen, kombiniert und an das eigene Problem angepasst (vgl. Barnett/Romeike 2017, S. 302). In dieser Perspektive wäre kreativ, geeignete Codefragmente und Algorithmen zu finden, sie technisch zu modifizieren und – falls nötig – mit selbst entwickeltem Code zu ergänzen. Standardisierung schließt Originalität und somit auch Kreativität nicht aus, sondern kann sie auch befördern. Da bestehende Bibliotheken in vielen Fällen die Bedürfnisse nicht exakt abdecken, bedarf es einer Modifikation eines Algorithmus oder einer Bibliothek, damit sie im gewünschten Kontext funktionieren. Nicht selten müssen originelle Workarounds entwickelt oder Verwertungstricks ‚out of the box‘ angewandt werden. Die Frage lautet: Was lässt sich mit den verfügbaren informatisch-technischen Möglichkeiten erreichen, indem man sie clever, effizient und weitsichtig einsetzt? Folgendes Beispiel aus der Unterrichtspraxis soll diesen Gedanken konkretisieren. *Calliope mini* ist ein Microcomputer für den Bildungsbereich, der sich durch eine vereinfachte Bedienung auszeichnet, was jedoch auch technische Limitationen nach sich zieht. Wenn Schüler*innen an offenen Projekten arbeiten, wie beispielsweise im Bereich von Umweltdaten, können diese Beschränkungen die technische Umsetzung der Ideen der Schüler*innen erschweren. So ist das gleichzeitige Versenden von zwei Zahlenwerten – zum Beispiel die Sensorwerte für den CO₂-Gehalt und für die Feinstaubmenge in der Umgebungsluft – von *Calliope mini* zu *Calliope mini* über Funk nicht möglich. Zwar können zwei Werte hintereinander versendet werden, jedoch ist für die Erfüllung der Gleichzeitigkeit ein ‚Trick‘ erforderlich: Die beiden analogen Sensoren geben jeweils Werte zwischen 0 und maximal

1.023 aus. Eine Möglichkeit für die gleichzeitige Versendung wäre, den ersten Wert mit 10.000 zu multiplizieren und den zweiten Wert zu addieren. Das Ergebnis dieser Rechnung kann nun über Funk versendet werden. Auf dem Empfänger-*Calliope mini* werden die notwendigen Umkehrrechenoperationen durchgeführt und die beiden Werte sind wieder einzeln nutzbar. Dieses Vorgehen zeigt eine von mehreren Möglichkeiten, die beschriebene technische Limitation kreativ zu umgehen. Dazu benötigen die Schüler*innen das Wissen über verschiedene Repräsentationen von Daten. Ebenso müssen sie abwägen, ob dieser ‚Trick‘ zu neuen Problemen führen könnte und andere Lösungsmöglichkeiten geeigneter wären. Beim ‚Recyclen‘ von Codefragmenten aus Bibliotheken in der Informatik werden Codezeilen übernommen, die für das gegebene Problem nicht relevant sind. Dies rührt unter anderem daher, dass Identifikation und Eliminierung überflüssiger Code-Zeilen aufwendig sind. Im Betrieb sind solche Programme fehleranfälliger und benötigen mehr Energieressourcen als optimierter oder generisch erzeugter Code. Eine kreative Herausforderung in diesem Zusammenhang wäre, einen schlanken Algorithmus zu entwickeln, der das Problem möglichst effizient löst.

Ein weiterer möglicher Zugang zu informatisch-technischer Kreativität – im informatikdidaktischen Diskurs bislang kaum präsent – ist das Hacking. Der Begriff Hacking ist hier nicht negativ im Sinne des unerlaubten Eindringens in Computersysteme in meist böser Absicht gemeint. Pias (2014, S. 15) beschreibt die Anfänge des Hackings wie folgt: „Es war das Herumspielen, das Ausprobieren, gepaart mit einer gewissen Respektlosigkeit gegenüber Vorschriften, System-

verwaltern oder Nutzungskontexten, welches das Hacken bestimmte“. Genner (2021, S. 2) streicht die unkonventionelle Anwendung von Technologie heraus, um durch die Transformation ihrer gewohnten Verwendung neue Perspektiven der Interaktion zu eröffnen. Dieses Verfahren ist nicht nur mit Software möglich, sondern auch mit Hardware (Dörig 2014, S. 27). Inspirierende Beispiele hierfür wären der Umbau eines Roboters zu einem Zeichenroboter, das Befüllen eines Druckers mit leitfähiger Tinte, um Leiterbahnen auszudrucken, die Befestigung eines Bechers auf einem Taster, der auslöst, wenn genügend Wasser im Becher ist, oder die Nutzung der einstellbaren Funkstärke des Microcontrollers *Calliope mini* als Distanzmesser. Ziel wäre, dass Schüler*innen auf dieser Grundlage eigene Hacking-Ideen entwickeln.

Aus bildungstheoretischer Perspektive gilt es, neben dem Erwerb von Fachkompetenzen und der Entwicklung kreativer Fähigkeiten, auch die Fähigkeit zum kritischen Denken und zum partizipativen und verantwortungsvollen Handeln zu stärken. Somit könnte Kreativität im Informatikunterricht nicht lediglich neutral, sondern als normatives Konstrukt betrachtet werden, das auf humanethischen Prinzipien basiert. Kreativität zeigt sich in dieser Perspektive nicht nur in originellen und gleichzeitig adäquaten Lösungen (wie auch z. B. intelligente Kampfroboter, Entwicklungen im Plattformkapitalismus ...), sondern auch im Bewusstsein, negative Folgen für Mensch, Gesellschaft und Umwelt abzuwenden. Informatisch-technischer Fortschritt ist von Ambivalenz geprägt, da er nicht notwendigerweise zur Problemlösung beiträgt, sondern nicht selten selbst Teil des Problems oder sogar dessen Ursache dar-

stellt. Bitcoins beispielsweise ermöglichen zwar die Abwicklung von Geldgeschäften unabhängig von Banken, verursachen durch das rechenaufwendige Mining aber einen hohen CO₂-Ausstoß. Kreativität im Informatikunterricht bewusst im Sinne einer nachhaltigen Entwicklung einzusetzen, setzt allerdings interdisziplinäres Lernen und Handeln voraus, wofür im Schulalltag Raum geschaffen werden muss. Potenzial in dieser Richtung bietet die Orientierung an den 17 UN-Zielen für nachhaltige Entwicklung in Verbindung mit *Design Thinking* Prozessen, wie es in der Maker Education seit einigen Jahren etablierte Praxis ist (vgl. z. B. Hampson/Marx 2019). Die Schüler*innen entwickeln dabei zu einem selbst gewählten UN-Nachhaltigkeitsziel mit analogen und digitalen Werkstoffen Prototypen, die zur Zielerreichung beitragen könnten. Auf diese Weise könnte nachhaltige Entwicklung – neben Storytelling, Animation und Gamedesign – mittelfristig ein weiterer sinnstiftender Rahmen für technische Kreativität und für die Erschaffung kreativer Produkte im Informatikunterricht werden. Damit ergäbe sich die Chance, Schüler*innen dafür zu sensibilisieren, digitale Technologie nicht nur innovativ, sondern auch weise und verantwortungsvoll und vor allem mit dem Blick auf den Wirkungszusammenhang von Produkten zu nutzen. So betrachtet wäre auch die oben skizzierte Aufgabenstellung, einen möglichst schlanken Algorithmus zu entwickeln, keine formale Fingerübung, sondern ein konzeptioneller Beitrag zur Einsparung von Energieressourcen.

6 SCHLUSSBEMERKUNG

Kreativität zählt zu den wesentlichen Voraussetzungen, um Zukunft aktiv und verantwortungsvoll mitzugestalten. Die Informatikdidaktik kann einen Beitrag zur Kreativitätsentwicklung leisten, indem sie fachliches Lernen mit kreativen Zugängen verbindet, Raum für narrative und gestalterische Kreativität gibt und den Schüler*innen gleichzeitig einen explorativen und lösungsorientierten Umgang mit informatisch-technischen Elementen nicht vorenthält.

Im Artikel lag der Fokus zu großen Teilen auf dem Programmieren und somit auf der Algorithmik, da in diesem Aspekt der Informatik die meisten Gestaltungsmöglichkeiten liegen. Mit Hacking und Physical Computing wurde außerdem ein möglicher kreativer Umgang mit Informatiksystemen angerissen (vgl. Brinda et al. 2008). Berührungspunkte zu den Bereichen Datenbanken und Datenstrukturen sind noch ausstehend. Hier würde sich zukünftig ein weiterer Blick lohnen, was Kreativität in diesem Kontext bedeuten könnte.

Ein normativ gefärbtes Kreativitätsverständnis mit dem Zielhorizont einer nachhaltigen Entwicklung hilft, technische Fingerübungen zu vermeiden und divergente wie konvergente Denkopoperationen der Schüler*innen in einen gesellschaftlich bedeutsamen und sinnstiftenden Kontext zu stellen. Eine auf diese Weise ausgerichtete kreative Informatikdidaktik hätte transdisziplinäre Bezüge und Überschneidungen mit dem Konzept der ‚Gestaltungskompetenz‘ von de Haan (2002) und kann dazu beitragen, dass Kreativität perspektivisch nicht ausschließlich dem wirtschaftlichen Wohlergehen oder der Erhaltung der Wett-

bewerbsfähigkeit dient, sondern Schüler*innen dafür sensibilisiert werden, Zukunftsherausforderungen mit technologischen Mitteln auf ökologisch und sozial verträgliche Weise anzugehen. Mit diesem Kreativitätsverständnis lässt sich schließlich die Brücke zu einem medienpädagogischen Anliegen schlagen: Kreativen Selbstaussdruck mit einem verantwortungsvoll-gestaltenden Umgang mit Medialität bzw. mit digitaler Technologie zu verbinden.

Literatur

Baccaglioni-Frank, Anna E./Santi, George/Del Zozzo, Agnese/Frank, Eric (2020). Teachers' Perspectives on the Interwining of Tangible and Digital Modes of Activity with a Drawing Robot for Geometry. In: *Education Sciences*, 10 (12), S. 387–407.

Barnett, Paul Joseph/Romelke, Ralf (2017). Creativity within computer science. In: Kaufman, James C./Gijveanu, Vlad Petre/Baer, John (Hrsg.), *The Cambridge handbook of creativity across domains*. Cambridge: Cambridge University Press, S. 299–321.

Blikstein, Paulo (2013). Digital Fabrication and 'Making' in Education: The Democratization of Invention. In: Walter-Herrmann, Julia/Böching, Corinne (Hrsg.), *FabLabs: Of Machines, Makers and Inventors*. Berlin: transcript, S. 203–222.

Bornemann, Stefan (2012). *Kooperation und Kollaboration. Das Kreative Feld als Weg zu innovativer Teamarbeit*. Wiesbaden: Springer VS.

Brennan, Karen/Balch, Christan/Chung, Michelle (2014). *Creative Computing. Learner workbook*. United Kingdom: Createspace.

Brennan, Karen/Haduong, Paulina/Veno, Emily (2020). *Assessing creativity in computing classrooms*. Harvard.

Brinda, Torsten/Fothe, Michael/Friedrich, Steffen/Koerber, Bernhard/Puhlmann, Hermann/Röhner, Gerhard/Schulte, Carsten (2008). Grundsätze und Standards für die Informatik in der Schule. *Bildungsstandards Informatik für die Sekundarstufe I*. In: *LOG IN*, 28 (150/151).

- Dagièné, Valentina/Futschek, Gerald/Stupurienè, Gabrielè (2019). Creativity in Solving Short Tasks for Learning Computational Thinking. In: *Constructivist Foundations*, 14 (3).
- De Haan, Gerhard (2002). Die Kernthemen der Bildung für eine nachhaltige Entwicklung. In: *ZEP – Zeitschrift für internationale Bildungsforschung und Entwicklungspädagogik* 25, 1/2002. S. 13–20.
- Dittert, Nadine/Wajda, Kamila/Schelhowe, Heidi (2016). *Kreative Zugänge zur Informatik: Praxis und Evaluation von Technologie-Workshops für junge Menschen*. Bremen.
- Dittert, Nadine/Robinson, Sarah/Thestrup, Klaus/Knudsen, Jacob (2019). Supporting creativity and collaboration. In: Iivari, Netta/Kinnula, Marianne/Rajanan, Mikko (Hrsg.), *Proceedings of the conference on Computing, Design and Making in Education – FabLearn Europe '19*. New York: ACM, S. 1–3.
- Dörig, Raffael (2014). *Gehackter Alltag*. In: Landwehr, Dominik (Hrsg.), *Hacking*. Basel: Christoph Merian Verlag, S. 24–33.
- Forthmann, Boris (2019). Die Beurteilung von Ideenqualität. In: Haager, Julia Sophie/Baudson, Tanja Gabriele (Hrsg.), *Kreativität in der Schule – finden, fördern, leben*. Wiesbaden: Springer, S. 75–95.
- Gerner, Alexander (2021). *On Hacking Cultures*. DOI: 10.13140/RG.2.2.19701.09441.
- Guven, Gokhan/Cakir, Nevin Kozcu/Sulun, Yusuf/Cetin, Gurcan/Guven, Emine (2020). Arduino-assisted robotics coding applications integrated into the 5E learning model in science teaching. *Journal of Research on Technology in Education*.
- Haager, Julia Sophie (2019). Bestandsaufnahme deutscher Schulbildung – Warum Kreativität nun Schule macht. In: Haager, Julia Sophie/Baudson, Tanja Gabriele (Hrsg.), *Kreativität in der Schule – finden, fördern, leben*. Wiesbaden: Springer, S. 193–206.
- Hampson, Gabi/Marx, Steven (2019). *WILMA – Wie lernen durch Machen. Eine Erfinderwerkstatt für Kinder und Jugendliche*. In: Ingold, Selina/Maurer, Björn/Trüby, Daniel (Hrsg.), *Chance MakerSpace. Making trifft auf Schule*. München: kopaed, S. 139–153.
- Hartmann, Werner/Näf, Michael/Reichert, Raimond (2007). *Informatikunterricht planen und durchführen*. Berlin: Springer.
- Hubwieser, Peter (2007). *Didaktik der Informatik: Grundlagen, Konzepte, Beispiele*. Berlin: Springer.
- Humbert, Ludger (2006). *Didaktik der Informatik*. Wiesbaden: Teubner.
- Ingold, Selina/Maurer, Björn/Trüby, Daniel (Hrsg.) (2019). *Chance MakerSpace. Making trifft auf Schule*. München: kopaed.
- Jackson, Philip W./Messick, Samuel (1965). The person, the product, and the response: conceptual problems in the assessment of creativity. In: *Journal of Personality*, 33 (3), S. 309–329.
- Jordanous, Anna (2016). Four PPPerspectives on computational creativity in theory and in practice. In: *Connection Science*, 28 (2), S. 194–216.
- Kaufman, James C./Beghetto, Ronald A. (2009). Beyond Big and Little: The Four C Model of Creativity. In: *Review of General Psychology*, 13 (1), S. 1–12.
- Kay, Ken (2010). 21st Century Skills: Why They Matter, What They Are, and How We Get There. In: Bellanca, James/Brandt, Ron (Hrsg.), *21st Century Skills. Rethinking How Students Learn*.
- Kerguenne, Annie/Schaefer, Hedi/Taberivand, Abraham (2017). *Design Thinking. Die agile Innovations-Strategie*. Freiburg: Haufe.
- Knobelsdorf, Maria/Romeike, Ralf (2008). Creativity as a pathway to computer science. In: Amillo, June (Hrsg.), *Proceedings of the 13th annual conference on Innovation and technology in computer science education – ITiCSE '08*. New York: ACM, S. 286–290.
- Komm, Dennis/Regez, Adrian/Hauser, Urs/Gassner, Marco/Lütscher, Pascal/Puchegger, Rico/Kohn, Tobias (2020). *Problem Solving and Creativity: Complementing Programming Education with Robotics*. In: Giannakos, Michail/Sindre, Guttorm/Luxton-Reilly, Andrew/Divitini, Monica (Hrsg.), *Conference on Innovation and Technology in Computer Science Education*. New York: ACM, S. 259–265.
- Levin, Golan/Brain, Tega (2021). *Code as creative medium. A handbook for computational art and design*. Cambridge: MIT Press.
- Meinel, Christoph/Leifer, Larry/Plattner, Hasso (Hrsg.) (2011). *Design thinking*. Berlin: Springer.
- Mishra, Punya/Yadav, Aman (2013). Rethinking Technology & Creativity in the 21st Century. In: *TechTrends*, 57 (3), S. 10–14.
- Modrow, Eckart/Strecker, Kerstin (2016). *Didaktik der Informatik*. Walter de Gruyter.

- Mork, Kirsten/Wilcox, John/Wood, Zoë (2020). Creative Choice in Fifth Grade Computing Curriculum. In: Giannakos, Michail/Sindre, Guttorm/Luxton-Reilly, Andrew/Divitini, Monica (Hrsg.), Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education. New York: ACM, S. 252–258.
- Niesyto, Horst (2006). film kreativ: Aktuelle Beiträge zur Filmbildung. München: kopaed.
- O’Callaghan, Gemma/Connolly, Cornelia (2020). Developing creativity in computer science initial teacher education through design thinking. In: Maguire (Hrsg.), United Kingdom & Ireland Computing Education Research conference. ACM, S. 45–50.
- Pias, Claus (2014). Kulturgeschichte des Hackens. In: Landwehr, Dominik (Hrsg.), Hacking. Basel: Christoph Merian Verlag.
- Resnick, Mitchel (2008). Sowing the Seeds for a More Creative Society. In: Learning & Leading with Technology, 35 (4), S. 18–22.
- Resnick, Mitchel/Maloney, John/Monroy-Hernández, Andrés/Rusk, Natalie/Eastmond, Evelyn/Brennan, Karen/Millner, Amon/Rosenbaum, Eric/Silver, Jay/Silverman, Brian/Kafai, Yasmin (2009). Scratch: Programming for all. In: Communications of the ACM, 52 (11), S. 60–67.
- Resnick, Mitchel/Silverman, Brian (2005). Some reflections on designing construction kits for kids. In: Eisenberg, Mike (Hrsg.), Proceedings of the 2005 conference on Interaction design and children. New York: ACM, S. 117–122.
- Rhodes, M. (1961). An analysis of creativity. Phi Delta Kappan, 42, S. 305–310.
- Romeike, Ralf (2007). Kriterien kreativen Informatikunterrichts. In: Didaktik der Informatik in Theorie und Praxis – INFOS 2007-12. GI-Fachtagung Informatik und Schule.
- Romeike, Ralf (2008a). Kreativität im Informatikunterricht. Potsdam. (Dissertation).
- Romeike, Ralf (2008b). What’s my challenge? The forgotten part of problem solving in computer science education. In: Mittermeir, Roland T./Sysło, Maciej M. (Hrsg.), Informatics Education – Supporting Computational Thinking. Berlin: Springer, S. 122–133.
- Romeike, Ralf (2011). Kreativität im Informatikunterricht. In: Schubert, Sigrid/Schwill, Andreas (Hrsg.), Didaktik der Informatik. Heidelberg: Spektrum Akademischer Verlag, S. 355–376.
- Runco, Mark A./Chand, Ivonne (1995). Cognition and creativity. In: Educational psychology review, 7 (3), S. 243–267.
- Schelhowe, Heidi (2018). Vom digitalen Medium und vom Eigen-Sinn der Dinge. Was Medienpädagogik mit der informatischen Bildung gewinnen kann. In: merz | medien + erziehung, 62 (4), S. 27–33.
- Schmalfeldt, Thomas (2020). Kompetenzaufbau und Kreativitätsförderung im Informatikunterricht mithilfe von Story Cards und Skill Cards. In: MNU Journal, (05.2020), S. 394–400.
- Schmoelz, Alexander/Barberi, Alessandro/Ollinger, Isabella/Krause, Sabine (2017). Editorial 4/2017: Kreativität/Ko-Kreativität. In: Medienimpulse, 55 (4).
- Schubert, Sandra/Loderer, Kristina (2019). Wie erkennt man Kreativität? Subjektive Beobachtung und objektive Messung. In: Haager, Julia Sophie/Baudson, Tanja Gabriele (Hrsg.), Kreativität in der Schule – Finden, Fördern, Leben. Springer, S. 39–74.
- Schubert, Sigrid/Schwill, Andreas (Hrsg.) (2011). Didaktik der Informatik. Heidelberg: Spektrum Akademischer Verlag.
- Stein, Morris I. (1953). Creativity and culture. In: Journal of Psychology, 36, S. 311–322. DOI: 10.1080/00223980.1953.9712897
- Tengler, Karin (2020). Klein, kreativ, Ozobot: Förderung von Kreativität und informatischem Denken durch spielerisches Programmieren. In: R&E-SOURCE.
- Veit, Daniel (2017). Der Wandel der Arbeitswelt im Zeitalter der Digitalisierung. In: ifo Schnelldienst 7/2017. S. 12–15.
- Vinayakumar, R./Soman, K.P. (2018). Digital Storytelling Using Scratch: Engaging Children Towards Digital Storytelling. Conference Paper.
- Wing, Jeannette M. (2006). Computational thinking. In: Communications of the ACM, 49 (3), S. 33–35.
- Wolz, Ursula/Auschauer, Michael/Mayr-Stalder, Andrea (2019). Programming embroidery with TurtleStitch. In: ACM SIGGRAPH 2019 Studio. New York: ACM, S. 1–2.
- Yadav, Aman/Cooper, Steve (2017). Fostering creativity through computing. In: Communications of the ACM, 60 (2), S. 31–33.