

Institut für Informatik  
Didaktik der Informatik

---

# **Kreativität im Informatikunterricht**

**Dissertation**  
zur Erlangung des akademischen Grades  
"doctor rerum naturalium"  
(Dr. rer. nat.)  
in der Wissenschaftsdisziplin "Didaktik der Informatik"

eingereicht an der  
Mathematisch-Naturwissenschaftlichen Fakultät  
der Universität Potsdam

von  
Ralf Romeike

Potsdam, den 28. August 2008



# **Gliederung**

## **Kapitel 1: Einleitung**

## **Kapitel 2: Kreativität**

- (1) Eine Leistung ist als kreativ zu bezeichnen, wenn sie zu persönlich neuen und verwendbaren Ideen, Lösungen oder Erkenntnissen führt.
- (2) Die Befähigung zu kreativem Handeln kann als höchstes Lernziel verstanden werden.
- (3) Kreativität und intrinsische Motivation stehen in einem wechselseitigen Verhältnis; dies kann im Unterricht als Chance aufgefasst werden.

## **Kapitel 3: Kreativität in der Informatik**

- (1) Informatik verkörpert Kreativität in verschiedenen Perspektiven: insbesondere in Fach-, Mensch-, Kunst-, Technologie-Perspektive und artifizieller Kreativität.
- (2) Jede Perspektive impliziert methodische oder thematische Anknüpfungspunkte für einen kreativen Informatikunterricht.

## **Kapitel 4: Kreativität in der informatischen Bildung**

- (1) Kreativität kann in der informatischen Bildung zur Verbesserung von Motivation und Leistung instrumentalisiert werden.
- (2) Eine zu eng ausgelegte Problemorientierung in der Praxis des Informatikunterrichts beschneidet diesen um seine kreativen Möglichkeiten.

## **Kapitel 5: Faktorenmodell kreativen Lernens im Informatikunterricht**

- (1) Charakteristische Einflussfaktoren kreativen Lernens im Informatikunterricht können durch ein Faktorenmodell beschrieben werden.
- (2) Kreative informatische Prozesse stellen das zentrale Element in einem kreativen Informatikunterricht dar.
- (3) Schüler, Fach sowie Informations- und Kommunikationstechnologien stellen vernetzte Faktoren dar, die Kreativität im Informatikunterricht fördern.

## **Kapitel 6: Untersuchungen zum Faktorenmodell**

- (1) Die Faktoren des vorgeschlagenen Modells lassen sich in Biographien von Studienanfängern der Informatik nachweisen.
- (2) Faktoren und Zusammenhänge des Modells werden in den Sichtweisen eines kreativen Schülers deutlich.
- (3) Zukünftige Informatiklehrer wertschätzen Kreativität, haben allerdings unkonkrete Vorstellungen von der Umsetzung im Informatikunterricht.

## **Kapitel 7: Kriterien kreativen Informatikunterrichts**

- (1) Die aus dem Faktorenmodell resultierenden Anforderungen an einen kreativen Informatikunterricht werden durch Kriterien operationalisiert.
- (2) Die Anwendung der Kriterien bei der Evaluation publizierter Unterrichtsvorschläge verdeutlicht den bisher geringen Stellenwert von Kreativität im Informatikunterricht.

## **Kapitel 8: Anwendung in einem Unterrichtsbeispiel**

- (1) Ein Unterrichtsbeispiel zur Einführung in die Programmierung verdeutlicht exemplarisch die Umsetzung eines kreativen Informatikunterrichts.

## **Kapitel 9: Evaluation des Unterrichtsbeispiels**

- (1) Die Durchführung und Evaluation des Unterrichtsbeispiels zeigt eine Steigerung von Motivation, Interesse und Leistungen und eine positive Veränderung des Informatikbilds im Pre-Post-Vergleich und im Vergleich mit dem Kontrollkurs.
- (2) Die Schüler beschreiben Informatik nun als Fach, in dem Kreativität benötigt wird und das Kreativität ermöglicht.

## **Kapitel 10: Zusammenfassung, Diskussion und Ausblick**

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>1</b>
1.1	Ausgangslage.....	1
1.2	Stellenwert der Kreativität in der Informatikdidaktik .....	1
1.3	Ziele.....	4
1.4	Struktur der Arbeit.....	4
<b>2</b>	<b>Kreativität</b> .....	<b>6</b>
2.1	Wissenschaftliche Forschungsansätze zur Kreativität.....	6
2.1.1	Entwicklung .....	6
2.1.2	Das 4-P-Modell.....	7
2.1.3	Ursprungsorientierte Sichtweisen .....	11
2.1.4	Unterscheidung verschiedener Ausprägungen von Kreativität .....	12
2.2	Motivation und Kreativität .....	13
2.3	Begriffsbestimmung .....	15
2.4	Kreativitätsförderung.....	17
2.5	Kreativität in der Lernwissenschaft.....	18
2.6	Schule und Kreativität .....	19
2.7	Kreativität in naturwissenschaftlichen Unterrichtsfächern.....	20
2.7.1	Mathematik .....	20
2.7.2	Physik.....	22
2.7.3	Chemie .....	22
2.7.4	Science Education.....	23
2.8	Zusammenfassung und Bewertung.....	23
<b>3</b>	<b>Kreativität in der Informatik</b> .....	<b>25</b>
3.1	Fach-Perspektive .....	25
3.1.1	Standpunkte zur Kreativität in der Informatik .....	26
3.1.2	Kreative Persönlichkeiten der Informatik .....	28
3.1.3	Kreativität in der Softwareentwicklung .....	29
3.1.4	Das Bausteinprinzip in der Informatik als Katalysator für Kreativität .....	33
3.1.5	Innovationen der Informatik .....	34
3.2	Mensch-Perspektive .....	35
3.3	Kunst-Perspektive .....	36
3.3.1	Informatiker als Künstler .....	37
3.3.2	Künstler als Informatiker .....	40
3.3.3	Symbiose von Informatik und Kunst .....	42
3.3.4	Kritik .....	42
3.4	Technologie-Perspektive .....	43
3.5	Artifizielle Kreativität .....	45
3.6	Situierung der Sichtweisen im Informatikunterricht .....	46

<b>4</b>	<b>Kreativität in der informatischen Bildung .....</b>	<b>48</b>
4.1	Zur Wahrnehmung von Kreativität in der Informatik durch Schüler .....	48
4.2	Kreativität in der Forschung zur informatischen Bildung .....	49
4.3	Verwandte Ansätze und Projekte kreativen Lernens .....	52
4.4	Herausforderungen als kreativer Zugang zu Problemen.....	54
4.4.1	Informatik-Lernen hat sich verändert.....	54
4.4.2	Probleme des Problemlöseansatzes in der Praxis.....	55
4.4.3	Ansätze zur Bewältigung der Probleme des Problemlösens .....	56
4.4.4	Der Challenge-Cycle .....	57
4.4.5	Schlussfolgerungen .....	59
<b>5</b>	<b>Faktorenmodell kreativen Lernens im Informatikunterricht .....</b>	<b>60</b>
5.1	Faktoren aus der Perspektive der Kreativitätsforschung .....	60
5.2	Faktoren aus der Perspektive der Informatik.....	61
5.3	Entwicklung des Faktorenmodells.....	62
5.4	Fazit .....	64
<b>6</b>	<b>Untersuchungen zum Faktorenmodell.....</b>	<b>66</b>
6.1	Kreativität in Biographien von Studienanfängern der Informatik .....	66
6.1.1	Biographien der Computernutzung .....	66
6.1.2	Forschungsfragen der Untersuchung.....	67
6.1.3	Analyseverfahren .....	67
6.1.4	Ergebnisse .....	69
6.1.5	Schlussfolgerungen .....	71
6.2	Kreativitätsverständnis und Erfahrungen von Informatikstudenten .....	72
6.3	Kreativität in der Schulinformatik aus Schülersicht .....	73
6.3.1	Ziel der Untersuchung .....	73
6.3.2	Methode.....	74
6.3.3	Ergebnisse .....	74
6.3.4	Fazit.....	76
6.4	Sichtweisen und Erfahrungen von Referendaren und Lehramtsstudierenden .....	76
6.4.1	Ergebnisse .....	77
6.4.2	Diskussion .....	78
6.5	Fazit der Untersuchung des Faktorenmodells.....	79
<b>7</b>	<b>Kriterien kreativen Informatikunterrichts .....</b>	<b>80</b>
7.1	Berücksichtigung der Fach-Dimension .....	80
7.2	Berücksichtigung der Schüler-Dimension .....	82
7.3	Anforderungen an die Unterrichtsumgebung .....	82
7.4	Kreativität in Unterrichtsbeispielen der Informatik.....	83
7.4.1	Zum Vorgehen.....	83
7.4.2	Kriterien der Fach-Dimension.....	84
7.4.3	Kriterien der Schüler-Dimension .....	84
7.4.4	Kriterien der Unterrichtsumgebung.....	85
7.4.5	Fazit.....	85

<b>8</b>	<b>Anwendung in einem Unterrichtsbeispiel .....</b>	<b>86</b>
8.1	Überlegungen zur Einführung in die Programmierung .....	86
8.2	Ein kreativer Einstieg in die Programmierung .....	87
<b>9</b>	<b>Evaluation des Unterrichtsbeispiels.....</b>	<b>89</b>
9.1	Auswertung der Unterrichtssequenz.....	89
9.1.1	Zum Vorgehen .....	89
9.1.2	Datenerhebung .....	89
9.1.3	Ergebnisse .....	90
9.1.4	Kritische Betrachtung .....	94
9.2	Untersuchung zum Verständnis von Kreativität in der Informatik .....	94
9.2.1	Datenerhebung und Analyse .....	95
9.2.2	Ergebnisse .....	96
9.2.3	Diskussion.....	98
9.3	Fazit.....	98
<b>10</b>	<b>Zusammenfassung, Diskussion und Ausblick .....</b>	<b>100</b>
10.1	Zusammenfassung .....	100
10.2	Diskussion .....	101
10.3	Ausblick .....	103
10.4	Fazit.....	104
<b>A</b>	<b>Unterrichtsbeispiel: Ein kreativer Einstieg in die Programmierung.....</b>	<b>105</b>
<b>B</b>	<b>Fragebogen zur Unterrichtserprobung .....</b>	<b>118</b>
<b>C</b>	<b>Lernerfolgskontrolle: Algorithmen und Programmierung .....</b>	<b>120</b>
<b>D</b>	<b>Arbeitsblatt zur Untersuchung des Verständnisses von Kreativität .....</b>	<b>123</b>
<b>E</b>	<b>Bildungsstandards kreativ erreichen – ein Unterrichtskonzept für die Sekundarstufe I...125</b>	
<b>F</b>	<b>Student Teachers’ Beliefs about Creativity.....</b>	<b>127</b>
	<b>Literaturverzeichnis.....</b>	<b>127</b>
	<b>Abbildungsverzeichnis.....</b>	<b>127</b>





# 1 Einleitung

## 1.1 Ausgangslage

Die Vorstellung ist nicht neu, dass wir unsere Schüler auf das Leben in einer sich verändernden Gesellschaft vorbereiten müssen, die immer mehr durch Informations- und Kommunikationstechnologien (IKT) beeinflusst wird und in der das verfügbare Wissen explodiert. Seit den 1990er Jahren wird diese neue Gesellschaft auch als Wissensgesellschaft bezeichnet. Vor diesem Hintergrund etablierte sich eine neue Vorstellung vom Lernen: das lebenslange Lernen. Viele Wissenschaftler setzen sich inzwischen damit auseinander und publizieren Ideen, wie die Schule diesen geänderten Anforderungen gerecht werden kann. Kreativität scheint in der Diskussion eine Schlüsselrolle zuzukommen. So verdeutlicht Resnick (2007b), dass Erfolg heutzutage nicht nur davon abhängt, was oder wie viel jemand weiß, sondern vor allem davon, ob er in der Lage ist, kreativ zu denken und zu handeln. Vor diesem Hintergrund nennt er die heutige Gesellschaft eine *Creative Society*.

Auch von deutschen Bildungspolitikern und Vertretern der Wirtschaft wird seit vielen Jahren gefordert, Schüler in allgemeinbildenden Schulen kreativer auszubilden; ein vorherrschender, zumeist auf konvergente Lösungsstrategien ausgerichteter Unterricht bereite die Schüler nicht ausreichend auf die späteren Anforderungen vor (vgl. Geser 1999).

Mit der Informatik existiert ein Unterrichtsfach, dem kreative Prozesse immanent sind und welches nicht nur durch die Anwendung von, sondern insbesondere durch die Auseinandersetzung mit kreativen Prozessen, z. B. Modellierungstätigkeiten, die Anwendung von Kreativität notwendig macht und fördert. Der besondere Beitrag der Informatik liegt hierbei nicht in einer weiteren kreativitätsfördernden Maßnahme, sondern in der spezifischen Ausprägung, in welcher kreative Handlungsformen in Denkprozesse integriert werden und somit von Schülern in ihren Handlungsweisen zu eigen gemacht werden können. Darüber hinaus eröffnen die Werkzeuge und Methoden der Informatik nicht nur Spezialisten der Informatik, sondern auch Laien vielfältige Möglichkeiten der kreativen Betätigung. Die kreative Beschäftigung mit Informatik an sich scheint nach eigenen Beobachtungen ein Grund für viele Schüler zu sein, sich überhaupt der Informatik zuzuwenden. Leider spiegelt sich diese Sichtweise der Informatik bisher in der Didaktik der Informatik kaum wider und wird damit vielen Schülern im Informatikunterricht kaum deutlich. So reflektieren Informatikcurricula und die Wahrnehmung von Informatik in der Öffentlichkeit eher selten das kreative Potential der Informatik. Stattdessen wird sie häufig als technisch, unkommunikativ und un kreativ angesehen.

Auch zu selten scheinen sich Informatiker und Informatiklehrer darüber Gedanken zu machen, was Kreativität in der Informatik eigentlich bedeutet und ausmacht. Fehlendes Wissen darüber, welche Entscheidungen und Methoden Schüler motivieren und befähigen, kreativ zu sein, kann aber Unsicherheit oder gar das Gegenteil von Kreativität bewirken.

## 1.2 Stellenwert der Kreativität in der Informatikdidaktik

Betrachtet man die Veröffentlichungen und Themen der deutschen Fachdidaktik, scheint Kreativität keinen besonderen Stellenwert zu besitzen – es existieren bis dato nahezu keine Veröffentlichungen im Themenbereich. Vor dem Hintergrund des populären Gebrauchs des Wortes *Kreativität* und dem Stellenwert als übergeordnetes Lernziel in der Schule wird Kreativität allerdings gelegentlich erwähnt – meist im Zusammenhang mit Wünschen oder Forderungen. Eine systematische Betrachtung existiert jedoch nicht.

Dennoch können in verschiedenen Arbeiten auch Aussagen zur Kreativität gefunden werden. Es bleibt allerdings zu beachten, dass, wenn auf Kreativität Bezug genommen wird, dies häufig oberflächlich und ohne Berücksichtigung der psychologisch-didaktischen Hintergründe geschieht. Schubert und Schwill (2004) unterstreichen die Notwendigkeit von Kreativität beim informatischen Modellieren:

*Es bestehen Missverständnisse über die kreativen Anforderungen, die der Mensch beim informatischen Modellieren und bei der rechnerverarbeitbaren Darstellung solcher Modelle erfüllen muss. Man nahm an, der kreative Spielraum des Menschen würde durch die Informatik stark eingeschränkt. Gezeigt hat sich, dass der Entwurf von Algorithmen und Datenstrukturen eine anspruchsvolle geistige Tätigkeit ist. (S. 108)*

Kreatives Denken wird als Teil der kognitiven Leistungen im informatischen Problemlöseprozess beschrieben; auf die erwähnten *kreativen Anforderungen* wird allerdings nicht weiter eingegangen. Darüber hinaus wird Kreativität an verschiedenen Stellen erwähnt, aber nicht weiter erläutert:

*Mit [der Top-Down-Methode] folgt man den kognitiven Empfehlungen für das Problemlösen und entwickelt persönliche Kreativität. (S. 109)*

Der Zusammenhang mit Kreativität wird an dieser Stelle nicht deutlich, zumal kreative Ansätze auch häufig induktiven bzw. Bottom-Up-Ansätzen folgen.

Schwer nachvollziehbar ist, wie die „*Reduzierung der Planungsentscheidung durch Empfehlung einer Verfahrensheuristik*“ mit „*Kreativität und Disziplin*“ in Einklang zu bringen sind (S.113). Die Beschreibung ist auch insofern problematisch, als Kreativität und Disziplin in verschiedenen Auffassungen von Kreativität eine Dichotomie darstellen, die nicht trivial aufzulösen ist (vgl. Glass 2006).

Im Weiteren betonen Schubert und Schwill die motivierende Wirkung kreativer Tätigkeiten im Informatikunterricht:

*Das konstruktive Element der Informatiklösung bringt stark motivierende Wirkung in den Lernprozess. Die Schüler besitzen fast grenzenlose Modellierungsmöglichkeiten. [...]*

*Die Tatsache, dass stets mehrere originelle und korrekte Lösungen gefunden werden können, die nicht mit der des Lehrers übereinstimmen müssen, erhöht bei kreativen Schülern die Freude am Experimentieren. (S. 114)*

Hubwieser (2000) zählt die Förderung von Kreativität sogar zu den „wichtigsten Zielen“ des Informatikunterrichts. Dennoch findet sich nur wenig Substanz in der Verwendung des Kreativitätsbegriffs. So wird Kreativität zwar häufig erwähnt, führt aber vor dem Hintergrund nicht näher erklärter Annahmen zu fragwürdigen Schlussfolgerungen, welche vor allem in den „Praxistipps“ deutlich werden (S. 18):

*„Förderung von Kreativität heißt vor allem Förderung von Aktivität.“*

Diese Annahme ist trügerisch. Aktivität ist zwar eine notwendige, aber nicht hinreichende Bedingung kreativen Lernens. Ein Blick in verschiedene Lehrbücher und Unterrichtssequenzen zeigt, wie diese Auffassung häufig falsch verstanden wird. So führen das Handeln nach vorgegebenen Mustern oder das Lösen von Problemen in eng vorgegebenen Kontexten und Verfahren nicht zu mehr Kreativität, sondern häufig zum Gegenteil.

*„Kreativität kann in einer fest vorgeplanten Umgebung nicht gedeihen, ein gewisser chaotischer Bereich ist unabdingbar (partielle Ordnungsstrukturen).“*

Dieses Argument wird leider häufig als Entschuldigung gebraucht, warum im regulären Unterrichtskontext Kreativität nicht berücksichtigt wird. Dabei ist auch in einem vorgeplanten unterrichtlichen Rahmen Kreativität möglich, ja das Einplanen kreativer Phasen durch die Lehrperson sogar angebracht, wünschenswert und kann durch die Gestaltung einer kreativitätsfördernden Umgebung unterstützt werden. Ebenso muss Selbstbestimmung in der Arbeit der Schüler (= eigenes Organisieren) nicht organisatorisches Chaos bedeuten. Im Gegenteil, Studien weisen darauf hin, dass geordnete, aber anregende und tolerante Umgebungen Kreativität begünstigen (vgl. Starko 2001).

*„Indirekter Lehrereinfluss fördert die Kreativität.“*

Dieser Praxistipp ist insofern richtig, als *direkter* Lehrereinfluss für kreatives Denken und Arbeiten hinderlich ist. Dies kann allerdings auch für indirekten Lehrereinfluss zutreffen. Lässt der Lehrer

Freiraum für freies Arbeiten, erwartet aber letztendlich ein funktionales Ergebnis, wird indirekt Druck oder Kontrolle ausgeübt und die Entfaltung von Kreativität nur schwer gelingen.

*„Stillarbeit, Schülervorträge oder Gruppenarbeit sind angemessene Arbeitsformen.“*

Kreatives Arbeiten bedeutet Ideen zu kommunizieren, anzuwenden und zu reflektieren. Dabei ist vor allem ein kommunikatives, nicht konkurrierendes, sondern gemeinsam anregendes Unterrichtsklima von Vorteil. Die klassische Stillarbeit, bei welcher der Lehrer anfangs Aufgaben austeilt und anschließend kontrolliert, lässt dagegen nur bedingt Kreativität zu. Auch Schülervorträge können – inhaltlich – nur begrenzt kreativ sein, zumal hier meist vorgegebenes Wissen wiedergegeben wird.

Auch Schulte (2003) hält es für wichtig, Kreativität in den Unterrichtskontext einzubinden. Im life<sup>3</sup>-Unterrichtskonzept werden Projekte unter anderem auch deshalb als bevorzugte Unterrichtsmethodik vorgeschlagen.

*Projekte erlauben es, die Kreativität und Gestaltungsideen der Schülerinnen und Schüler einzubinden, sie eigene Gestaltungserfahrungen machen zu lassen und deutlich werden zu lassen, dass unterschiedliche Entwürfe denkbar sind und je nach Ziel unterschiedliche Aspekte einer Situation modelliert werden müssen. (S. 53)*

So wurde das life<sup>3</sup>-Phasenmodell u. a. mit dem Ziel entwickelt, dass Softwareentwicklung, insbesondere informatisches Modellieren, Schülern als kreativer Prozess deutlich wird:

*Modellieren bedeutet mehr als das Notieren einer Lösung. Es gibt keine universell anwendbaren Idealrezepte, sondern situierte Lösungen und Heuristiken. (S. 174)*

Des Weiteren unterstreicht Schulte (2001) kreative Problemfindungsprozesse in der objektorientierten Analyse als Chance der Objektorientierung für einen besseren Informatikunterricht.

Laut Modrow (2002) herrscht *„Einigkeit [...] weitgehend darüber, dass der Informatikunterricht die Kreativität der Schülerinnen und Schüler stärken [...] soll.“* (S. 58)

Als Konsequenz dessen sollten die Schüler zumindest phasenweise Probleme analysieren und modellieren sowie die Modelle implementieren und testen.

Thomas (2002) erwähnt eine *„Stärkung des ‚Schüler-Ichs‘ durch die kreative und stark selbstbestimmte Konstruktion von informatischen Produkten, an denen sich Erfolg und Misserfolg unmittelbar erfahren lässt [...].“* (S. 6)

Weiterhin führt er aus, dass dieser Aspekt im Rahmen der Allgemeinbildungsdiskussion zum Informatikunterricht allerdings selten erwähnt wird. Schließlich stellt Thomas fest, dass informatisches Modellieren Schüler zu kreativen Tätigkeiten befähigen kann:

*Nicht zuletzt bietet das Modellieren von informatischen Modellen dem Schüler die Möglichkeit kreative Lösungen zu Problemen zu entwickeln, diese zu realisieren und zu überprüfen. (S. 71)*

Diethelm (2007) berichtet, dass bei der objektorientierten Modellierung eine *„realitätsnahe Herangehensweise [...] auf die Schüler extrem motivierend und Kreativität fördernd [wirkt].“* (S. 116)

Diese erreicht sie z. B. durch ein kleines Industriepraktikum mit Lego Mindstorms-Robotern. Aus ihren Erfahrungen mit dem Tool *Fujaba* berichtet Diethelm allerdings, dass dieses aufgrund mangelnder Intuitivität die Schüler in ihrer Kreativität gehemmt hat. Sie regt an, Kriterien für Lernumgebungen und IDEs zu entwickeln, die gelten müssen, um die Kreativität der Schüler zu fördern.

Immer wieder wurde Kreativität auch für den Informatikunterricht gefordert. So empfahl Friedrich (1995) schon vor 13 Jahren, Kreativität als Kriterium für die Auswahl von Unterrichtsthemen zu berücksichtigen, indem diese *„Möglichkeiten zur selbständigen kreativ-forschenden Erarbeitung der Problemlösung bieten.“*

Auch die Gesellschaft für Informatik e.V. (GI) verweist bereits in frühen Empfehlungen für den Informatikunterricht in der Sekundarstufe II auf das Potential, das kreatives Gestalten in der

Informatik für die Förderung schöpferischen Denkens bietet (GI 1993). Beispielhaft werden sprachliche und graphische Gestaltung und Modellierung angegeben.

Vor diesem Hintergrund bleibt es erstaunlich, dass es so wenige Publikationen im Bereich der Didaktik der Informatik gibt, die sich mit Kreativität im Informatikunterricht beschäftigen. Sucht man nach einer expliziten Thematisierung von Kreativität in der deutschen Fachzeitschrift zum Ideenaustausch von Informatiklehrern LOG IN, findet sich wenig. Allein in einem Heft aus dem Jahr 1995 wurde Kreativität im Zusammenhang mit dem Computer als Werkzeug zur Musik- oder Textgenerierung thematisiert. Die informatikspezifischen Möglichkeiten kreativen Tuns wurden dabei nicht berücksichtigt. Mangelndes Bewusstsein für kreative Unterrichtsphasen spiegelt sich auch in den in der LOG IN vorgeschlagenen Unterrichtssequenzen wider. Betrachtet man diese unter dem Aspekt kreativer Schülertätigkeiten, räumen nur wenige der vorgestellten Unterrichtsskizzen kreativen Handlungen explizit Raum ein. Mitunter wird sogar gefürchtet, dass Schüler „ihre Kreativität zu viel spielen lassen“ (Janneck 2006, S. 65). Tatsächlich sind unkreative Schüler im Unterricht für den Lehrer „handlicher“ als kreative Schüler (vgl. Westby und Dawson 1995; Sternberg und Lubart 1991), so dass Kreativität mitunter im Unterricht sogar unerwünscht ist.

### 1.3 Ziele

Die (leider noch zu selten geführte) Diskussion und Thematisierung von Kreativität in der Informatik und im Bezug auf den Informatikunterricht erfolgt bisher in der Regel unscharf, vorurteilsbehaftet, unreflektiert und teilweise wissenschaftlich unkorrekt oder unbegründet. Ziel dieser Arbeit ist es, die Diskussion auf eine wissenschaftliche Grundlage zu stellen, d. h. das Phänomen<sup>1</sup> der Kreativität im Informatikunterricht, und damit auch in der Informatik, zu konkretisieren, zu systematisieren, Vorteile und elementare Aspekte für den Informatikunterricht sowie bestimmende Merkmale herauszuarbeiten, zu operationalisieren sowie zu überprüfen.

Aus diesem Vorhaben ergeben sich folgende zentrale Fragen:

1. Welches Verständnis von Kreativität sollte aus Sicht der Fachdidaktik Informatik verwendet werden?
2. Wie manifestiert sich Kreativität in der Informatik bzw. im Informatikunterricht und welche Chancen ergeben sich daraus?
3. Wie muss Informatikunterricht gestaltet werden, damit er das Potential, das Kreativität zugesprochen wird, nutzt und fördert?
4. Welche Auswirkungen hat die explizite Berücksichtigung von Kreativität im Informatikunterricht auf das Lernerlebnis und den Lernerfolg?

Ziele der Arbeit sind damit weniger inhaltliche Fragen (die meiner Ansicht nach bereits an anderen Stellen ausgiebig diskutiert werden), sondern Fragen der Methodik, Unterrichtsorganisation und Thematik.

### 1.4 Struktur der Arbeit

Das komplexe Phänomen der Kreativität als Grundlage dieser Arbeit bedarf einer Aufarbeitung der Erkenntnisse der Kreativitätsforschung, der Klärung des Begriffs *Kreativität* und einer Einschätzung und Beurteilung aus informatikdidaktischer Sicht. Diese Diskussion wird in Kapitel 2 geführt. Insbesondere werden hierbei die verschiedenen Sichtweisen und Definitionsversuche erörtert, da sie maßgeblich die Vorstellungen von Kreativität prägen und für die Herausarbeitung eines Kreativitätsverständnisses für den Informatikunterricht grundlegend sind. Des Weiteren werden Forschungserkenntnisse zur Förderung und von Einflussfaktoren von Kreativität betrachtet sowie die Rolle von Kreativität in den Lernwissenschaften und in der Schule herausgearbeitet. Zur

---

<sup>1</sup> Vor dem Hintergrund, dass Kreativität im Informatikunterricht zwar auftritt, allerdings bisher noch nicht charakterisiert und hinterfragt wurde, kann von einem Phänomen gesprochen werden.

Orientierung erfolgt eine Aufarbeitung der Bemühungen anderer Fachdidaktiken, Kreativität in den naturwissenschaftlichen Unterricht zu integrieren.

Kapitel 3 ordnet die Diskussion zur Kreativität innerhalb der Informatik und schafft mit Hilfe verschiedener Sichtweisen eine Diskussionsgrundlage. Anhand der verschiedenen Verständnisse von Kreativität in der Informatik wird die Bedeutung von Kreativität in der Informatik verdeutlicht. Darüber hinaus wird auf die Frage eingegangen, welches Potential Kreativität für die Informatik bietet und welche Konsequenzen für den Informatikunterricht daraus erwachsen.

Nach Klärung dieser Hintergründe erfolgt in Kapitel 4 eine Betrachtung von Kreativität in der informatischen Bildung, insbesondere der internationalen Forschung zur Informatikdidaktik. Hierbei soll geklärt werden, welche Lösungsansätze es für die Integration von Kreativität in der informatischen Bildung gibt und welche Erfahrungen und Ergebnisse damit erzielt wurden. Das Kapitel schließt mit einer Analyse der Frage, warum Kreativität bisher im Informatikunterricht, v. a. beim Problemlösen, nur wenig Berücksichtigung findet, und einem Vorschlag, wie dies geändert werden könnte.

Vor dem Hintergrund der theoretischen Grundlagen der Kreativitätsforschung, der Analyse der Kreativität in der Informatik und der didaktischen Forschung werden in Kapitel 5 die herausgearbeiteten Faktoren von Kreativität in der Informatik in einem Faktorenmodell integriert, welches das Phänomen der Kreativität im Informatikunterricht zugänglich, greifbar und überprüfbar machen soll sowie als Grundlage für die Planung und Durchführung von kreativem Informatikunterricht dienen kann.

Eine Überprüfung des Faktorenmodells mittels verschiedener empirischer Studien erfolgt in Kapitel 6. Untersucht werden Computernutzungsbiographien von Studienanfängern, Äußerungen eines herausragend kreativen Informatikschülers sowie die Vorstellungen und Erfahrungen von Studienreferendaren und Lehramtsstudenten der Informatik hinsichtlich Kreativität.

In Kapitel 7 wird das Faktorenmodell durch Kriterien kreativen Informatikunterrichts operationalisiert. Die Kriterien werden anschließend herangezogen, um in einer empirischen Untersuchung den Stellenwert von Kreativität in publizierten Unterrichtsvorschlägen festzustellen.

In den Kapiteln 8 und 9 wird zur Verdeutlichung des Modells und der Kriterien ein Unterrichtsbeispiel vorgestellt, erprobt und dabei untersucht, wie sich eine kreative Unterrichtssequenz zur Einführung in die Programmierung auf die Schüler, deren Bild von der Informatik und deren Leistungen auswirkt.

In Kapitel 10 erfolgt eine abschließende Diskussion und Bewertung der Ergebnisse, eine Einordnung in den aktuellen Forschungsstand sowie ein Ausblick.

## 2 Kreativität

*Do something creative every day!*<sup>2</sup>

Kreativität ist vielleicht einer der am kontroversesten diskutierten Untersuchungsgegenstände der Psychologie und zugleich ein im allgemeinen Sprachgebrauch häufig und recht unbekümmert verwendeter Begriff.<sup>3</sup> Die wissenschaftliche Untersuchung von Kreativität grenzt sich teilweise deutlich vom alltäglichen Sprachgebrauch ab. In diesem Kapitel werden die Entwicklung und der aktuelle Stand der Kreativitätsforschung überblicksweise dargestellt. Hierzu werden zunächst wissenschaftliche Ansätze der Kreativitätsforschung erörtert, verschiedene Definitionsversuche kontrastiert, eine eigene, für diese Arbeit zu verwendende Definition herausgearbeitet sowie die Rolle der Kreativität in den Lernwissenschaften und in der Schule betrachtet. Erste Implikationen für den Informatikunterricht werden verdeutlicht.

### 2.1 Wissenschaftliche Forschungsansätze zur Kreativität

#### 2.1.1 Entwicklung

Die wissenschaftliche Untersuchung der Kreativität ist noch relativ jung.<sup>4</sup> Dennoch hat sie eine Vielzahl unterschiedlicher Ansätze und Sichtweisen hervorgebracht, aber auch viele Fragen noch nicht klären können (vgl. Mumford 2003). Eine erste systematische Studie zur Kreativität wurde von Galton (1869) im Rahmen seiner Untersuchungen über Genies durchgeführt. Als bald wendete sich die psychologische Forschung allerdings überwiegend der Intelligenz als Gegenstand des Forschungsinteresses zu; Kreativität wurde als Teil intelligenten Denkens verstanden. Die frühe Kreativitätsforschung Anfang des 20. Jh. kann grundsätzlich in vier verschiedene Traditionen zusammengefasst werden (vgl. Craft 2001):

**Die psychoanalytische Tradition:**

Kreativität als Trieb und zentrales intrinsisches Element des Menschen.

**Die behavioristische Tradition:**

Kreatives Verhalten als Reaktion auf spezielle Stimuli.

**Die humanistische Tradition:**

Kreativität als Teil der Selbstverwirklichung des Menschen mit dem Ziel, mit inneren Bedürfnissen und Potentialen im Einklang zu stehen.

**Die kognitive Tradition:**

Kreativität als Resultat assoziativer Prozesse und divergentem Denken.

Im Wesentlichen war diese frühe Forschung von philosophischer Spekulation geprägt; sie findet sich aber partiell auch in modernen Sichtweisen wieder.

Einen deutlichen Schub erfuhr die Kreativitätsforschung<sup>5</sup> in den 1950er Jahren. Joy Paul Guilford forderte in seiner Antrittsrede als Präsident der *American Psychological Association* eine verstärkte Betrachtung und Analyse divergenter Denkprozesse, welche er erstmals als Kreativität bezeichnete (Guilford 1950). Vor dem Hintergrund des Sputnikschocks<sup>6</sup> 1957 wurden zahlreiche Förderprogramme aufgelegt und die Forschung zur Kreativität intensiviert.

Trotz dieser vorübergehenden Intensivierung der Kreativitätsforschung trat alsbald Ernüchterung ein. So stellten die Forscher fest, dass die Variablen der Kreativität nur schwer zu fassen und nicht

---

<sup>2</sup> Werbeslogan eines Papier-/Bastelwaren-Kette in den USA, <http://www.paper-source.com>.

<sup>3</sup> Nicht zuletzt kann Kreativität auch als Modewort verstanden werden (vgl. Hentig 2000).

<sup>4</sup> Die systematische Untersuchung begann im Wesentlichen mit der 2. Hälfte des 20. Jahrhunderts.

<sup>5</sup> Vor allem in den USA.

<sup>6</sup> Durch den Start des ersten künstlichen Satelliten demonstrierte die Sowjetunion ihre technologischen Fähigkeiten.

isoliert zu betrachten sind. Infolgedessen blieb Kreativität ein über lange Zeit ignoriertes Forschungsthema.<sup>7</sup> Auch in Überblicks- oder Einführungswerken der Psychologie ist Kreativität nur marginal beschrieben.<sup>8</sup> Eine Ursache dafür könnte sein, dass Kreativität kaum in eine psychologische Tradition einzuordnen ist und verschiedene Teilgebiete der Psychologie, wie Persönlichkeits-, Entwicklungs-, Sozial- und Kognitive Psychologie, berührt. Vor dem Hintergrund eines unklaren und mehrdeutigen Kreativitätsbegriffs und unterschiedlichen Vorannahmen über kreatives Verhalten haben sich vielfältige Sichtweisen in der Kreativitätsforschung herausgebildet, die sich in unterschiedlichen Forschungsstrategien, Bezugssystemen und Kommunikationsformen äußern.

Eine häufig angewandte Möglichkeit, die unterschiedlichen Forschungsansätze zu charakterisieren, ist das 4-P-Modell, welches in Abschnitt 2.1.2 dargestellt wird. Hier wird der Fokus auf die unterschiedlichen Aspekte von Kreativität gelegt: die kreative Person, der kreative Prozess, das kreative Produkt und das kreative Umfeld (engl.: Press). Andere Sichtweisen berücksichtigen den Ursprung oder verschiedene Ausprägungen der Kreativität. Diese Sichtweisen werden in den Abschnitten 2.1.3 und 2.1.4 charakterisiert.

### 2.1.2 Das 4-P-Modell

Die Einteilung der Kreativitätsfaktoren hinsichtlich der „4 P“ wird Rhodes (1961) zugeschrieben und stellt die vier Faktoren da, die Kreativität maßgeblich beeinflussen sollen.

#### Die kreative Persönlichkeit

Ziel dieses Strangs der Kreativitätsforschung ist es, Persönlichkeitsmerkmale zu bestimmen, an denen kreative Personen erkannt und damit auch gefördert werden können. Darüber hinausgehende Überlegungen ziehen die allgemeine Förderung dieser Persönlichkeitsmerkmale in Erwägung. Trotz teilweise deutlich unterschiedlicher Forschungsergebnisse aufgrund verschiedener Voraussetzungen, können einige Merkmale kreativer Persönlichkeiten übereinstimmend angegeben werden:

Hohe Motivation, die Ausdauer und Bereitschaft, Hindernisse zu überwinden, intellektuelle Neugier und Offenheit für neue Erfahrungen, starker Einsatz, Nonkonformismus, Unabhängigkeit im Tun und Handeln, Toleranz für Unklarheiten, der Wille zu wachsen, Risikobereitschaft, der Glaube an Überzeugungen, Selbstsicherheit sowie breit gefächerte Interessen (vgl. Craft 2001; Sternberg und Lubart 1991; Funke 2000; Simonton 1999).

Kritisierend ist an diesen Ergebnissen zu erwähnen, dass die Untersuchungen an geschichtlich herausragend produktiven Persönlichkeiten vorgenommen wurden und damit nicht unmittelbar auf „gewöhnliche“ Kreativität anwendbar sind (Craft 2001). Zudem liegt den Studien keine einheitliche Definition von Kreativität zugrunde. Nichtsdestotrotz können die identifizierten Persönlichkeitseigenschaften in einem kreativen Unterricht als Orientierung dienen. Offenbar finden sich hier auch Persönlichkeitsmerkmale wieder, die häufig Informatikern zugeschrieben werden, aber im Schulkontext eher unerwünscht sind (z. B. Nonkonformismus).

#### Kreativitätsdiagnostik

„Vielleicht hat sich kein psychologisches Konzept als so schwierig zu messen erwiesen, wie die Kreativität“ (Krampe 1993). Guilford legte mit der Differenzierung des Denkprozesses in konvergentes und divergentes Denken, wobei er letzteres dem kreativen Denken zuordnete, die Grundlage für eine empirische Kreativitätsforschung. Anhand einzelner und vergleichbarer Faktoren von Kreativität entwickelte er ein Messinstrument, mit welchem er kreative Leistungen

<sup>7</sup> Nur 0,5% aller Artikel, die zwischen 1975 und 1994 in psychologischen Fachzeitschriften veröffentlicht wurden, behandeln das Thema (Sternberg, Pretz und Kaufman 2002).

<sup>8</sup> So wird bspw. im fast 1000-seitigen Standardwerk der Psychologie von Zimbardo und Gerrig (2004) Kreativität nur auf 3 Seiten unter dem Oberthema „Intelligenz“ thematisiert, wobei Seiten 2 und 3 sich mit „außergewöhnlicher Kreativität und Wahnsinn“ beschäftigen. Selbst in verschiedenen Lehrbüchern zur Lernpsychologie wird Kreativität kaum berücksichtigt (z. B. Edelman 1996; Lefrancois und Leppmann 1994).

von Personen zu erfassen versuchte. Ziel verschiedener Kreativitätstest ist es auch, das kreative Potential von Personen zu bestimmen. Die Arbeiten von Guilford wurden im Wesentlichen im „Torrance Test of Creative Thinking“ (TTCT) weiterentwickelt (Torrance 1974). Kern des TTCT ist das Messen unkonventioneller Antworten, also divergentem Denken. Die Aufgaben bestehen in der Regel aus der Beschreibung einfacher Situationen und dem Auftrag, möglichst viele Ideen zu äußern (siehe Abbildung 1).

**Ungewöhnliche Verwendungen:** Liste möglichst viele interessante und ungewöhnliche Verwendungsmöglichkeiten für Pappkartons auf.

**Stell dir vor,** Wolken hätten Fäden angeheftet, die zur Erde runterhängen. Was würde passieren? Stelle so viele Ideen und Vermutungen auf, wie du kannst.

**Verbesserung:** Lasse dir die cleversten, interessantesten und ungewöhnlichsten Wege einfallen, einen Plüsch-Spielzeugelefanten so zu verändern, dass Kinder mehr Spaß beim Spielen damit haben.

Abbildung 1: Aufgabenbeispiele aus dem TTCT (Version kreatives Denken mit Worten).

Bestehend aus einem figuralen und einem verbalen Test werden die zu testenden Personen mit Aufgaben konfrontiert, in denen sie z. B. eine Geschichte weitererzählen, ein Spielzeug weiterentwickeln oder Figuren ergänzen sollen. Der Test liefert Ergebnisse in den Kategorien Flüssigkeit (Anzahl der Antworten), Flexibilität (Anzahl der verschiedenen Antwortkategorien) und Originalität (statistische Seltenheit der Antwort)<sup>9</sup>.

Validität und Reliabilität von Kreativitätstests sind allerdings umstritten (vgl. Baer 1994a, b; Cropley 2000; Mumford 2003; Mumford, Marks et al. 1998; Seiffge-Krenke 1974). So werden zum einen auch Faktoren getestet, die eher auf die Intelligenz des Probanden zurückzuführen sind. Zum anderen ist es generell fraglich, ob sich Kreativität mittels eines Tests erfassen lässt. Kreative Leistungen sind, im Gegensatz zu anderen kognitiven Leistungen, stark von Umgebungsfaktoren abhängig. So können sich bspw. Stress und Zeitdruck in einer Testsituation negativ auf das Testergebnis auswirken. Ein gutes Abschneiden in einem Kreativitätstest bedeutet in der Regel eben nicht, dass auch außerhalb des Tests schöpferische Tätigkeiten ausgeführt werden.

Dennoch wird an neueren Testentwicklungen gearbeitet, welche Kreativität in einen größeren Zusammenhang als nur kreatives Denken oder kreatives Produzieren stellen, weitere Persönlichkeitsmerkmale testen und damit vielversprechender erscheinen (vgl. Cropley 2000).

### Der kreative Prozess

Eines der ersten und verbreitetsten Modelle des kreativen Prozesses stammt von Wallas (1926). Nach diesem Modell werden beim kreativen Denken vier Phasen durchlaufen: Vorbereitung, Inkubation, Illumination (als zentrales Ereignis) und die Verifikation des Ergebnisses.<sup>10</sup> Im Folgenden wurden verschiedenste weitere Modelle vorgeschlagen, welche sich zwar grundsätzlich ähneln, aber unterschiedliche Schwerpunkte setzen. So argumentiert Lubart (2000), dass das klassische 4-Phasenmodell nicht ausreicht, um die Komplexität des kreativen Prozesses abzubilden. Osborn (1953) identifiziert zu Beginn des kreativen Prozesses eine Orientierungsphase, nach der Vorbereitung eine Phase der Analyse sowie Ideation und das Generieren und Ausbreiten von Alternativen. In der vorletzten Stufe erfolgt die Synthese der Lösung. Isaksen und Treffinger (1985) legen ein Augenmerk bereits auf einen Schritt früher – das Finden eines Ziels oder Prob-

<sup>9</sup> Hierzu enthalten die Auswertungsmaterialien Tabellen, in welchen häufig genannte Antworten verzeichnet sind. Für eine quantitative Testgesamteinschätzung enthält der Test eine Vergleichstabelle, welche entsprechend einer empirisch bestimmten statistischen Einschätzung der Antworten im Verhältnis zum Alter der Versuchsperson eine Art „Kreativitätsquotienten“ ausgibt.

<sup>10</sup> Vgl. auch Kapitel 2.1.3, die inspirationsorientierte Sichtweise.

lems. Koberg und Bagnall (1981) berücksichtigen zusätzlich die Situation, in welcher der kreative Prozess stattfinden soll und machen deutlich, dass ein kreativer Prozess nur möglich ist, wenn sich die Situation als herausfordernd darstellt.

Wallas	Osborn	Isaksen&Treffinger	Koberg&Bagnall
		Ziel finden	
		Fakten finden	
	Orientierung	Problem finden	Akzeptieren der Situation
Vorbereitung	Vorbereitung		
	Analyse	Ideen finden	Analyse
			Definition
	Ideation		Ideation
Inkubation	Inkubation	Lösung finden	
Illumination	Illumination		Auswählen
	Synthese		Implementation
Verifikation	Evaluation		Evaluation
		Akzeptanz finden	

Abbildung 2: Modelle des kreativen Prozesses im Vergleich.

Trotz der vielen verschiedenen Modelle für den kreativen Prozess teilen diese eine Reihe von Gemeinsamkeiten. Alle Modelle berücksichtigen die Vorbereitung und Analyse von Umfeld und Problem. Ein besonderes Augenmerk verdienen jene Modelle, die über den Problemlöseprozess hinausgehen. So berücksichtigen neuere Modelle zunehmend ein systematisches Erzeugen von Lösungsmöglichkeiten und versuchen auch die Phase abzubilden, in welcher ein Ziel oder eine Herausforderung gefunden wird. An ihnen wird deutlich, dass das Finden eines interessanten Ziels, welches sich als gewünschte Herausforderung darstellt und somit auch die Motivation zur Bewältigung anregt, sowie das abschließende Vorstellen der Lösung wesentliche Bestandteile von kreativen Prozessen sind.

Aus Sicht des Informatikunterrichts lassen sich aus diesen Modellen wichtige Schlussfolgerungen für das (kreative) Problemlösen in der Informatik ziehen. Hierauf wird in Kapitel 4 genauer eingegangen.

Amabile (1996) berücksichtigt in ihrem *Componential Model of Creativity* neben Motivation weitere zusätzliche externe Einflüsse des kreativen Prozess (vgl. Abbildung 3). Hierzu zählen die soziale Umgebung, Fertigkeiten im Fachgebiet und kreativitätsrelevante Fertigkeiten. Dieses Modell ist eines der wenigen empirisch überprüften Modelle (vgl. Conti, Coon und Amabile 1996) und stellt damit eine gute Grundlage für die weitere Betrachtung des kreativen Prozesses in dieser Arbeit dar. Aus diesem Grund wird in Kapitel 3 Amabiles Modell bei der Betrachtung kreativer Prozesse in der Softwareentwicklung zugrunde gelegt.

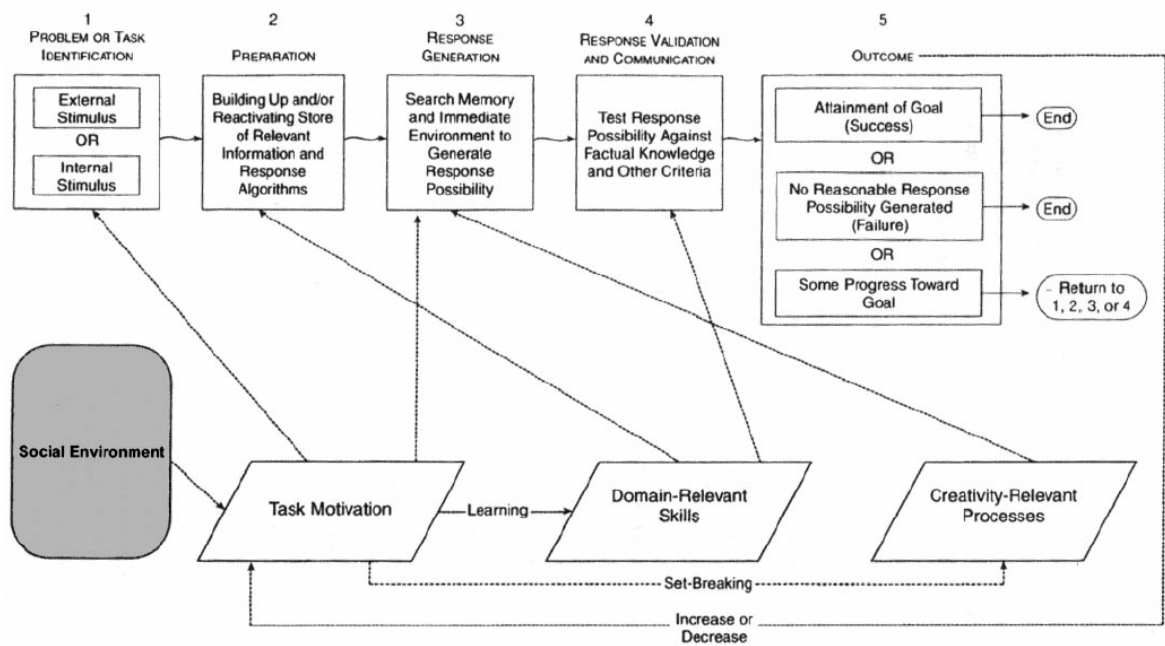


Abbildung 3: Amabiles Componential Model of Creativity (zitiert nach Starko 2001).

### Das kreative Produkt

Verschiedene Bemühungen in der Kreativitätsforschung legen ihren Schwerpunkt auf das Produkt kreativer Prozesse. In der Regel kommt Kreativität erst an einem Produkt zur Geltung, welches damit die Kreativität einer Person Außenstehenden offenbart. *Produkt* bezeichnet hierbei nicht nur Artefakte, sondern auch Ideen, Lösungen, Theorien und Leistungen. Der Vorteil dieser Betrachtungsweise für die Kreativitätsforschung liegt darin, dass sich Produkte, insbesondere wenn sie als Artefakte vorliegen, von mehreren Personen einschätzen und bewerten lassen. Da von einer Person meist nicht nur ein Produkt existiert, können damit auch unterschiedliche kreative Leistungen charakterisiert werden. Zur Bewertung werden Neuheit, Qualität, Nützlichkeit und ggf. Elaboration als Kriterien herangezogen. Dennoch ist es hinsichtlich dieser Kriterien schwierig, eine differenzierte oder gar *quantitative* Wertung von Kreativität vorzunehmen, da das *absolut Kreative* als Bezugsmaß nicht existiert.

Sternberg, Pretz und Kaufman (2002) geben mit dem *Propulsion Model of Creative Contributions* eine Taxonomie an, nach der sich kreative Produkte bzgl. ihres Einflusses auf die Weiterentwicklung eines Fachgebiets einordnen lassen. Im Schulkontext ist dieses Modell allerdings zu vernachlässigen, das hier keine Paradigma verändernden Innovationen zu erwarten sind.

Im TTCT wird die Originalität einer Antwort aufgrund ihrer statistischen Seltenheit, aber nicht als Ausmaß der Ungewöhnlichkeit, bewertet. Ein weiterer Gesichtspunkt, kreative Produkte zu beurteilen, ist der Grad der Elaboration. Allerdings fehlt auch hier in der allgemeinen Betrachtung ein Bezugsmaß.

Die Unterstützung von Originalität innerhalb einer Schulklasse (verschiedene Ideen) und elaborierten Lösungen (Detailgetreue) fördern einen kreativen Unterricht unter diesem Gesichtspunkt.

### Das kreative Umfeld<sup>11</sup>

Nur einige Kreativitätsforscher berücksichtigen den Einfluss des persönlichen oder soziokulturellen Umfelds auf Kreativität. Hier sind zwei Sichtweisen zu unterscheiden. Zum einen die Frage, wann Kreativität erkannt und als solche attribuiert wird und welchen Restriktionen Kreativität

<sup>11</sup> Englisch: Press, auch Umwelt.

aufgrund gesellschaftlicher Vorstellungen und Normen unterliegt (vgl. Sternberg 1988; Sternberg und Lubart 1991; Csikszentmihalyi 1997). Auf diese situationsorientierte Sichtweise wird in Abschnitt 2.1.3 eingegangen. Zum anderen interessiert es vor allem Pädagogen, welche Anforderungen bei der Schaffung eines kreativitätsfördernden Umfelds zu beachten sind und welche Einflussvariablen Kreativität unterstützen bzw. unterdrücken (vgl. Hennessey 2007; Greene 2002; Candy und Edmonds 2000). Da diese Variablen der Unterrichtsumgebung durch Lehrpersonen gestaltbar sind, ist dieser Aspekt Inhalt der Bemühungen, Kreativität zu fördern, und wird in diesem Zusammenhang erneut aufgegriffen werden (vgl. Abschnitt 2.4).

### 2.1.3 Ursprungsorientierte Sichtweisen

Ist Kreativität ein Resultat des Zufalls, ein Produkt harter Arbeit oder durch Umgebungsfaktoren bedingt? Zur Beantwortung dieser Frage lassen sich die Forschungsansätze hinsichtlich des Ursprungs von Kreativität einordnen (vgl. Shneiderman 2000). Alle drei Sichtweisen stellen grundlegende Aspekte von Kreativität dar, die gleichermaßen wichtig sind.

#### Inspirationsorientierte Sichtweise

Die inspirationsorientierte Sichtweise wird gut durch den *Heureka!-Effekt*<sup>12</sup> illustriert, bei welchem, scheinbar „aus heiterem Himmel“, ein großer Durchbruch gelingt, d. h. eine Lösung für ein Problem gefunden wird. Kreativität ist demnach das Ergebnis unbewusster Prozesse. Zu den anekdotischen Belegen dieser Sichtweise wird u. a. die Erfindung des Benzolrings durch Kekulé angeführt. Nach eigenen Berichten sah Kekulé in einem Traum eine kreisförmig verwundene Schlange, die ihn zur entscheidenden Idee führte. Auch Poincaré berichtet von einem solchen Moment bei der Erfindung der Fuchsschen Funktionen. Nach langer, intensiver, aber erfolgloser Arbeit an möglichen Lösungen kam ihm die entscheidende Idee als er – vom Problem abgelenkt – für eine Exkursion einen Bus bestieg. Solche Schilderungen sind es, die dem Begriff der Kreativität für Außenstehende das Mystische und Unerklärliche verleihen. Die beschriebene, plötzlich einsetzende Erkenntnis wird als Illumination bezeichnet. Einig sind sich die meisten „Inspirationisten“, dass Vorbereitung für eine Inspiration essentiell ist. Wallas (1926) stellte die Illumination als dritte Phase des kreativen Prozesses dar, der eine intensive Vorbereitung (Präparation) und eine Phase der Inkubation vorausgehen. Während der Präparation wird ein zu lösendes Problem aktiv im Bewusstsein erfasst und formuliert. In einer Inkubationsphase wird das Problem – vor allem durch das Unterbewusstsein – weiter bearbeitet. Nun kann es passieren, dass aus heiterem Himmel die Erkenntnis zur Problemlösung eintritt, welche anschließend nur noch ausgearbeitet werden muss. Trotz der „magischen“ Komponente dieser Sichtweise sind sich Inspirationisten einig, dass fast ausschließlich die Beschäftigung mit dem Problem selbst für die Lösung verantwortlich ist.

Dieses Modell von Kreativität suggeriert, bei einer Problembearbeitung aus bekannten Strukturen auszubrechen, zwischendurch ganz andere Probleme zu bearbeiten oder bspw. einen Spaziergang zu machen. Kreativitätstechniken wie Brainstorming, freies Assoziieren, divergentes und laterales Denken (De Bono 1970) entstammen dieser Sichtweise.

Reicht es das Gehirn einfach „machen zu lassen“, ist es dann „am kreativsten“ (vgl. Spitzer 2008)? Diese Annahme ist trügerisch. Das temporäre Abwenden von einem Problem mag eine Illumination begünstigen, erzwingen lässt sie sich jedoch nicht. Eine der Grundvoraussetzungen bleibt die intensive Beschäftigung mit dem Problem.

#### Strukturorientierte Sichtweise

Eine geordnete, strategische Herangehensweise an ein Problem kann ebenfalls zu kreativen Leistungen führen. Grundlage ist das intensive Studium vorheriger Arbeiten zu einem Problem sowie methodische Techniken, mit denen mögliche Lösungen systematisch und umfassend

<sup>12</sup> Entsprechend einer Anekdote soll Archimedes nach der zufälligen Entdeckung des Archimedischen Prinzips der Hydrostatik laut „Heureka!“ rufend durch die Stadt gelaufen sein.

erkundet werden. Ein Modell zur systematischen Lösung von Problemen stammt von Polya (1945). Beim Lösen eines mathematischen Problems ist demnach vier Schritten zu folgen:

1. Verstehen des Problems
2. Erstellen eines Lösungsplans
3. Ausführen des Plans
4. Reflektieren des Ergebnisses und des Arbeitsprozesses

Hierbei können verschiedene Heuristiken verwendet werden, wie Analogienbildung, Generalisieren, Variieren und Spezialisieren. Verschiedene Ansätze des Software Engineering können dieser Sichtweise von Kreativität zugeordnet werden. Die Theorie des Erfinderischen Problemlösens (TRIZ) professionalisiert die systematische Herangehensweise an das Finden kreativer Problemlösungen mittels 40 innovativer Prinzipien (vgl. Orloff 2006). Es gibt Versuche, die TRIZ-Prinzipien in der Informatik anzuwenden (Willms, Wentzlaff und Specker 2000).

Dem strukturorientierten Ansatz sind auch Weths Kreativitätsroutinen zuzuordnen, mit denen Schüler im Mathematikunterricht systematisch neuartige mathematische Begriffe schaffen und erkunden (vgl. Weth 1999).

### **Situationsorientierte Sichtweise**

Die situationsorientierte Sichtweise berücksichtigt externe Einflussfaktoren kreativer Prozesse und Leistungen. So stellen der soziale und intellektuelle Kontext Schlüsselfaktoren im kreativen Prozess dar (vgl. Csikszentmihalyi 1997). Kreativität ist damit variablen Standards unterworfen, die durch Bezugspersonen und Gesellschaften bestimmt werden. Leistungen, die heute der Allgemeinheit oder Spezialisten nicht als kreativ erscheinen, können bereits eine Generation später als hochkreativ empfunden werden – ein Schicksal, welches einige Musiker, Maler und Wissenschaftler betraf.<sup>13</sup> Ebenso spielen Abhängigkeiten eine Rolle: So waren und sind sowohl Künstler und Musiker als auch Wissenschaftler gewissen gesellschaftlichen Einschränkungen unterworfen, wenn sie die Gewohnheiten ihres Publikums bzw. gesellschaftliche Normen und Vorstellungen nicht verletzen, sich und ihr Fachgebiet aber gleichzeitig weiterentwickeln wollen.

In der situationsorientierten Sichtweise spielt auch der Einfluss des *unmittelbaren Umfelds* auf die Entwicklung von Kreativität eine Rolle, wie Familie, Freunde oder Lehrer sowie persönliche Herausforderungen, der Drang, sich schöpferisch zu betätigen, oder die Suche nach Anerkennung.

Unter Berücksichtigung dieses Ansatzes für einen kreativen Informatikunterricht ist es wichtig, für ein kreativitätsunterstützendes Umfeld zu sorgen, Offenheit für Neues zu demonstrieren und willkommen zu heißen. Bestehen Informatiklehrer im Unterricht auf (ggf. überholten) Paradigmen, überholten Programmiersprachen oder konventionellen Lösungen, können diese Kreativitätshemmer darstellen.

### **2.1.4 Unterscheidung verschiedener Ausprägungen von Kreativität**

Lange Zeit standen ausschließlich außergewöhnlich kreative Leistungen und Entdeckungen im Zentrum wissenschaftlicher Untersuchungen zur Kreativität. Im Schulkontext ist nicht zu erwarten, dass ein Schüler zu solch außergewöhnlichen Entdeckungen fähig ist (bzw. dass der Schüler oder die Lehrperson in der Lage sind, diese zu erkennen). Aus diesem Grund ist es notwendig, verschiedene Ausprägungen von Kreativität zu unterscheiden.

Boden (1995) beschreibt zwei Bedeutungen von Kreativität. Zum einen *Historische Kreativität* (H-Kreativität), welche Ideen und Leistungen bezeichnet, die noch nie da gewesen und damit geschichtlich neu sind. Hierzu zählen bspw. neue wissenschaftliche Entdeckungen und Erfindun-

---

<sup>13</sup> Van Gogh schuf hunderte Bilder, die während seiner Lebenszeit als merkwürdig und exzentrisch empfunden wurden. Auch die Werke des französischen Malers Édouard Manet stießen zu Lebzeiten auf Ablehnung. Die Werke des Komponisten Gustav Mahler waren bis weit nach seinem Tod umstritten und fanden erst in der zweiten Hälfte des 20. Jahrhunderts volle Anerkennung.

gen, neue Kunststile oder neue Musikrichtungen. Häufig wird *H-Kreativität* gemeint, wenn in der Kreativitätsforschung von Kreativität die Rede ist.

Im Gegensatz dazu steht P-Kreativität<sup>14</sup>, welche auf eine Person bezogene neue Ideen bezeichnet. Eine ähnliche Ausprägung bezeichnet Craft (2001) als „little c creativity“<sup>15</sup> und unterstreicht mit dieser Bezeichnung Kreativität, die in jedem Individuum gefördert werden kann, nicht nur bei wenigen besonders begabten. Diese, auf das Individuum bezogene Kreativität ist wichtig beim Lösen eigener Probleme, im täglichen Leben oder im Beruf. Für die Schule steht damit die P-Kreativität im Zentrum des Interesses. Da P-Kreativität Voraussetzung für H-Kreativität ist, verspricht die Förderung von P-Kreativität, auch h-kreativen Leistungen den Weg zu bereiten.

Shneiderman (2000) unterscheidet Kreativität in drei Stufen. In der eingeschränktesten Bedeutung von Kreativität werden nur Paradigma verändernde Innovationen gezählt. Hierzu zählen große Durchbrüche der Wissenschaft genauso wie die Entwicklung eines neuen Stils in der Kunst oder Musik. Eine lockerere Definition umfasst das, was als „normale Wissenschaft“ bezeichnet werden kann; evolutionäre Beiträge in der Forschung, welche in der Regel durch Anwendung und Überarbeitung bekannter Paradigmen entstehen. Als Beispiele erwähnt Shneiderman selbst Diagnosestellungen von Ärzten und journalistisches Schreiben. Persönliche oder Improptu-Kreativität, als dritte Bedeutung von Kreativität, ist sehr weit gefasst und umfasst spontane und persönliche Aktivitäten, wie eine lebhafte Konversation oder Kindererziehung. Sie hat nur auf wenige Menschen Einfluss und damit nur eine begrenzte Relevanz für die Kreativitätsforschung.

## 2.2 Motivation und Kreativität

Eine im Schulkontext zentrale Frage ist die, was Menschen überhaupt dazu veranlasst, kreativ zu sein.<sup>16</sup> Intrinsische Motivation<sup>17</sup> wird inzwischen als Haupteinflussfaktor für kreative Leistungen angesehen<sup>18</sup> (vgl. Ruscio und Amabile 1996; Runco und Chand 1995). Extrinsische Motivation<sup>19</sup> dagegen kann sich negativ auf Kreativität auswirken. Teilweise wurde sogar davon ausgegangen, dass extrinsische Motivatoren per se kreativitätshemmend wirken. Ursache hierfür war u. a. eine Studie von Lepper, Greene und Nisbett (1973), bei welcher Kinder im Vergleich zur Kontrollgruppe deutlich die Lust am weiteren Malen verloren, wenn ihnen für das Bild eine Belohnung versprochen wurde. Dieser Unterminierungseffekt konnte später ebenfalls bei Personen jeglichen Alters nachgewiesen werden (Hennessey 2007). Mit weiterer Forschung wurde allerdings festgestellt, dass in Abhängigkeit des Gefühls der eigenen Kompetenz und Selbstbestimmtheit (vgl. Deci und Ryan 1985), extrinsische Motivation auch kreativitätsfördernd wirken kann. Hierbei muss zwischen informierenden und kontrollierenden Einwirkungen unterschieden werden. Informierende extrinsische Einwirkungen können sich positiv auf Kreativität auswirken, z. B. durch Präzisierung der Anforderungen, Ermutigung zu besonderen oder außergewöhnlichen Arbeiten und ggf. auch durch konstruktives Feedback<sup>20</sup>. Nach Collins und Amabile (1999) ist Einfluss durch extrinsische informierende Motivatoren für langfristige Kreativität sogar unabdingbar. Kontrollierende

<sup>14</sup> Psychologische Kreativität.

<sup>15</sup> Im Kontrast zu „big C Creativity“.

<sup>16</sup> Ursprünglich wurde der Einfluss von Motivation auf kreative Leistungen im Zusammenhang mit der Kreativitätsdiagnostik betrachtet. Motivationale Einflussfaktoren verfälschten die Testergebnisse in erheblichen Maße, so dass man versuchte, diese kontrollieren zu können. In der weiteren Entwicklung hat sich ein eigener Forschungsschwerpunkt herausgebildet, der sich mit der Rolle von Motivation in kreativen Prozessen auseinandersetzt.

<sup>17</sup> Intrinsische Motivation bezeichnet Motivation „von innen heraus“ aufgrund eines inneren Antriebs und der Beschäftigung selbst, aber ohne externe Belohnung. Intrinsische Motivation führt zu besseren Ergebnissen beim Problemlösen und zu einem besseren Verständnis beim Lernen.

<sup>18</sup> Nach dem *Intrinsic Motivation Principle of Creativity* sind Menschen dann am kreativsten, wenn sie durch ihr Interesse, Vergnügen, Befriedung und die Herausforderung (Challenge) durch eine Aufgabe/Arbeit motiviert sind statt durch externen Druck oder Zwang (vgl. Hennessey und Amabile 1987).

<sup>19</sup> Extrinsische Motivation ist bedingt durch äußere Einflüsse, wie Belohnungen oder Bestrafungen. Sie bewirkt nur bessere Leistungen bei reiner Wiedergabe von auswendig gelerntem Wissen, Präzisionsarbeiten unter Zeitdruck und bei der Erfüllung bekannter Wiederholungsaufgaben.

<sup>20</sup> Gemeint ist nicht „gut-gemeintes“ Feedback, welches „rechtzeitig“ auf Fehler hinweisen will.

Einwirkungen wirken sich dagegen negativ aus: Der Akteur bekommt hierdurch das Gefühl, nicht seine eigenen Vorstellungen umsetzen zu können, sondern von außen gelenkt zu werden. Problematisch ist, dass intrinsische Motivation schnell zerstört werden kann.

Unter bestimmten Umständen sind Synergieeffekte extrinsischer und intrinsischer Motivation möglich. So kann sich bestenfalls bei einer ursprünglich extrinsisch motivierten Aufgabe auch intrinsisches Interesse entwickeln. Bei langfristigen Aufgaben ergänzen sich extrinsische und intrinsische Motivation unter Umständen in unterschiedlichen Phasen gegenseitig.

Die motivationale Orientierung für eine Aufgabe hat direkten Einfluss auf die Kreativität, mit welcher eine Aufgabe gemeistert wird. Dabei hängt die motivationale Orientierung im Wesentlichen von der Umgebung ab, z. B. wie der Lehrer ein Projekt oder eine Aufgabe vorstellt (Hennessey 2007).

Wie kann intrinsische Motivation im Zusammenhang mit Kreativität im Informatikunterricht am ehesten erreicht werden? Durch eine Unterrichtsatmosphäre, in welcher sich Schüler unabhängig, selbstbestimmt und in Kontrolle ihres eigenen Lernprozesses fühlen. Selbständige Informatikprojekte bieten sich dafür an. Extrinsisches Feedback sollte nur informellen, aber nicht kontrollierenden Charakter haben.

Während Motivation eine wichtige Bedingung für Kreativität ist, kann kreatives Tun im Gegenzug auch Motivation hervorrufen: Es macht Spaß, kreativ zu sein. Im kreativen Tun findet der Mensch Selbstbestätigung und Selbstverwirklichung. Csikszentmihalyi prägte für ein intensives Aufgehen in einer kreativen Betätigung den Begriff *Flow*.

### Kreativität erleben: Flow

Ein Zustand, in dem die Arbeit und die aktuelle Tätigkeit gut und erfolgreich *läuft*, wird als *Flow* bezeichnet (Csikszentmihalyi 1990). Hierbei stehen die Anforderungen einer Aufgabe mit den eigenen Fähigkeiten in einem optimalen Verhältnis. Die Arbeit geht leicht von der Hand, dabei herrscht ein Gefühl der Kompetenz, Selbstbestimmung und Bestätigung, das eine Person ganz in der Arbeit aufgehen lässt, so dass auch äußere Einwirkungen nebensächlich werden. Neben intrinsischer Motivation, die eine Voraussetzung für Flow darstellt, identifiziert Csikszentmihalyi (1997) neun Hauptelemente, die für dieses Erlebnis zuständig sind:

1. Jede Phase des Prozesses ist durch klare Ziele gekennzeichnet
2. Man erhält ein unmittelbares Feedback für das eigene Handeln
3. Aufgaben und Fähigkeiten finden sich im Gleichgewicht
4. Handeln und Bewusstsein bilden eine Einheit
5. Ablenkungen werden vom Bewusstsein ausgeschlossen
6. Man hat keine Versagensängste
7. Selbstvergessenheit
8. Das Zeitgefühl ist aufgehoben
9. Die Aktivität wird autotelisch (Ziel seiner selbst)

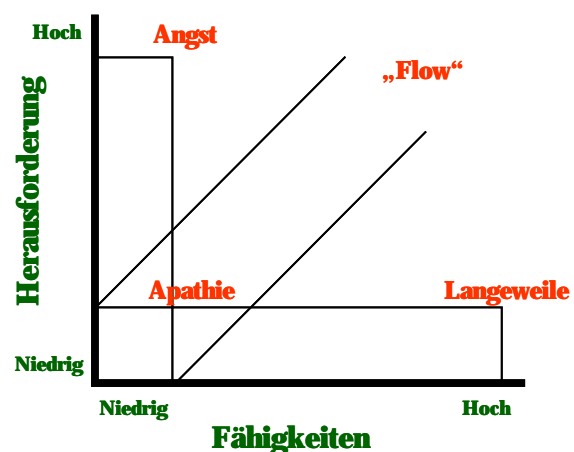


Abbildung 4: Flow als optimale Balance zwischen Anforderungen und Fähigkeiten.

Viele Beispiele von Flowsituationen beinhalten kreative Tätigkeiten; ob Künstler beim Malen oder Wissenschaftler, die von ihrer Arbeit gefesselt sind. Flow kann auch im Informatikunterricht erreicht werden. Jeder Lehrer sollte die Momente kennen, in denen Schüler in sich selbst versunken an einer Aufgabe sitzen und sogar das Pausenklingeln überhören. Ein optimales Verhältnis zwischen Fähigkeiten und Anforderungen sollte bei jeder Aufgabe angestrebt werden. Trifft eine

potentiell herausfordernde und kreative Aufgabe auf unzureichend entwickelte Fertigkeiten ist mit Frustration und Widerstand zu rechnen. Andererseits verhindern zu geringe Herausforderungen, dass sich intrinsisches Interesse an der Lösung der Aufgabe einstellt. Bei einer heterogenen Schülerschaft ist eine Abstimmung auf die unterschiedlichen Fertigkeiten aber nur durch offene Aufgaben und der Betonung von Selbständigkeit zu erreichen.

### 2.3 Begriffsbestimmung

An den verschiedenen Forschungsansätzen wurden die unterschiedlichen Verständnisse von Kreativität deutlich. Diese setzen sich in den Definitionen der Kreativität fort. Im Folgenden werden charakteristische Definitionsversuche dargestellt um einerseits, die Komplexität des Begriffs zu verdeutlichen, und andererseits, die Grundlage für die Herausarbeitung einer im Kontext dieser Arbeit zu verwendenden Definition zu legen.

Sternberg, Pretz und Kaufman (2002) verstehen unter Kreativität

*the ability to produce work that is novel (i.e. original, unexpected), high in quality, and appropriate (i.e. useful, meets task constraints).* (zitiert nach Sternberg 2004, S. 4)

Nicht beantwortet wird in dieser Definition die Frage, was *neu* bedeutet und wie subjektive Aussagen, wie *hohe Qualität* und *Nützlichkeit*, zu bewerten sind.

Auch Ruscio und Amabile (1996) beziehen sich in ihrer Definition auf das Bearbeiten einer (allerdings offenen) Aufgabe:

*We define creativity in our research as a novel and appropriate response to an open-ended task. The task has to be open-ended, rather than straightforward, in order to allow room for creativity or flexibility.*

In dieser Definition beziehen Ruscio und Amabile Umgebungsfaktoren mit ein, die für die Unterrichtsgestaltung wichtige Hinweise implizieren: Nur offene Aufgaben erlauben es den Schülern, sich kreativ auszudrücken.

Die kreative Bearbeitung solch offener Aufgaben erfordert nach Boden (1990) systematisches und strategisches Vorgehen. So charakterisiert sie schöpferisches Denken als

*a matter of using one's computational resources to explore, and sometimes to break out of, familiar conceptual spaces.* (S. 108)

Der Unterschied zwischen einem außergewöhnlich kreativen und einem weniger kreativen Menschen ist demnach „*nicht irgendeine besondere Fähigkeit, sondern größeres Wissen (in der Form praktisch angewandter Kenntnisse) und der Wille, sich dieses anzueignen und es zu benutzen*“ (Boden 1995, S.39).

Feldhusen und Goh (1995) verweisen auf die Komplexität der beeinflussenden Faktoren:

*Within the individual, there is a complex system of cognitive skills, abilities, personality factors and motivations as well as styles, strategies, and metacognitive skills that work together to produce adaptive or creative behaviours. The product or effects of creative activity ultimately define the success or failure of creative efforts.* (S. 234)

Die Problematik des komplexen Kreativitätsbegriffes schlägt sich bei Häcker und Stapf (1998) in einer sehr umfassenden Definition nieder: Kreativität wird hier definiert als

*noch nicht genügend scharf umgrenzten Begriff für ein Gefüge intellektueller und nichtintellektueller (motivationaler, einstellungs- und temperamentsmäßiger) Persönlichkeitszüge, die als Grundlage für produktive, originale, schöpferische Leistungen angesehen werden (im Sinne von Prozessen des Umordnens, Planens, Entwerfens, Erfindens, Entdeckens). Kriterien solcher Leistungen findet man z. B. in der Originalität und Neuartigkeit der Problemlösung, der Offenheit und Flüssigkeit des Produktionsprozesses. Vom stärker individualistischen Begriff des Schöpferischen (etwa der Reformpädagogik) unterscheidet sich der Begriff der Kreativität durch den soziokulturellen Bezugsrahmen (Kriterium: Nützlich für die Gesellschaft), sozial-*

*psychologisch verweist er in seiner Bedeutung als Fähigkeit und Bereitschaft zur Innovation auf die Interdependenz von Kreativität und Gesellschaft. (S. 467)*

Glass (1995) und Couger (1990) verwenden im Bezug auf Kreativität in der Softwareentwicklung eine pragmatische Definition von Kreativität:

*To be creative, a solution must satisfy one or more of the following conditions:*

- 1. The product of the thinking has novelty and value*
- 2. The thinking is unconventional, in the sense that it requires modification or refection of previously accepted ideas.*
- 3. The thinking requires high motivation and persistence, taking place over a considerable span of time (continuously or intermittently) or at high intensity.*
- 4. The problem as initially posed was vague and ill-defined, so that part of the task was to formulate the problem itself. (S. 117)*

Glass und Couger kombinieren damit in ihrer Definition verschiedene Aspekte von Kreativität, stellen Anforderungen nicht nur an das Produkt und den Prozess, sondern auch an die ursprüngliche Aufgabe selbst.

Ansätze einer Definition sind offensichtlich über verschiedene Zugänge möglich: Durch Beschreibung einer kreativen Person, die Charakterisierung eines Prozesses, die Evaluation eines Produkts oder die Betrachtung der Umgebung. Da Ideen im Zentrum jeglicher Betrachtungsweise stehen, soll sich die Definition von Kreativität in dieser Arbeit auf Anforderungen an Ideen beziehen.

In den Definitionen finden sich drei Kernpunkte wieder:

1. Ideen müssen neu, anders, innovativ sein.
2. Sie müssen von hoher Qualität sein.
3. Sie müssen entsprechend einer Aufgabe angebracht und nützlich sein.

Eine Idee oder Reaktion auf ein Problem ist also kreativ wenn sie neu, gut und relevant ist.

Im Schulunterricht kann nicht erwartet werden, historisch kreative Leistungen zu erreichen. Der Fähigkeit, h-kreative Leistungen hervorzubringen, gehen Erfolgserlebnisse und Kompetenzen im Hervorbringen p-kreativer Leistungen voraus. Aufgabe des Unterrichts muss es sein, Schüler zu auf das Individuum bezogenen kreativen Leistungen zu befähigen und zu ermutigen. Eine Leistung (dazu gehören Produkte, Denken und Prozesse) wird entsprechend in dieser Arbeit kreativ genannt, wenn sie durch eine oder mehrere der folgenden Bedingungen charakterisiert ist:

1. Die Leistung ist für das Individuum neu und nützlich bzw. sinnhaft.<sup>21</sup>
2. Die Leistung geht über bekannte Verfahren und Ideen hinaus, z. B. durch Erweiterung, Kombination, Variation oder Spezialisierung bekannten Wissens und bekannter Verfahren.
3. Die Leistung basiert auf hoher Motivation, Ausdauer und Arbeitsintensität.
4. Das ursprüngliche Problem war unklar definiert bzw. wurde durch die Person im Fokus selbst gestellt/aufgeworfen. Die Präzisierung der Aufgabe ist damit Teil des Prozesses.

Zusammenfassend soll eine Leistung als kreativ bezeichnet werden, wenn sie zu persönlich neuen und verwendbaren Ideen, Lösungen oder Erkenntnissen führt.

---

<sup>21</sup> In verschiedenen Zusammenhängen empfiehlt es sich, anstatt von Nützlichkeit von Sinnhaftigkeit zu sprechen, um bspw. Kunst nicht auszuschließen. Andere Definitionen sprechen auch von „angemessen“.

## 2.4 Kreativitätsförderung

Eines der ursprünglichen Ziele der Kreativitätsforschung war die Förderung von Kreativität, welches die Annahme voraussetzt, dass Kreativität gefördert werden kann. Die bis hierher analysierten Modelle von Kreativität bieten verschiedene Ansatzpunkte, um den kreativen Prozess oder dessen Ergebnis zu verbessern. Einen wesentlichen Einfluss hat Motivation (vgl. Abschnitt 2.2), welche u. a. durch ein geschickt gestaltetes Umfeld angeregt werden kann (vgl. Feldhusen und Treffinger 1985; Davis 1991). Sternberg und Lubart (1991) identifizieren darüber hinaus folgende Einflussfaktoren von Kreativität: Intelligenz, Wissen, Denkstil und Persönlichkeit. Allerdings liegen diese, mit Ausnahme des Wissens, nicht im Einflussbereich eines Lehrers.

Amabile (1998) gibt folgende forschungsbasierte Kategorien für die Förderung von Kreativität an, die sich auch durch einen Lehrer im Unterricht berücksichtigen lassen (vgl. auch James, Lederman Gerard und Vagt-Traore 2004):

**Herausforderung:** Die wohl effektivste Methode, Kreativität bei einem Schüler anzuregen, ist die, ihn mit passenden Aufgaben zu betrauen. Wenn eine Aufgabe den Fähigkeiten entspricht und sich für den Schüler als interessant oder als Herausforderung herausstellt, wird intrinsische Motivation entfacht. Es darf dabei nicht zu wenig sein – sonst entsteht Langeweile, ist die Herausforderung zu groß, droht ein Kontrollverlust und Versagensangst. Für einen Lehrer besteht die Aufgabe demnach darin, Aufgaben zu finden, in denen jeder Schüler seine Herausforderung findet und die auf die Fähigkeiten aller Schüler passt, z. B. indem die Aufgabe variabel gestaltbar ist. Viele Forscher empfehlen deshalb, dass Schüler sich ihre Probleme selbst stellen (Problemfinden) oder dass unklar definierte Probleme gestellt werden (vgl. Sternberg und Lubart 1991; Fasko 2000; Treffinger 1980; Feldman 1988).

**Freiheit:** Das Gefühl, eigene Lösungswege einschlagen zu können, vermittelt Selbstkontrolle und Kreativität. Freiheit meint hierbei nicht Beliebigkeit des Ziels. Konkrete Vorstellungen von dem, wie ein Ziel aussehen soll, können Kreativität sogar begünstigen. Intrinsische Motivation entsteht durch eine unabhängige Wahl von Herangehensweise und Aktivitäten. Typische Schul- und Arbeitssituationen bedeuten meist das Gegenteil von Freiheit. Durch eine geschickte Organisation kann aber auch diese Hürde überwunden werden, im Informatikunterricht z. B. durch Projekte.

**Ressourcen:** Ressourcen umfassen vor allem Umgebungsvariablen, die den kreativen Prozess beeinflussen: Zeit, Platz, Materialien und Nachschlagemöglichkeiten sollten vom Lehrer passend gewählt und zur Verfügung gestellt werden.

**Arbeitsgruppen:** Durch eine heterogene Zusammensetzung in Arbeitsgruppen kommen mehr Ideen zum Vorschein. Bei Schülern mit ähnlicher Arbeitsweise finden sich zwar schneller Lösungen und

### 25 Ways to Develop Creativity

#### THE PREREQUISITES

1. Modeling Creativity
2. Building Self-Efficacy

#### BASIC TECHNIQUES

3. Questioning Assumptions
4. Defining and Redefining Problems
5. Encouraging Idea Generation
6. Cross-Fertilizing Ideas

#### TIPS FOR TEACHING

7. Allowing Time for Creative Thinking
8. Instructing and Assessing Creativity
9. Rewarding Creative Ideas and Products

#### AVOID ROADBLOCKS

10. Encouraging Sensible Risks
11. Tolerating Ambiguity
12. Allowing Mistakes
13. Identifying and Surmounting Obstacles

#### ADD COMPLEX TECHNIQUES

14. Teaching Self-Responsibility
15. Promoting Self-Regulation
16. Delaying Gratification

#### USE ROLE MODELS

17. Using Profiles of Creative People
18. Encouraging Creative Collaboration
19. Imagining Other Viewpoints

#### EXPLORE THE ENVIRONMENT

20. Recognizing Environmental Fit
21. Finding Excitement
22. Seeking Stimulating Environments
23. Playing to Strengths

#### THE LONG-TERM PERSPECTIVE

24. Growing Creatively
25. Proselytizing for Creativity<sup>a</sup>

Abbildung 5: 25 ways to develop creativity (Sternberg und Williams 1996, zitiert nach Fasko 2000).

Kompromisse, allerdings werden sie weniger unterschiedliche Lösungswege und Ideen erkunden.

**Ermutigende Betreuung:** Extrinsischer Einfluss kann sich negativ auf kreative Leistungen auswirken. Einwirkungen, die das Gefühl der Selbstkontrolle stärken, üben dagegen einen positiven Einfluss aus. Dieser kann im Unterricht verschiedene Formen annehmen: Ermutigung, angebotenes Feedback, Aufbau von Selbstvertrauen oder Hilfe bei der Strukturierung einer Aufgabe, wenn sich ein Schüler überfordert fühlt. Ebenfalls dazu gehören die Förderung von Neugier, Risikobereitschaft, Selbstvertrauen und exploratives Lernen.

**Institutionsweite Unterstützung:** Nach Möglichkeit sollte Kreativität durch die ganze Schule oder gar das ganze Schulsystem als Ziel verfolgt werden. Damit verliert kreatives Arbeiten das Ungewöhnliche in einem sonst restriktiven Kontext und kann zu einer Selbstverständlichkeit werden. Sternberg und Williams (1996) geben darüber hinaus 25 Tipps zur Anregung von Kreativität (vgl. Abbildung 5).

## 2.5 Kreativität in der Lernwissenschaft

Auch in der Lernwissenschaft erlangt Kreativität eine zunehmend größere Bedeutung. Das wohl am häufigsten zur Vorbereitung und Evaluierung von Unterricht eingesetzte Modell ist die Lernziel-Taxonomie von Bloom (1974). Auf höchster kognitiver Stufe wird dort die Evaluation über das eigene Handeln beschrieben. Anderson und Krathwohl (2001) unterzogen Blooms Taxonomie einer Überarbeitung nach lernpsychologischen Gesichtspunkten und empirischen Ergebnissen. Diese Ergebnisse implizieren, dass nicht mehr die Evaluation an der komplexesten kognitiven Stufe anzusiedeln ist, sondern das *Kreieren*. Anderson und Krathwohl begründen diese Neuordnung mit der höheren Komplexität induktiver Prozesse im Gegensatz zu deduktivem Vorgehen. Während beim Evaluieren deduktiv ein Ganzes in seine Teile zerlegt werden muss und diese bzgl. gewisser Kriterien bewertet werden müssen, beinhalten das Hervorbringen und Gestalten ein induktives Finden der richtigen und zusammenpassenden Teile sowie das Bewerten und bestmögliche Zusammensetzen dieser vor dem Hintergrund der zu erfüllenden Anforderungen. Damit stellt sich Induktion als komplexer dar als Deduktion. Verschiedene alternative Klassifikationssysteme teilen diese Auffassung (vgl. Anderson und Krathwohl 2001). Insbesondere die Klassifikationen von Ausubel (1974) und Haladyna (1997) fassen ebenfalls Kreativität als höchstes kognitives Lernziel auf.

Moderne Lerntheorien scheinen sich allerdings nur schwer im Schulalltag durchsetzen zu können. Fachunterricht folgt zu häufig noch dem Prinzip des Nürnberger Trichters. Bereits Piaget wies darauf hin, dass Lernen ein aktiver Prozess ist, in welchem der Lernende eine eigene Repräsentation der Welt vor dem Hintergrund seiner Erfahrungen konstruiert. Die Theorie des situierten Lernens empfiehlt den sozialen und physikalischen Lernkontext zu berücksichtigen. Das darf allerdings nicht dazu führen, dass Unterrichtsinhalte in haarsträubende Kontexte eingebettet werden, die für den einzelnen Schüler belanglos sind. Lernen ist ein Prozess, in dem der Lernende in ständiger wechselseitiger Beziehung mit seiner Umgebung steht. Verschiedene Ansätze, z. B. das Entdeckende Lernen, unterstreichen, dass vor allem dann mit einer positiven Einstellung und am besten gelernt wird, wenn eigene Fragen und Probleme im Lernprozess bewältigt werden. Problematisch dagegen ist es, wenn sich das Entdecken alleinig auf das Entdecken curricular vorgegebener Routinen beschränkt.

Papert greift in seiner Lerntheorie des Konstruktivismus das Prinzip des Lernens als Konstruktionsprozess auf, fügt aber die Idee hinzu, dass Lernen durch das eigenhändige Konstruieren eines Artefakts<sup>22</sup> unterstützt wird (Papert und Harel 1991). Im Gegensatz zu Piaget hält Papert es für wichtig, dass Lernende direkt in Situationen eintauchen, statt sie von außen zu betrachten. Hierzu zählt auch die Verbundenheit mit dem Lerngegenstand: Der Schlüssel zum Lernen liegt im Einswerden mit dem betrachteten Phänomen, was wiederum zu bedeutungsvollem Lernen führt.

---

<sup>22</sup> Bspw. die Konstruktion eines Hauses mit Bausteinen, die Entwicklung einer Theorie oder das Programmieren eines Computers.

Vorteil des Schaffens eines realen Produkts ist die Möglichkeit, dieses auszuprobieren, zu zeigen, zu diskutieren und zu bewundern (Papert 1993). Konstruieren des Wissens bleibt damit nicht nur eine versteckte Angelegenheit des Geistes, sondern wird durch „greifbare“ Artefakte unterstützt und nachvollziehbar. Artefakte schaffen Verständnis durch deren Interpretation; Verständnis kann repräsentiert werden in der Konstruktion eines Artefakts. Durch dieses Wechselspiel von Repräsentation und Interpretation entwickeln sich Artefakte und Verständnis evolutionär, was die Konstruktion des Wissens darstellt. Konkretes Denken ist nach Papert demnach keine Entwicklungsstufe, die bewältigt werden muss, sondern eine andere Repräsentation des Denkens, die ihre Berechtigung und Anwendungen hat. Auch wenn das Schaffen von Artefakten und Verständnis im Sinne des Konstruktivismus von Papert nicht explizit als Kreativität bezeichnet wird, so stellt es doch in dem herausgearbeiteten Verständnis von Kreativität zumindest eine Vorstufe zur Bewältigung kreativer Aufgaben dar und kann für die Gestaltung eines kreativen Informatikunterrichts wichtige Anhaltspunkte liefern. Kern der Informatik ist die Schaffung künstlicher Artefakte. Computertechnologie stellt eine fruchtbare Umgebung dar, um Ideen zu generieren und auszutesten (Papert 2000), da Artefakte auch Konstruktionen der künstlichen Welt mit einschließen. Die Konstruktion von Artefakten und ein Lernverständnis im Sinne des Konstruktivismus bieten sich damit in der Informatik geradezu an und können ein bedeutungsvolles und kreatives Lernen ermöglichen.

## 2.6 Schule und Kreativität

Grundsätzlich wird in Schulen Kreativität zu wenig berücksichtigt. Studien zeigen, dass Kreativität bei Schulkindern mit jedem Schuljahr kontinuierlich sinkt (Lepper, Corpus und Iyengar 2005). Ein Grund hierfür könnte sein, dass Kreativität kaum von Intelligenz unterschieden wird oder dass Lehrer nicht wissen, wie sie Kreativität unterrichten sollen. Zusätzlich ist es grundsätzlich für den Lehrer schwieriger, kreative Lösungen – und damit viele verschiedene Lösungen – im Unterricht zu berücksichtigen und zu bewerten. So kritisiert Hentig (2000) die Lehrmethoden in der Schule gar als kreativitätsfeindlich und bezweifelt, dass die Förderung von Kreativität in der Schule tatsächlich versucht wird:

*Es läßt mich vermuten, daß uns an wirklicher Kreativität gar nicht so viel liegt. Sie bringt ja so viel eigene Probleme, Schwierigkeiten, Unregelmäßigkeiten mit sich! (S. 70)*

Kreative Schüler sind schwieriger im Unterricht zu handhaben, passen sich ungern an und stören auch mal den „geregelten Unterrichtsablauf“. Nicht selten neigen Lehrer gar dazu, kreatives Verhalten negativ zu beurteilen (Westby und Dawson 1995; Sternberg und Lubart 1991). Stattdessen dominieren in Schulen konvergente Aufgaben. Während Kreativität intrinsischer Motivation entspringt, findet sich in der Schule ein Bewertungssystem, welches auf extrinsischer Motivation basiert. Schüler neigen dann zu einem Problemvermeideverhalten, d. h. sie wählen sichere, ausgetretene Wege, die risikoarm und erfolgversprechend sind, aber für Kreativität nicht viel Raum lassen (vgl. Sternberg und Lubart 1991). Frühe Evaluation und Perfektionsstreben sind in diesem Kontext als Kreativitätshemmer bekannt (vgl. Osborn 1953).

Wie empirische Studien zeigen, bemängeln auch heutige Berufstätige, dass in ihrer Schulzeit wichtige Sozialkompetenzen, insbesondere Kreativität, zu wenig gefördert wurden (vgl. Feller 1996). Aus der Wirtschaft wird ebenfalls mangelnde Kreativität bei Schulabgängern angemahnt (vgl. Geser 1999). In der Schule scheint stattdessen ein besonderer Wert auf Gedächtnisleistungen und analytische Fähigkeiten gelegt zu werden, wobei doch kreative und praktische Fähigkeiten genauso wichtig für den Erfolg im Leben sind, vielleicht sogar wichtiger (vgl. Sternberg 2003). Ein Unterrichtsstil, der Kreativität anregt und belohnt, kann darüber hinaus sogar schulische Leistungen steigern (Sternberg 2003). Auch unter den schultypischen formalen Voraussetzungen sollte die Förderung intrinsischer Motivation und Kreativität möglich sein.

## 2.7 Kreativität in naturwissenschaftlichen Unterrichtsfächern

Um die Bedeutung von Kreativität im Informatikunterricht im Vergleich mit anderen Unterrichtsfächern erfassen und ggf. auf deren Erfahrungen zurückgreifen zu können, werden in diesem Abschnitt Ansätze zur Berücksichtigung von Kreativität in anderen naturwissenschaftlichen Fächern untersucht.

### 2.7.1 Mathematik

Erste Untersuchungen von Kreativität in der Mathematik stammen von Hadamard (1949), welcher den kreativen Prozess vor dem Hintergrund des Kreativitätsmodells von Wallas<sup>23</sup> analysiert. Für den Mathematikunterricht ist dieses Modell allerdings nicht zu verwenden, da sich eine Illumination nicht erzwingen lässt. Weth (1999) hat sich im Rahmen einer Habilitationsschrift mit den Möglichkeiten beschäftigt, die Mathematik bietet, Kreativität einzusetzen und zu fördern. Er stellt fest, dass Mathematik sich bisher „auf das Beherrschen von Wissen und die Einübung von Verfahren“ beschränkt (Weth 1999, S.34). Insbesondere in Konkurrenz mit anderen Unterrichtsfächern schneidet Mathematikunterricht damit bisher schlecht ab. Mathematikunterricht ist noch nicht kreativitätsfördernd, kann es nach Weths Meinung aber sein.

Weth verdeutlicht, dass Problemlösen nicht mit Kreativität gleichzusetzen ist. In der Mathematik bieten Knochecken zwar Anknüpfungspunkte für kreatives Tun, allerdings handelt es sich dabei immer um konvergente Aufgaben. Im Unterrichtskontext können sowohl Zeitdruck als auch die mangelnde Aussicht, das einzig richtige Ergebnis zu finden, Schüler demotivieren. Des Weiteren stellt Weth den positiven Einfluss von Werkzeugen, insbesondere Computern, auf kreatives Verhalten heraus.

Vollrath (1987) postulierte „Begriffsbildung als schöpferisches Tun im Mathematikunterricht“. Mathematik müsse sich vom ausschließlichen Problemlösen freimachen und stattdessen den Schülern die Möglichkeit geben, ein eigenes Stück Mathematik zu „erfinden“.<sup>24</sup> Dies gelingt besonders gut im selbständigen Bilden von Begriffen. Weth operationalisiert Vollraths Vorschläge, indem er „Kreativitätsroutinen“ (Handlungsstrategien) angibt, mit deren Hilfe Schüler geführt eigene Begriffe erfinden und erforschen können. Zu den Begriffschaffungsroutinen gehören:

- Spezialisieren
- Verallgemeinern
- Modifizieren
- Kombinieren
- Analogisieren
- Iterieren
- gezieltes und freies Experimentieren

*Gehe von einer gegebenen Definition aus und ändere eine Bedingung ab.*

*Als Ausgangsbegriff wählen wir wieder die Definition des Quadrats:*

*Ein Quadrat ist ein Viereck mit vier gleichlangen Seiten und einem rechten Winkel.*

*Begriffsbildung: Ein Spezialfünfeck ist ein Fünfeck mit vier gleichlangen Seiten und einem rechten Winkel.*

Abbildung 6: Beispiel für die Kreativitätsroutine „Modifizieren“ (Weth 1999, S. 70).

<sup>23</sup> Vgl. Abschnitt 2.1.2.

<sup>24</sup> Auch in der Informatikdidaktik/im Informatikunterricht besteht die Tendenz, alleinig das Problemlösen in den Vordergrund zu stellen. Das vielfältige Potential, was die Informatik bietet, zu verschiedenen Lösungen, Ideen und Produkten zu kommen, wird durch konvergente Aufgabenstellungen häufig unterdrückt. Genauer wird auf diesen Punkt im Kapitel 4, Abschnitt 4.4 eingegangen.

Das Bilden eines mathematischen Begriffs umfasst

1. die Schaffung/Konstruktion des Begriffes und
2. die mathematische Durchdringung des Begriffs, für welche „Entdeckungs- und Erforschungsroutinen“ angegeben werden.

*Eine natürliche Zahl heißt in der Mathematik „vollkommen“, wenn sie gleich der Summe ihrer echten Teiler ist; zum Beispiel  $6 = 1+2+3$ . Erfinde und untersuche „arme“ (oder „reiche“) Zahlen.*

Abbildung 7: Beispiel: Erfinden mathematischer Begriffe (Weth 2001, S. 65).

Weth stellt eine prototypische Unterrichtssequenz vor, welche in der Lehrerbildung erprobt wurde. Die geschilderten Ergebnisse sind überwiegend positiv. Untersuchungen über den Einsatz seiner Konzepte in der Unterrichtspraxis sind allerdings nicht dokumentiert.

Das Verwenden von „Routinen“ und „Vorgehensweisen“ scheint auf den ersten Blick im Gegensatz zu Kreativität zu stehen. Tatsächlich spielen aber im kreativen Prozess gerade Heuristiken, Erfahrung und Fach- und Methodenkenntnis eine wichtige Rolle.

Weitere Vorschläge für einen kreativen Mathematikunterricht macht Winter (1999), welcher ein stärkeres Zusammenspiel von Gestalt und Zahl vorschlägt. Hiermit könne die heuristische Kompetenz der Schüler gestärkt und damit kreatives Denken angeregt werden.

*Das Aufdecken von Wechselbeziehungen zwischen Gestalt und Zahl ist eine der kreativsten Formen des Mathematiklernens, weil dabei ästhetische Momente wirksam sind. (S. 19)*

Winter stellt hiermit allerdings eine spezielle, ästhetisch-künstlerische Sichtweise von Kreativität in den Mittelpunkt.

Auch Bruder (2001) geht der Frage nach, wie Mathematikunterricht (in der Oberstufe) gestaltet sein muss, damit ein kreativer Umgang mit Mathematik ermöglicht, von den Lernenden gewollt und bewältigt werden kann. Ihrer Auffassung nach äußert sich Kreativität im Mathematikunterricht

- *im Aufwerfen von Fragen und infrage Stellen von Sachverhalten oder Darstellungen zu mathematischen Themen;*
- *im Entdecken beziehungsweise Erfinden (subjektiv neuer) mathematischer Zusammenhänge, Problemlösemethoden oder Anwendungen von Mathematik – insbesondere auch im Erkennen mathematischer Beschreibungsmöglichkeiten von Alltagszusammenhängen;*
- *im Finden und Ausprobieren eines subjektiv neuen oder nicht konventionellen Lösungsweges zu einer gegebenen Aufgabe;*
- *im Variieren und eigenen Erfinden von Aufgaben;*
- *beim (originellen) Präsentieren, Begründen und Werten von Arbeitsergebnissen. (S. 46)*

Diese Tätigkeiten finden sich allesamt auch im Informatikunterricht wieder. Das Potential für Kreativität im Informatikunterricht geht meiner Ansicht nach allerdings weit darüber hinaus. Während kreatives Arbeiten im Mathematikunterricht den Beschreibungen nach oft nur innerfachlich motiviert betrieben wird (z. B. beim Erfinden eigener Definitionen), stehen im Informatikunterricht viel häufiger anwendungsrelevante und nutzbare Produkte am Ende des kreativen Prozesses.

### 2.7.2 Physik

Jablko (1994) versteht Kreativität im Physikunterricht als das Erklären eines Phänomens oder Sachverhalts. Das „Beschreiten neuer Wege“ durch die Schüler will er realisiert sehen indem die Schüler „die Gelegenheit haben, eigene Gedanken und Ideen auch einzubringen“ (S.328). Jablko hält dafür einen problemorientierten Unterricht für geeignet, wobei er Erklärungsprobleme präferiert.

*Beispiel: Den Schülern wird erläutert und demonstriert, dass in einen Standzylinder ein brennbares Gas [...] eingefüllt wurde und sich in anderen Luft befindet. Dann werden die beiden Gefäße übereinander gestellt, und es wird die Trennscheibe vorsichtig entfernt.*

*Führen wir sofort danach in beiden Standzylindern eine Flammprobe durch, so findet – wie wohl von allen erwartet – in einem keine Reaktion, im anderen eine Verpuffung statt. Nun wird ein gleiches Gefäßpaar demonstriert, nur erfolgte hier die Füllung vor ca. einer Stunde. Zum Erstaunen der Schüler ergibt hier die Flammprobe für beide Gefäße eine Verpuffung, in beiden Standzylinder befanden sich also wenigstens Teile eines brennbaren Gases. In einem nächsten Schritt ist das Problembewusstsein durch die Herausarbeitung der Problemformulierung zu verstärken. Danach erfolgt das Erklären. Und nun wird den Schüler offenbart, was der Erkenntnisstand der Physik besagt, oder die Situation wird mit komplexer Sicht auf die Persönlichkeitsentwicklung der Schüler genutzt. (S. 329)*

Stellt das (durch den Lehrer) gestellte Problem auch verschiedene Lösungsmöglichkeiten in Betracht, so gibt es bei Erklärungsproblemen meist nur eine richtige Lösung, die durch geschicktes Schlussfolgern und kreatives Experimentieren gefunden werden muss. Insbesondere weniger interessierte Schüler werden bei solchen Aufgaben (die irgendwann vom Lehrer „aufgelöst werden“) wenig Motivation zeigen, sich kreativ mit der Lösungssuche zu beschäftigen.

Mackin (1996) beschreibt einen Ansatz, Konzepte der Physik in Projekten so zu unterrichten, dass sie für die Schüler interessant und bedeutungsvoll sind. Hierbei regt sie Kreativität an, indem sie die Schüler physikalische Phänomene ihrer Lebenswelt diskutieren und eigene Experimente aufstellen lässt. Ziel der Projekte ist es, dass die Schüler für sich selbst, aber auch für andere, Physik erklären können, also ihr Wissen teilen. Der Abschluss der Projekte erfolgt mittels einer Präsentation, die verschiedenartig gestaltet sein kann (Modell, Video, Audio, Song, Artikel, Poster, Report etc.). Mackin berichtet, dass die Schüler nicht nur mit den aufgeworfenen Fragestellungen und Problemen, sondern auch in den Bearbeitungen, Lösungen und Präsentationen, viel Kreativität einbrachten.

### 2.7.3 Chemie

Gärtner (1997) fasst zusammen, dass nur wenige Chemie-Lehrpläne Kreativität explizit als Lernziel ausweisen. Er begründet dies damit, dass das „über empirische Befunde abgesicherte Wissen über Stoffe und stoffliche Veränderungen, [und] zum anderen der streng, durch Experimente geprägte Erkenntnisprozess“ für innovative und kreative Prozesse offenbar nur wenig Raum lassen (S. 12). Dennoch fordert er, auch kreative Zielsetzungen zu verfolgen und formuliert Rahmenbedingungen und methodische Aspekte, die hierfür einzuhalten sind.

*So kann der Nachvollzug bereits bekannter Sachverhalte Möglichkeiten kreativen Schaffens eröffnen, sofern auf Versuchs- und Bauanleitungen ganz oder teilweise verzichtet wird und die Schülerinnen und Schüler somit Gelegenheit erhalten, auch eigene Wege zu gehen. Gegenstand eines solchen nachmachenden Unterrichts kann zum Beispiel die Anfertigung von Low-Cost-Equipment sein, das im eigenen Unterricht verwandt werden soll [...] oder auch anderen Zwecken dienlich ist. (S. 13)*

Neben diesem „nachmachenden Unterricht“ schlägt Gärtner auch eine „nacherfindende“ Vorgehensweise für den Chemieunterricht vor, bei welcher die Schüler, ohne sich an einem Vorbild zu orientieren, in eine Forscherrolle schlüpfen und versuchen, „Neues“ zu entdecken.

Als Drittes schlägt Gärtner für einen kreativen Chemieunterricht einen „forschend-entwickelnden Unterricht“ (vgl. Schmidkunz, Lindemann und Schmidkunz 1976) als Unterrichtsmethode vor.

Vorteile dieser Methodik sieht Gärtner in einem tatsächlichen Forschungsprozess, der sich auf jeden empirisch fassbaren Vorgang in der Lebenswelt der Schüler anwenden lässt. Damit könnten auch triviale Alltagsphänomene überraschend neu betrachtet werden. „Trial-and-Error-Phasen“ werden dabei explizit in die Unterrichtsgestaltung im Rahmen von „Problembausteinen“ mit einbezogen.

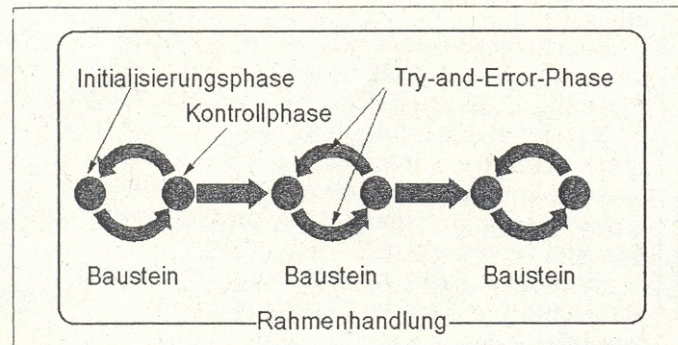


Abbildung 8: Phasen der Problembausteine (zitiert nach Gärtner 1997, S. 13).

Als geeigneten Rahmen für kreativen Unterricht empfiehlt Gärtner Projekte und Mini-Projekte, deren kreativitätsfördernde Wirkung er als erwiesen sieht.

Auch Krämer (2007) berichtet von Problemen mit einem unkreativen Chemieunterricht, der u. a. von vielen Formeln und Gleichungen als auch einem späten Beginn des Fachs im Unterrichtskanon herrühre. Als Lösung empfiehlt sie einen kreativen Chemieunterricht, der an Alltagserfahrungen (Alltagschemie) angeknüpft, experimentiert und neue Erkenntnisse präsentiert. Inwieweit in den Projekten allerdings fachliche Kreativität gefördert wird, ist nicht dokumentiert. Dennoch berichtet Krämer von einer höheren Beliebtheit des Faches, einem besseren Verständnis und höherer Motivation der Schüler aufgrund kreativer Elemente im Chemieunterricht.

#### 2.7.4 Science Education

Einige internationale Forscher haben sich mit der Rolle und Bedeutung von Kreativität in der *Science Education* auseinandergesetzt. Die ausführliche Darstellung dieser Ansätze ist an dieser Stelle aus Platzgründen nicht möglich. Ausführungen hierzu sind z. B. bei Moravcsik (1981) und überblicksweise bei Kind und Kind (2007) zu finden.

### 2.8 Zusammenfassung und Bewertung

Im Folgenden werden Faktoren, die sich in der Diskussion von Kreativität als wesentlich gezeigt haben und mögliche Implikationen für den Informatikunterricht zusammengefasst.

Bei der wissenschaftlichen Untersuchung von Kreativität lassen sich vier Haupteinflussfaktoren identifizieren: Die Betrachtung des Individuums, also **menschlicher** Faktoren (Persönlichkeit, Motivation), die Betrachtung des **Umfelds** und der Einflussfaktoren von Kreativität, die Betrachtung kreativer **Prozesse** sowie die Betrachtung kreativer **Produkte**. Alle Faktoren können sowohl unter dem Aspekt der Bewertung als auch der Förderung von Kreativität betrachtet werden. Nicht nur vor dem Hintergrund der Messbarkeitsproblematik und potentiellen Störfaktoren bei der Diagnostik individueller Kreativität, sondern auch vor dem Hintergrund der allgemeinen Förderung aller Schüler, soll der Aspekt der Bewertung von Kreativität im Informatikunterricht für diese Arbeit ausgeschlossen werden.<sup>25</sup> In jedem Schüler steckt kreatives Potential.<sup>26</sup> Aufgabe des

<sup>25</sup> Das bedeutet auch, dem Versuch des Bewertens und Klassifizierens von Kreativität („Wer ist am kreativsten?“ oder „Ist dies kreativ?“) zu widerstehen.

<sup>26</sup> Wie verschiedene Ansätze verdeutlichen, kann Kreativität auch als inhärentes Bedürfnis des Menschen verstanden werden, z. B. um sich selbst zu verwirklichen (vgl. Flaherty 2005; Conti und Amabile 1999).

Informatikunterrichts muss es sein, dieses Potential herauszufordern, zu fördern und für die Unterrichtszwecke (nämlich Motivation, Interesse, Lernen) zu nutzen. Letztendlich scheint vor dem Aspekt der motivierenden Funktion von Kreativität das Gefühl kreativen Tuns wichtiger zu sein, als tatsächliche große Entdeckungen zu erreichen. Nur so entwickelt sich die Fähigkeit, sich an tatsächliche Entwicklungen (z. B. mittels evolutionärer Kreativität) heranzuwagen.

Intrinsische Motivation ist das Kernelement für Kreativität. Kreativitätsorientiert zu unterrichten heißt damit auch, aus den üblichen Bewertungsstrukturen der Schule auszubrechen, da übliche Bewertungsverfahren intrinsische Motivation unterlaufen. Stattdessen müssen andere Formen der Belohnung gefunden werden, wie z. B. durch bedeutungsvolle Aufgaben und/oder durch Feedback, welches die Schüler durch die Präsentation ihrer kreativen Produkte erhalten. Aus Sicht der Unterrichtsplanung stellt sich die Gestaltung einer kreativen Unterrichtsumgebung als wesentlich dar, wobei neben offenen Aufgaben, dem Einräumen von Freiheiten, Unterstützung und Ermutigung auch dem Stellen einer Herausforderung und der Ermöglichung eines Gefühls von Kompetenz und Selbstbestimmung wichtige Rollen zukommen.

Die Bemühungen in den naturwissenschaftlichen Unterrichtsfächern, Kreativität in den naturwissenschaftlichen Unterricht zu integrieren, teilen ein subjektives, schülerorientiertes Verständnis von Kreativität, in welchem subjektiv neue Entdeckungen, Nacherfinden und Wiederentdecken bekannter Zusammenhänge im Vordergrund stehen. Es wird für wichtig erachtet, in den jeweiligen Fächern den Schüler Gelegenheiten zu geben, ihre Kreativität einzubringen, unabhängig von der Altersstufe. Kreativität ist für jeden Schüler realisierbar. Auf fachspezifische Weise werden dabei unterschiedliche Facetten von kreativen Ausdrucksmöglichkeiten deutlich. Gerade in Chemie und Physik wird weniger etwas „Schöpferisches“ unter einer kreativen Schülertätigkeit verstanden, sondern das Entdecken neuer Wege und Erkenntnisse im Experimentieren. Gemeinsam ist allen dargestellten naturwissenschaftlichen Unterrichtsfächern, dass die Projektmethode als geeignetes Mittel zur Kreativitätsentfaltung eingesetzt wird.

In der Mathematik wird vor allem eine strukturorientierte Sichtweise von Kreativität im Unterricht angewandt; mit Hilfe von Strategien, Werkzeugen und Methoden werden systematisch neue Ideen produziert. Diese Herangehensweise ist auch im Informatikunterricht vorstellbar, da in der Informatik das systematische Entwickeln von Lösungen einen zentralen Stellenwert besitzt.

Insgesamt existieren nur wenige Publikationen zum Thema Kreativität in den einzelnen naturwissenschaftlichen Fachdidaktiken. Es ist erfreulich, dass in verschiedenen naturwissenschaftlichen Fächern Anstrengungen unternommen werden, Kreativität einzusetzen und zu fördern. Mitunter scheint dieser Versuch allerdings aufgesetzt, unnatürlich oder entspricht nur einem nacherfindenden oder passiv-erklärenden Verständnis von Kreativität. Durch ihren aktiven gestaltenden Kern hat die Informatik – und damit der Informatikunterricht – durchaus bessere Möglichkeiten, Kreativität zu fördern.

Häufig wird Kreativität eingesetzt, um den Unterricht interessanter zu gestalten und die Motivation der Schüler zu steigern. Zusammenfassend lässt sich feststellen, dass Kreativität im Unterricht die Chancen bietet, Schülermotivation zu steigern, Aufmerksamkeit, Neugier und Konzentration zu verbessern und dadurch letztendlich einen besseren Lernerfolg zu erzielen.

### 3 Kreativität in der Informatik<sup>27</sup>

*Is programming an art, a science, a tool, or a trade?  
All those things. (Bob Frankston)<sup>28</sup>*

*In computer science, the limit is your imagination!  
The more creative you are, the further you are going to get. (Gustavo Alonso)<sup>29</sup>*

Die Frage nach Kreativität in der Informatik löst häufig Kontroversen aus. Zu unterschiedlich sind die persönlichen Auffassungen von dem, was Kreativität für den Einzelnen bedeutet, und zu unterschiedlich ist das, was der Einzelne unter Informatik versteht. In diesem Abschnitt will ich den Versuch wagen, verschiedenste Betrachtungsmöglichkeiten von Kreativität in der Informatik zusammenzutragen. Wie unterschiedlich der Begriff *Kreativität* gebraucht werden kann und welche unterschiedlichen Aspekte man in diesem Spannungsfeld betrachten kann, wurde bereits im Kapitel 2 verdeutlicht.

Auch eine Festlegung, was denn unter „Informatik“ zu verstehen sei, ist durchaus nicht einfach. Einschlägige Publikationen (z. B. Rechenberg (2000): *Was ist Informatik*, Desel (2001): *Das ist Informatik* oder das *GI-Positionspapier* „Was ist Informatik“ (GI 2006)) charakterisieren die Wissenschaft Informatik häufig anhand ihrer Teilgebiete: z. B. Technische, Praktische, Theoretische und Angewandte Informatik. Dieses Bild der Informatik vermag allerdings nicht sonderlich, Schüler und Studenten für die Informatik zu begeistern (vgl. auch Maass und Wiesner 2006). Diese recht konservative Betrachtungsweise verschließt die Augen gegenüber den vielfältigen Lebensbereichen, in denen gerade die kreative Beschäftigung mit Informatik Einzug gehalten hat und deren Bedeutung für den Einzelnen. Einen zentralen Bestandteil nimmt in der Informatik die Softwareentwicklung ein, an welcher sich kreative Prozesse besonders manifestieren. Auch Schwill (1993) schreibt, dass sich Denk- und Arbeitsweisen der Informatik vor allem am Softwareentwicklungsprozess konkretisieren. Vor diesem Hintergrund werde ich mich im Folgenden vor allem auf diese Tätigkeiten der Informatik beziehen, die im Zusammenhang mit der Erstellung von Software stehen. Tatsächlich werden Informatik und Programmieren in der Betrachtung von Kreativität häufig sogar fast synonym benutzt (z. B. in Leach und Ayers 2005). Dies bedeutet nicht, dass in anderen Bereichen der Informatik Kreativität nicht existiert oder notwendig ist – im Gegenteil, Kreativität ist nach meiner Auffassung ein wichtiger Teil jeder Wissenschaft. In der Informatik wird sie aber besonders an der Entwicklung von Software bzw. am Programmieren deutlich. Für die Schulinformatik ist dieser Aspekt von Bedeutung, da Schüler hier selbst relativ leicht innerhalb der Informatik kreativ tätig werden können.

Unweigerlich betrachtet werden müssen in diesem Zusammenhang auch Aspekte der Kunst im Verständnis der Informatik, deren Anwendung und Einfluss. Des Weiteren werde ich auf die Bedeutung kreativen Tuns in der Informatik für die Akteure, die Rolle von Informatiksystemen für Kreativität sowie auf artifizielle Kreativität eingehen. Die Betrachtung erfolgt jeweils aus dem Blickwinkel der entsprechenden Sichtweise, wobei die Perspektiven weder als vollständig noch als disjunkt zu verstehen sind.

#### 3.1 Fach-Perspektive

In Aufsätzen und Artikeln lassen sich verschiedenste Aussagen zur Kreativität in der Informatik finden. Oftmals sind es Softwareentwickler, die über ihre Tätigkeit sprechen; gelegentlich wird

<sup>27</sup> Teile dieses Abschnitts wurden bereits zum Teil sinngemäß und zum Teil wortwörtlich in (Romeike 2007c) und (Romeike 2008d) veröffentlicht.

<sup>28</sup> Zitiert nach Lammers (1986).

<sup>29</sup> Zitiert nach (ETH-Zürich 2003).

Kreativität gefordert (z. B. Smith, Paradise und Smith 2000) und selten Kreativität in der Informatik umfassender betrachtet (z. B. Glass 2006). Im Folgenden werde ich wesentliche Aussagen zur Kreativität aus Sicht des Faches Informatik analysieren.

### 3.1.1 Standpunkte zur Kreativität in der Informatik

*Computer science is the first engineering discipline ever in which the complexity of the objects created is limited by the skill of the creator and not limited by the strength of the raw materials. (Brian K. Reid in Frenkel 1987)*

Die Informatik wird von praktizierenden Informatikern als kreatives Gebiet angesehen.<sup>30</sup> „Die Konstruktion des Produkts erfordert Ideen, viele kleine und manchmal auch einige größere. Jedes Projekt ist ‚Neuland‘ – denn sonst könnte man das System bereits kaufen“ (Eich-Soellner 2005). Betont wird, dass – im Gegensatz zu anderen Disziplinen, wie z. B. der Chemie oder Physik – die Realitäten nicht fest sind. An der eigenen Welt kann „mitgebaut“ werden.

Innerhalb der Informatik wird vor allem dem Gebiet der Softwareentwicklung und des Programmierens eine kreative Rolle zugeschrieben; bspw. äußern sich verschiedene Pioniere der Informatik in diese Richtung (vgl. Lammers 1986; Littler 2005):

**Ray Ozzie:** *Programmers are very creative, self-directing, self-motivating people.*

**Bill Gates:** *So much judgment and creativity goes into a programming project.*

**Guido van Rossum:** *To me, [programming] relates strongly to creativity, which is very important to my line of work.*

Aus gleichem Grund fordern Noble und Biddle (2002) sogar, dass jeder Informatiker Programmiergrundkenntnisse besitzen sollte: *“It is there that the original source of creativity lies.”* Diese Auffassung ist vergleichbar mit Künstlern, die ihr Handwerk als „*Urquell des schöpferischen Gestaltens*“ beherrschen sollten (vgl. Gropius 1918). Möglicherweise verändert sich aber diese Bedingung mit dem Fortschritt der Informatik selbst. Die Grundlagen, die notwendig sind, um innerhalb der Informatik kreativ tätig zu werden, vereinfachen sich durch verständlichere Software und intuitive Benutzerschnittstellen kontinuierlich. Selbst Kreativitätsforscher nennen Programmieren als kreative Tätigkeit in einem Zug mit Malen und Geschichten schreiben (vgl. Hennessey 2007, S. 38).

Auch junge Informatiker sehen ihr Betätigungsfeld als *kreativ* an. In einer informellen Umfrage unter Studierenden und in einer Studie mit Lehramtsstudierenden und Referendaren (vgl. auch Kapitel 6) wurden Kreativität in der Informatik bejaht und verschiedene Perspektiven verdeutlicht. Unter anderem wurden folgende Standpunkte vertreten:

1. Besonders Softwareentwicklung ist kreativ innerhalb der Informatik:

*„Der Entwurf von Algorithmen ist unserer Meinung nach ein stark kreativer Prozess.“*

*„Nicht zuletzt ist die Software-Entwicklung eine schöpferische Tätigkeit, an deren Ende meist ein individuelles, einzigartiges (Unikat) Werk (Programm, Algorithmus) geschaffen wurde.“*

2. Kreativität in der Informatik basiert auf dem Bausteinprinzip:

*„Auch in der Informatik sind es Probleme, die sich nur kreativ mit Klötzen aus dem Informatikbaukasten lösen lassen.“*

3. „Logische“ und „künstlerische“ Kreativität sind möglicherweise zu trennen.

Für Informatiker ist die Frage um Kreativität in ihrem Metier offenbar interessant. In kontroversen Online-Diskussionen werden verschiedenste Aspekte von der Kreativität in der Softwareentwicklung erörtert (siehe Abbildung 9). Auch hierbei wird deutlich, dass der Kreativitätsbegriff häufig

---

<sup>30</sup> Vgl. (Leach und Ayers 2005; Glass 2006; Saunders und Thagard 2005; McBreen 2001) und Kapitel 3.1.2.

aus unterschiedlichen Verständnissen heraus verwendet wird, woraus sich zwangsläufig Kontrollversen ergeben.

**tomtiger:** Für mich ist Kreativität die Realisierung einer Idee. Mit einigen Zeilen Code ein effizientes Programm zu erstellen, das ist kreativ. Analytisches Denken benötigt der Maler auch, er muss nicht nur Farbmischungen etc. im Kopf haben und kombinieren können, er muss auch eine Vielzahl von Techniken beherrschen und richtig einsetzen. Nicht mehr macht ein Programmierer, nur dass der eben ein größeres Repertoire hat.

**tomtiger:** Ein richtiger Informatiker ist anzusehen wie ein Künstler, Programmierung und/oder Hardwareentwicklung, aber auch Administration ist äußerst kreativ, geradezu künstlerisch, ich benötige beispielsweise Inspiration, um arbeiten zu können. So wie ein Maler oder Bildhauer in seine Arbeit versunken ist, ist auch der Informatiker in seine Arbeit versunken, man lebt für sein Werk, Äußerlichkeiten sind wenig relevant.

**wollust:** Die eigentliche Schwierigkeit bei der Kunst des Programmierens ist ja außerdem gar nicht die Beherrschung der Programmiersprache an sich, sondern die Kunst, Probleme in kreativer und vor allem effizienter Weise zu lösen. Was programmiertechnisch vielleicht gut ist, muss deswegen noch lange kein guter Lösungsansatz für eine Problemstellung sein.

**albundyfan:** Also ich bin ja selbst Programmierer, aber ich würde nie, wie einige Schreiberlinge hier, behaupten, dass ich irgendetwas Kreatives vollbringe; auch wenn ich ein Problem löse, bin ich deshalb noch lange nicht kreativ, sondern habe einfach ein analytisches Denken, das mir in der Problemstellung hilft....

**markinger:** Die großen Probleme liegen eher in der Analyse von komplexen Sachverhalten und dann im ingenieurmäßigen Entwurf und Umsetzung einer Lösung. Wichtiger als Kreativität ist dabei vor allem die Nachvollziehbarkeit und Dokumentation der Lösung!

Abbildung 9: Ausschnitte einer Online-Forumsdiskussion zur Kreativität in der Informatik (fm4.orf.at 2004), Rechtschreibung korrigiert.

Die öffentliche Meinung zur Kreativität in der Informatik unterscheidet sich allerdings deutlich von der Meinung vieler Informatiker. Studien zeigen, dass Informatik eher als un kreativ, unsozial und anwendungsfern eingeschätzt wird (Margolis und Fisher 2002; AAUW 2000; Rich, Perry und Guzdial 2004; Barker, Garvin-Doxas und Roberts 2005). Nach einer Studie zum Image des Ingenieurberufs<sup>31</sup> bezeichnen drei Viertel der befragten Jungingenieure ihr Berufsfeld als kreativ, während diese Meinung nur von 27% der Öffentlichkeit in Deutschland geteilt wird ((VDE 2003), vgl. Abbildung 10). In diesem Punkt tut sich die größte Diskrepanz zwischen Eigen- und Fremdwahrnehmung auf. Möglicherweise ist hier auch ein Grund für die Rückläufigkeit intrinsisch motivierter Studienanfänger zu suchen (vgl. Romeike und Schwill 2006).

Während Befürworter eines kreativen Informatikbilds vor allem den schöpferischen Aspekt informatischer Tätigkeiten betonen, beziehen sich Kritiker v. a. auf Charaktermerkmale kreativer



Abbildung 10: Image des Ingenieurberufs (VDE 2003).

<sup>31</sup> Die erwähnte Studie bezieht sich auf den Ingenieurberufs insgesamt. Ein Großteil der Befragten Jungingenieure stammt allerdings aus dem Bereich Elektro- und Informationstechnik/Informatik. Es ist zu vermuten, dass das Bild der Informatik in der Öffentlichkeit nicht wesentlich anders ist. Eine Studie hierzu ist mir nicht bekannt.

Personen, wie Unangepasstheit und Eigensinn. So wird von Problemen berichtet, die „kreative“ Informatiker verursachen können, weil sie sich nicht an Absprachen halten und eigenmächtig „coole Features“ implementieren, die nicht gefragt waren. Statt Kreativität wird die Wichtigkeit von Disziplin und formalen Methoden unterstrichen. In seinem Weblog führt McIlree (2007) dazu aus:

*The developer [...] spent most of his time implementing 'cool' features to an application that the business stakeholders neither asked for or wanted.*

*A couple of development teams using 'creative,' let-it-all-hang-out-and-do-what-you-feel approaches to development had difficulty completing projects and getting them into production.*

Andere befürchten, dass Kreativität in der professionellen Softwareentwicklung in Hinsicht auf Entwicklungskosten, Qualität und langfristige Wartbarkeit einen Kontrollverlust herbeiführen könnte (Cusumano 1991). Glass (2006) und Kidder (2000) verdeutlichen dagegen, dass Disziplin und formale Methoden Kreativität nicht ausschließen müssen und dass auch in der professionellen Softwareentwicklung ein kreativer Ansatz erfolgversprechend sein kann. Ebenso wehrt sich Glass gegen Feststellungen, dass in der Softwareentwicklung vor allem Routinearbeiten (clerical tasks) zu absolvieren sind und dass Disziplin und formale Methoden Flexibilität und Heuristiken ausschließen bzw. diesen überlegen sind. Nach seiner Feststellung sind in der Softwareentwicklung vor allem kognitive Prozesse involviert, die für Kreativität charakteristisch sind.

Welches Bild von der Informatik haben Informatikschüler? Erfahrungen zeigen, dass viele Schüler das gesellschaftliche Bild von der Informatik teilen. Im späteren Teil dieser Arbeit werde ich diese wichtige Frage<sup>32</sup> an verschiedenen Stellen erneut aufgreifen.

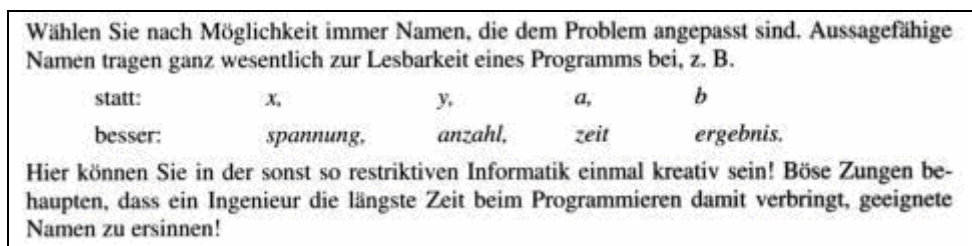


Abbildung 11: (Fehl-)Vorstellung von Kreativität in der Informatik (aus Küveler und Schwoch 2006, S. 42).

### 3.1.2 Kreative Persönlichkeiten der Informatik

Nicht selten sind es Personen, an denen Kreativität verdeutlicht wird. Wolfgang A. Mozart und Albert Einstein werden bspw. unstrittig als kreative Persönlichkeiten angesehen. Auch (oder gerade) in der Informatik gibt es eine Vielzahl kreativer Persönlichkeiten, welche die Entwicklung der Informatik vorangetrieben haben. Ebenso vermag das Fach kreative Personen herauszubilden. Leach und Ayers (2005) untersuchten Aufsätze über und Interviews mit prominenten Informatikern mit dem Ziel, die Rolle von Kreativität in deren Biographien zu ergründen. Übereinstimmend finden sich bei nahezu allen Informatikern:

- *frühe Begegnungen* mit Kreativität (z. B. im Elternhaus)
- ein *frühes selbständiges Auseinandersetzen* mit algorithmischen oder mathematischen Problemen, welche die Informatiker herausforderten
- *Situationen*, die sich für sie als unbefriedigend oder herausfordernd darstellten und zur weiteren Auseinandersetzung motivierten

<sup>32</sup> Diese Frage stellt sich nicht nur aus der schulischen Sicht für die Gewinnung und Motivierung von Informatikschülern als wichtig dar, sondern z. B. auch vor dem Hintergrund, Schüler für ein Informatikstudium zu begeistern.

- *Stimuli*, welche die Informatiker zu weiteren Gedanken inspirierten (andere Arbeiten, Hintergrundwissen, nahe stehende Personen, interessante Uni-Kurse, provokante Fragen)
- Eine *kreativitätsbegünstigende Umgebung* (die Möglichkeit, eigenen Fragestellungen nachzugehen, diese zu explorieren und mit möglichen Lösungen zu experimentieren)

Vergleichbare Aussagen finden sich bei Shasha und Lazere (1998).

“*I was intrigued by the vision of a highly intelligent, maybe super-intelligent artefact.*” (Edward A. Feigenbaum)

“*All understanding begins with our not accepting the world as it appears.*” (Alan C. Kay)

“*A good idea has a way of becoming and solving problems other than for which it was intended.*” (Robert E. Tarjan<sup>33</sup>)

Abbildung 12: Aussagen kreativer Persönlichkeiten zu ausschlaggebenden Faktoren ihrer Entwicklung (zitiert nach Shasha und Lazere 1998).

### 3.1.3 Kreativität in der Softwareentwicklung

Das Entwickeln von Software wird hinlänglich als kreative Tätigkeit bezeichnet. Während vergleichsweise allerdings beim Komponieren von Musik Kreativität generell nicht in Frage gestellt wird, ist das bei der Softwareentwicklung anders, da Informatik von vielen Menschen grundsätzlich nicht als kreatives Fachgebiet angesehen wird. Um Gemeinsamkeiten zwischen den beiden Tätigkeiten aufzuzeigen, werde ich im Folgenden Parallelen zwischen musikalischer Komposition und Improvisation und dem Entwickeln von Software (Softwaredesign) darstellen<sup>34</sup>. Um die Parallelen zum *Softwaredesign* deutlicher zu machen, werde ich anstatt von musikalischer Komposition/Improvisation von *Musikdesign*<sup>35</sup> sprechen.

#### Ein fachübergreifender Blick

Design bezeichnet nach dem *Oxford English Dictionary* (Oxford 2007):

1. *A plan or drawing produced to show the look and function or workings of something before it is built or made.*
2. *The art or action of producing such a plan or drawing.*
3. *Underlying purpose or planning: the appearance of design in the universe.*

Offensichtlich ist *designen* eine urkreative Tätigkeit. Die Definitionen enthalten aber auch dreimal die Aussage, dass *Planen* am Design beteiligt ist. Spreche ich nun von Musikdesign mag sich z. B. ein Jazzliebhaber darüber wundern. Basiert Jazzmusik nicht auf Spontaneität? Kann das geplant werden? Hört man von bekannten Jazzgrößen nicht immer wieder Sätze wie „Ich spiele einfach was ich fühle“? Die Antwort auf diese Fragen ergibt sich aus einer Aussage von Wynton Marsalis<sup>36</sup>:

*Jazz is not just, ‘Well, man, this is what I feel like playing.’ It’s a very structured thing that comes down from a tradition and requires a lot of thoughts and study.*

(Marsalis, zitiert nach Berliner 1994)

<sup>33</sup> Erfinder des Planaren Testens, (planary testing), Anwenden von depth-first search, efficient network flows, union find und amortization.

<sup>34</sup> Ein weitere interessante Betrachtung ist der Vergleich des Entwickelns von Software und kreativem Schreiben (vgl. Ruete 1990).

<sup>35</sup> Der Begriff *Musikdesign* wird selten verwendet, da die Begriffe Komposition oder musikalische Improvisation intuitiver sind. Vor allem im Kontext moderner Musik und der Entwicklung von Musik mit Hilfe von Computern wird der Begriff „music design“ zunehmend auch für den Prozess des musikalischen Kreierens verwendet.

<sup>36</sup> Wynton Marsalis ist einer der bedeutendsten Jazztrompeter der Gegenwart.

Ein Jazzmusiker greift auf einen großen Vorrat an Mustern, Modellen, Konzepten, Skalen, Harmoniewendungen und Outlines zurück (vgl. Ligon 1996; Barrett 1998), die er im Moment der Improvisation im Design seiner Musik<sup>37</sup> anwendet und in Echtzeit aufführt.<sup>38</sup> Auch ein Komponist tut Ähnliches: Basierend auf einer beherrschten Fülle an Wissen (einem Wissens-Pool an Fakten, Konzepten etc.) entsteht das musikalische Design seiner Komposition. Hierbei sind entsprechend des Kulturkreises, der Hörgewohnheiten und persönlicher Vorlieben gewisse Richtlinien und Vorgaben wie Tonalität, Stil sowie harmonische und rhythmische Regeln einzuhalten.<sup>39</sup> Voraussetzend, dass es in der Regel einen Grund für ein musikalisches Werk gibt,<sup>40</sup> lassen sich die genannten Faktoren in einem Modell des musikalischen Designprozesses abbilden (vgl. Abbildung 13). Dieses Blackbox-Modell stellt Schlüsselfaktoren dar, die in der Musikausbildung gelehrt werden<sup>41</sup> (vgl. Scheuer 2003; Ligon 1996; Henze und Baisch 1998; LaMotte 1996).

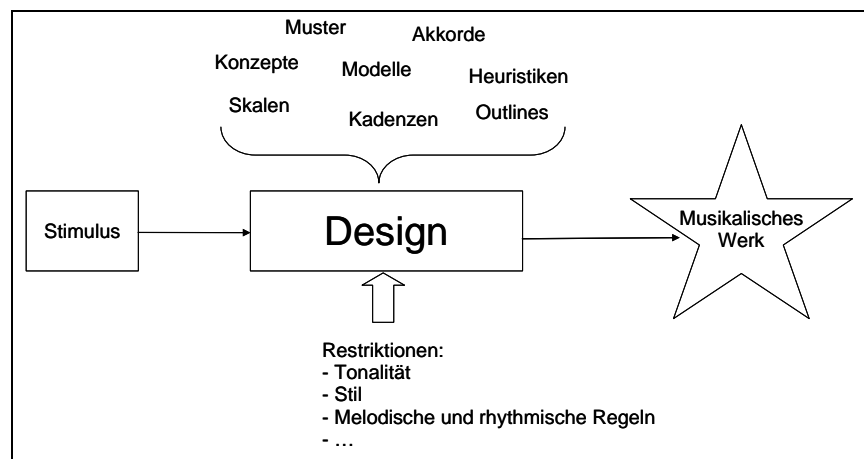


Abbildung 13: Blackbox-Modell mit Faktoren, die den musikalischen Designprozess beeinflussen.

Software-Design ist diesem Vorgehen in gewissem Sinn ähnlich. Betrachtet man die Einflussfaktoren im Blackboxmodell werden Parallelen deutlich: Auch beim Software-Design steht ein Stimulus am Beginn des Prozesses. Oft wird dieser Problem, Aufgabe oder Anforderung genannt. Es kann aber auch jede andere Motivation für jemanden sein, eine Software zu entwickeln. Der Designprozess wird dabei stark von Bedingungen und Einschränkungen (Richtlinien, Vorgaben, Regeln) beeinflusst, die vom Softwareentwickler beachtet werden müssen. Zu diesen gehören die Umstände wie, wo und von wem die Software benutzt werden soll genauso wie die Grenzen der verwendeten Programmiersprache, verfügbare Ressourcen und vieles mehr. Während des Designprozesses verwendet der Programmierer einen Schatz an Problemlösemethoden, Vorgehensweisen,

<sup>37</sup> Das ist es, was einen guten Jazzmusiker ausmacht: Hinter seiner Musik steckt ein durchdachtes Design.

<sup>38</sup> Barrett (1998) nennt das Erlernen der Grundlagen und Regeln "Preparing to be spontaneous".

<sup>39</sup> Auch hier zeigen sich unterschiedliche Auffassungen von Kreativität. Möchte ein Musiker zu seiner Lebzeit als kreativ gelten (und damit auch entsprechend entlohnt werden), muss er sich vermutlich stärker an die Normen und Geschmäcker seines Kulturkreises halten. Große Komponisten wie bspw. Franz Schubert und Gustav Mahler werden heutzutage rückblickend als sehr kreativ eingeschätzt. Zu ihrer Lebzeit wurde allerdings ihr Schaffen, und damit ihre Kreativität, nur bedingt anerkannt.

<sup>40</sup> In der Regel steht vor dem Beginn einer musikalischen Komposition oder Improvisation ein Impuls, eine Impression oder ein Auftrag.

<sup>41</sup> Bemerkenswert ist an dieser Stelle allerdings, dass die Grundfertigkeiten in der Musik offenbar um ein Deutliches komplexer sind als die für das Entwickeln von Software nötigen Grundlagen. Dies hat zur Folge, dass im schulischen Musikunterricht wirklich kreative, also z. B. komponierende, Tätigkeiten kaum stattfinden. So stellt Scheuer (2003) fest: „Allerdings lassen sich die vorliegenden Modelle des Komponierens mit Laien [...] nicht ohne weiteres auf die Unterrichtssituation in der allgemeinbildenden Schule übertragen, da sie sich vor allem auf den Einzelunterricht beziehen und nicht aus der schulpädagogischen, sondern primär aus der Sicht des Komponisten und in der Tradition von Satzlehren formuliert sind.“ Das Erlernen der Grundlagen für Jazzimprovisation (ohne Berücksichtigung des Beherrschens eines Instruments) dauert sogar mehrere Jahre (Barrett 1998).

Heuristiken und Erfahrungen und wendet diese zur Findung und Realisierung einer Softwarelösung an. Auch dieser Prozess resultiert in einem Produkt.<sup>42</sup>

Bestätigt werden diese Faktoren durch eine Studie mit professionellen Softwaredesignern von Guindon und Curtis (1988). In ihrer Untersuchung zum Designprozess spezifizierten und kategorisierten sie die Randbedingungen (z. B. „design process constraints“, „problem domain constraints“ oder „functional constraints“) und die Wissensressourcen (z. B. Algorithmen, Designschemata, Designmethoden).

Dieser Prozess kann kreativ genannt werden, so lange er auf ein neues Problem angewandt wird, neue Software das Ziel ist und es kein bekanntes Lösungsmuster für die Aufgabe gibt. Kreativität wird hierbei genauso verlangt wie sie in Jazzimprovisation oder musikalischer Komposition benötigt wird.<sup>43</sup> Offensichtlich werden unterschiedliche Prozesse auch einen unterschiedlichen Grad an Kreativität verlangen. Das Modell zeigt aber wichtige Faktoren auf, die den kreativen Prozess beeinflussen: Ein umfangreicher Vorrat an Wissen, Konzepten und Erfahrungen sind die Basis für kreatives Design in der Softwareentwicklung.<sup>44</sup> Im Informatikunterricht wird bereits auf die Vermittlung dieser Inhalte viel Wert gelegt. Darüber hinaus muss allerdings die Frage, wie das erlangte Wissen im kreativen Designprozess angewandt werden kann, stärker berücksichtigt werden.

### Softwaredesign als kreativer Prozess

Auch wenn das Wissen um die Einflussfaktoren des kreativen Prozesses wichtig ist, sagt das Blackbox-Modell nicht viel über den Designprozess selbst aus. In einer empirischen Studie zum Softwaredesignprozess identifizierten Curtis, Guindon et al. (1987) fünf Schritte:

1. *Verstehen des Problems*
2. *Dekomposition des Problems in Ziele und Objekte*
3. *Auswählen und Erstellen von Plänen zur Problemlösung*
4. *Implementieren der Pläne*
5. *Reflektieren des Designproduktes und -prozesses*

Diese Schritte beschreiben den Designprozess noch recht allgemein. Für Lehrer implizieren sie die unterschiedlichen Phasen, die beim Unterrichten von Softwaredesign betont werden müssen. Allerdings geben diese Schritte im Wesentlichen auch den generellen Problemlöseprozess wieder und haben auch einiges mit dem Software-Life-Cycle gemeinsam. Interessanter ist die genauere Betrachtung des Schrittes „Auswählen und Erstellen von Plänen“ als Kern des Designprozesses. Hier wurden die folgenden Teilschritte identifiziert:

- 3.1. *Erzeugen ein mentales Modells einer möglichen Lösung für das Problem*
- 3.2. *Ausführen des Modells in Gedanken um zu prüfen, ob es das Problem tatsächlich löst (Simulation)*
- 3.3. *Bei unkorrektem Ergebnis des Beispiels, Überarbeitung des Modells zur Behebung der Mängel, anschließend erneute Prüfung des Beispiels*
- 3.4. *Bei schließlich korrektem Ergebnis werden ein weiteres Beispiel ausgewählt und die Schritte zwei und drei wiederholt*
- 3.5. *Nachdem genügend Beispiele den Test in Schritt 4 bestanden haben, wird angenommen, dass das Modell ein passendes Designmodell darstellt und die Repräsentation des Designs beginnt (vgl. Glass 2006, S. 339)*

<sup>42</sup> Mitunter endet die Arbeit des Softwaredesigners mit dem Modell der Software und die Implementation wird anderswo übernommen. Dies wäre vergleichbar mit der schriftlichen Skizze einer musikalischen Komposition.

<sup>43</sup> Nicht zuletzt wird auch behauptet, dass kreative Musiker grundsätzlich auch gute Informatiker sein könnten (Melymuka 1998).

<sup>44</sup> Die Kreativitätsforschung gibt sogar an, dass wenigstens 10 Jahre Betätigung in einem Fachgebiet notwendig sind, um h-kreative Leistungen hervorbringen zu können.

Diese Schritte machen den kognitiven und iterativen Charakter des Software-Designs deutlich. Offensichtlich sind auch Versuch und Irrtum und heuristische Herangehensweisen natürlich und wichtig im Software-Design. Dieser Aspekt wird häufig übersehen, wenn Lehrende nur den Top-Down-Modellierungsansatz unterstützen.

Die Feststellung, dass ein großer Teil des Prozesses kognitiver Art ist, wird von Goorhuis' Theorie der konstruktivistischen Modellbildung in der Informatik geteilt (Goorhuis 1994). Goorhuis beschreibt vorgenannte Schritte 1 und 2 des Softwaredesignprozesses als Kogitationsphase, welche zu einem internalen Modell führt. Daran schließt sich, analog zu den Schritten 3.1. bis 3.5., eine Konstruktionsphase mit einem zweiten internalen Modell als Ergebnis an, welches den Ausgangspunkt für die Realisierung des Modells darstellt. Weitere Belege für die Wichtigkeit kognitiver Faktoren finden sich in Aussagen von Softwarepionieren (Lammers 1986).

Das Modell des kreativen Softwaredesignprozesses soll vervollständigt werden, indem Parallelen zu Amabiles Kreativitätsmodell gezogen werden (Amabile 1996). Das Modell basiert auf drei Komponenten, die den kreativen Prozess beeinflussen: Domainrelevantes Fachkönnen/Expertise (der von mir genannte Wissens-Pool), Kreativitätsfertigkeiten (z. B. Erfahrung, Heuristiken und Ideefindungstechniken) und Motivation für die Aufgabe.<sup>45</sup> Auch in diesem Modell werden die verschiedenen Phasen grob eingeteilt in Problemidentifikation, Vorbereitung, Generierung von Reaktionen, Überprüfung der Responses, Kommunikation und das Produkt zum Schluss.

Da ich, die bis hierher angeführte Argumentation zugrunde legend, das Designen von Software als kreativen Prozess betrachte, können die Erkenntnisse nun in einem Modell zusammengefasst werden, welches den kreativen Softwaredesignprozess darstellt (vgl. Abbildung 14).

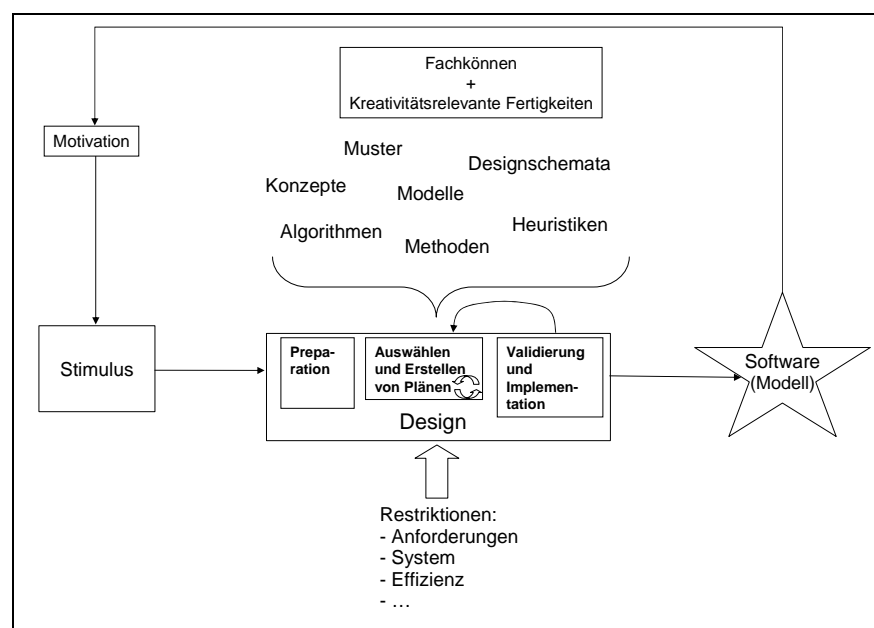


Abbildung 14: Modell des kreativen Softwaredesignprozesses.

Ein Anreiz, z. B. eine Idee, ein Problem oder eine Aufgabe, stellt den Beginn des Prozesses dar. Ein fremdes gestelltes Problem regt weniger intrinsisches Interesse an als ein selbstgestelltes Problem (vgl. Ruscio und Amabile 1996). In der zweiten Phase erfolgt eine Auseinandersetzung mit dem Problem, dabei sind Fachkenntnisse entscheidend und ermöglichen es, verschiedene Möglichkeiten zielgerichtet auszuprobieren. Sowohl Fachkenntnisse als auch kreativitätsrelevante Fertigkeiten und Erfahrungen sowie Motivation spielen eine wesentliche Rolle beim Explorieren, Selektieren und Evaluieren möglicher Lösungen. Das Produkt des kreativen Prozesses wirkt sich schließlich wieder auf die Motivation aus, ein neues Projekt in Angriff zu nehmen.

<sup>45</sup> Wie in Kapitel 2.2 gezeigt, Motivation ist essentiell für Kreativität.

Interessant ist die fachübergreifende Relevanz von Regeln für die Kreativität. Regeln sind besonders vordergründig in der Softwareentwicklung von Bedeutung und stehen traditionell im Gegensatz zur unbegrenzten Freiheit des kreativen Denkens. Boden (1995) macht deutlich, dass diese Beschränkungen des Denkens eben nicht im Widerspruch zur Kreativität stehen, sondern diese sogar erst möglich machen. Erst durch die vorhandenen Beschränkungen wird ein Gebiet struktureller Möglichkeiten kartographiert, welches sich anschließend erforschen und zu etwas Neuem verändern lässt.

Abschließend kann festgestellt werden, dass Kreativität als ein essentieller Teil in der Softwareentwicklung notwendig ist. Sinnvoll bleibt, das Entwickeln von Software als Ganzes zu betrachten und dabei auch persönliche Faktoren zu berücksichtigen. Mit Blick auf die Schulinformatik ist zu beachten, dass Inhalte und Ziele eines allgemeinbildenden Informatikunterrichts nicht eine professionelle Softwareentwicklung sein können, sondern der Einblick in die Methoden der Informatik und ein Verständnis dieser. Eine Betonung der kreativen Aspekte der Softwareentwicklung verspricht, dieses zu erleichtern und zu motivieren.

### 3.1.4 Das Bausteinprinzip in der Informatik als Katalysator für Kreativität

Ein solides Grundwissen im Fachgebiet ist eine notwendige Voraussetzung für kreative Prozesse. Vorteilhaft für Kreativität in der Informatik ist, dass die zugrunde liegenden Fakten und Konzepte als *Bausteine* aufgefasst werden können, welche wohl definiert und dokumentiert und damit einfach anzuwenden sind. Als *Baukästen* können bspw. aufgefasst werden:

- Datentypen, welche nach dem Baukastenprinzip modelliert werden (vgl. Schubert und Schwill 2004),
- Programmierparadigmen mit ihren Baukastensystemen,
- Designpatterns,
- die Fundamentalen Ideen der Informatik, welche die wichtigsten Konzepte und Strategien der Informatik beschreiben (Schwill 1993).

Glass (2006) spricht bezüglich verinnerlichter Wissensbausteine von einem Satz von Primitiven, die jeder Software-Designer und Programmierer besitzt und welcher von Person zu Person variiert. Eine Primitive bezeichnet danach den Punkt, an welchem der Designer mit dem „Designen“ aufhört und wo das Implementieren beginnen kann – ein Satz gut verstandener Elemente, die keines weiteren Nachdenkens bedürfen. Entsprechend wird beim Entwurf von Software auf eine Anzahl an Primitiven zurückgegriffen, die wie Bausteine zusammengefügt werden. Auch auf abstrakterer Ebene kann Softwareentwicklung als das Komponieren von Softwarebausteinen verstanden werden. So verdeutlicht Leon (1997):

*Creativity in software development comes form putting together established building blocks to create real solutions to problems. [...]*

*Creativity means not wasting the time and resources on a developing a function that can be substituted by an existing and time-tested one. Whatever problem you are working on, there is a high probability that, it or a similar one has already been solved. Most software solutions involve putting together common building blocks, which have been used many, many times.*

Viele Programmieransätze folgen diesem Prinzip, so z. B. das Kombinieren von Code in Prozeduren in der strukturierten Programmierung oder Kapselung in der Objektorientierung.

Ein Vorteil von Bausteinen für Kreativität wird bei der Betrachtung spielender Kinder offensichtlich: Kinder spielen gern mit Bausteinen, weil diese eine begrenzte und handhabbare Anzahl von Verwendungsmöglichkeiten haben, die Verwendungsmöglichkeiten sind verständlich und eine nahezu unbegrenzte Anzahl an verschiedenen Konstruktionen kann mit ihnen erschaffen werden. Beim Spiel mit solchen Bausteinen entwickeln Kinder eine enorme Kreativität und lernen sowohl über Strukturen als auch über Stabilität.

Bruce, Buckingham et al. (2004) verwenden den Begriff *Bausteine* zur Beschreibung eines Ansatzes, wie Lernende an das Programmieren herangehen. Hierbei stellt Experimentieren und

Explorieren des vorhandenen Materials einen wesentlichen Teil des Lernprozesses dar. Auch Programmierumgebungen, die speziell für das Erlernen der Programmierung konzipiert wurden, machen von der Baustein-Metapher Gebrauch. Basierend auf LOGO, welches bereits bausteinartige Anweisungen verwendet, wurde mit LOGO-Blocks das Bausteinkonzept erfolgreich visualisiert. Inzwischen existieren einige darauf aufbauende Entwicklungen. Programmierumgebungen wie Scratch<sup>46</sup> oder Alice<sup>47</sup> lassen Programmieranfänger Software auf eine Art und Weise „bauen“, als wenn sie mit Bausteinen spielen würden: Durch Experimentieren, Lernen, Verstehen der Konzepte und kreatives Spiel. Dieses Verständnis von Software scheint für kreative Ideen sehr fruchtbar zu sein. Aus dieser Perspektive ist Kreativität in der Informatik für jeden Schüler erreichbar; sie basiert auf dem Zusammensetzen von gut verstandenen Bausteinen um Lösungen für bekannte und unbekannte Probleme zu finden und dabei eigene Ideen umzusetzen. Für den Informatikunterricht suggeriert diese Sichtweise, Konzepte der Informatik in einer Art und Weise zu unterrichten, dass die zugrunde liegenden Bausteine deutlich und unterscheidbar und zugleich die Beziehungen der Bausteine zueinander bewusst werden.

### 3.1.5 Innovationen der Informatik

Innovation bezeichnet die Verwirklichung kreativen Denkens. Trotz der relativ jungen Geschichte der Informatik ist aus ihr bereits eine Fülle von Innovationen hervorgegangen. Damit hat die Informatik – vielleicht wie keine andere Wissenschaft vor ihr – in kürzester Zeit die Gesellschaft verändert. Auch das Selbstverständnis der Informatik wird zunehmend von dem bestimmt, was sie geschaffen hat. So illustriert die Gesellschaft für Informatik ihr Verständnis für Informatik auf einer 24-seitigen Darstellung von Innovationen durch Informatik ((GI 2006), vgl. Abbildung 15).

Viele Innovationen zogen eine Reihe an (informatischen) Anwendungen nach sich und wirkten damit als Anlass für kreative Leistungen und Produkte, welche zumeist im Bereich der Software anzusiedeln sind. Eine Möglichkeit, Innovationen im Unterricht herauszustellen, ist die historische Betrachtung der Informatik (bspw. in Thomas 2005), wie es auch in verschiedenen Schulbüchern gemacht wird.<sup>48</sup> Viele dieser geschichtlichen Darstellungen haben allerdings gemeinsam, dass nur die (hardware-) technische Seite der Entwicklung thematisiert wird. Der Großteil heutiger Innovationen vollzieht sich allerdings softwareseitig. Ebenso ist es Schülern fast ausschließlich im Bereich der Software möglich, eigenständig informatisch kreativ tätig zu werden. Eine Betrachtung der Softwareinnovationen kann dies zusätzlich anregen. Eine ausführliche Darstellung dieser findet sich bei Wheeler (2008).

Letztendlich beeinflussen Innovationen der Informatik auch die Gesellschaft, welche darüber hinaus dann wieder ihre Ansprüche und Einwirkungen auf die Informatik ausübt, was allerdings auch Probleme mit sich bringen kann (siehe Datenschutzproblematik oder Computerkriminalität).

<p>Innovation durch Informatik...</p> <ul style="list-style-type: none"> <li>... für die Wissenschaft</li> <li>... für die Wirtschaft</li> <li>... für die Technik</li> <li>... auf dem Bildungssektor</li> <li>... für die Kultur</li> <li>... für Individuum und Gesellschaft</li> </ul>
--

Abbildung 15: Informatik als Darstellung anhand ihrer Innovationen (GI 2006).

<sup>46</sup> <http://scratch.mit.edu>.

<sup>47</sup> <http://www.alice.org>.

<sup>48</sup> Hierbei darf allerdings nicht vergessen werden, dass es sich bei vielen Innovationen nur um Vorläufer der Informatik handelt, nicht um Innovationen der Informatik selbst.

## 3.2 Mensch-Perspektive

*Software Development is and always has been fun. (R. Glass)*

In den Lebensläufen bedeutender Informatiker wurde deutlich, dass Kreativität für viele von ihnen eine wichtige Rolle spielt. Die Möglichkeit, sich kreativ ausdrücken zu können, kann eine wichtige Motivation darstellen, sich mit Informatik zu beschäftigen. So übt Softwareentwicklung einen faszinierenden Einfluss auf Programmierer aus. Open-Source-Programmierer verbringen oft sogar mehr als 20 Stunden ihrer wöchentlichen Freizeit mit dem Entwickeln für Software, für die sie nicht einmal bezahlt werden. Diese Menschen treibt offenbar eine starke intrinsische Motivation, wie man es auch von Künstlern kennt, die sich intensiv ihrer Arbeit widmen (Moravcsik 1974).

Lakhani und Wolf (2005) und Luthiger Stoll (2006) untersuchten die Motivation von Softwareentwicklern in Open-Source-Projekten. Sie identifizierten auf Vergnügen basierende intrinsische Motivation, die aus dem Gefühl, etwas Kreatives zu tun, herausragt, als stärkste und verbreitetste Triebkraft. Offensichtlich stehen auch hier Kreativität und intrinsische Motivation in einem sich wechselseitig bedingenden Verhältnis. Zudem fanden sie eine Reihe kreativitätsverwandter Faktoren, die für die intrinsische Motivation und folglich für die Beteiligung am Projekt verantwortlich waren. Dazu gehören:

### **Gebrauch:**

Eine Software benutzen zu wollen war in der Studie von Lakhani und Wolf ein wichtiger Grund, sich an Open-Source-Projekten zu beteiligen. Demnach hat der spätere Entwickler als Nutzer ein Problem, welches er mit der vorhandenen Software nicht lösen kann, so dass er sich entschließt, eine Software selbst zu programmieren oder vorhandene Software entsprechend anzupassen. Saunders und Thagard (2005) prägen im gleichen Zusammenhang den Begriff „frustration-based creativity“, welche den Drang zu neuen Entwicklungen aufgrund einer fehlenden, aber benötigten Funktionalität bezeichnet. Dies ist eine typische Quelle für Kreativität in der Informatik.

### **Reputation:**

Eine der Maximen der Open-Source-Community ist die, dass die Beiträge der einzelnen Entwickler genau nachverfolgbar sind. Damit hat jeder einzelne die Möglichkeit, eine Reputation aufzubauen, die bei einer späteren Bewerbung hilfreich sein kann. Auch lässt sich so das Ansehen innerhalb der Community stärken.

### **Identifikation mit der Gruppe:**

Die Einbindung einer Person in eine Gruppe kann zu einer starken Identifizierung mit den Gruppenzielen führen.

### **Lernen:**

Die Möglichkeit, in den Projekten etwas zu lernen, ist ebenfalls ein Grund für Open-Source-Programmierer, sich an den Projekten zu beteiligen. Hier haben sie die Möglichkeit, Erfahrungen zu sammeln und ihre Fertigkeiten zu verbessern.

Die hier dargestellten Eigenschaften lassen sich darauf zurückführen, dass das Erstellen von Software, welches nach vorgenannter Definition als kreativ zu bezeichnen ist, Spaß macht. Ähnlich wie bei der Kunst oder anderen kreativen Tätigkeiten fühlen sich Programmierer wohl bei dem, was sie tun, und haben Spaß, verbunden damit, etwas Nützliches zu tun, ggf. sich einer Gruppe zugehörig zu fühlen und sogar beim Lernen.

Wie Blum (1991) beschreibt, ist ein professioneller Softwareentwickler zwischen Spaß und Eintönigkeit hin und her gerissen. Während im Software-Life-Cycle einige Tätigkeiten, wie z. B. Dokumentation, Organisation, Testen etc. eher eintönig sind, so sieht auch er Programmieren als die Phase, die am meisten Spaß macht.

Luthiger Stoll (2006) stellt fest, dass kommerzielle Softwareentwickler weniger Spaß an ihrem Tun empfinden als Freizeitprogrammierer. Er führt das auf die unterschiedlichen Produktionsbedingungen, unter welchen Software-Entwickler arbeiten, zurück:

*Diese Differenzen sind aber nicht durch den Druck von Abgabeterminen oder die Existenz von formalen Hierarchien verursacht, [...] Der Grund dafür, dass Programmieren im Kontext von Open Source mehr Spaß macht, könnte vielmehr darin liegen, dass die Software-Entwickler in Open-Source-Projekten eine glaubwürdigere Projekt-Vision und vor allem eine ihren Bedürfnissen besser entsprechende Herausforderung vorfinden. (Luthiger Stoll 2006, S. 201)*

Die Ergebnisse dieser Studie werfen die Frage auf, ob sich aus den Umständen, welche Motivation und Kreativität bei Open-Source-Programmierern erwecken, Implikationen für die Gestaltung des Informatikunterrichts ableiten lassen. Auf mögliche Implikationen und die Berücksichtigung persönlicher Herausforderungen in der Unterrichtsgestaltung werde ich in Kapitel 4 eingehen.

### 3.3 Kunst-Perspektive

*Programming is a creative process. [...] You can take pride in your work. It's construction; it's art. Why does anybody like to build things, or draw things, or sculpt things? I don't know that I can verbalize it, but I'd say it's a lot of fun. To me, it's a very creative, artistic medium. (Brian K. Reid, zitiert nach Frenkel 1987, S. 823)*

Kunst, die *schöpferische Tätigkeit des Menschen mit Tönen, Sprache oder verschiedenen Materialien* (Bertelsmann 2004) ist der Inbegriff von Kreativität im traditionellen Sinn.<sup>49</sup> Im vorherigen Abschnitt habe ich bereits dargestellt, dass das Entwickeln von Software mit künstlerischen Prozessen, wie z. B. musikalischer Komposition, verglichen werden kann. So erstaunt es nicht, dass einige Informatiker ihr Tun als Kunst verstehen: Der Informatiker als Künstler. Hier finden sich zwei verschiedene Sichtweisen. Zum einen richtet sich die Aufmerksamkeit auf den Prozess selbst (meist Programmieren); der Informatiker *versteht* das, was er tut, als Kunst, basierend auf Fertigkeiten, Wissen und Freude am Schaffen schöner Software (vgl. Knuth 1974). Diesem Verständnis entspringt das ästhetische Empfinden für „schönen Code“ und „ansprechenden“ Softwareentwurf. Zum anderen wird weniger der Prozess reflektiert, sondern der Informatiker (auch hier oftmals Programmierer) *betätigt sich* als Künstler. In dieser Sichtweise richtet sich die Aufmerksamkeit auf das kreative, künstlerische Produkt, ein meist programmiertes Kunstwerk, welches Expertise, Eleganz, oft auch Effizienz und Kreativität gleichermaßen ausdrückt. Festivals und eine aktive Community/Szene bieten diesen Programmierern ein Podium.

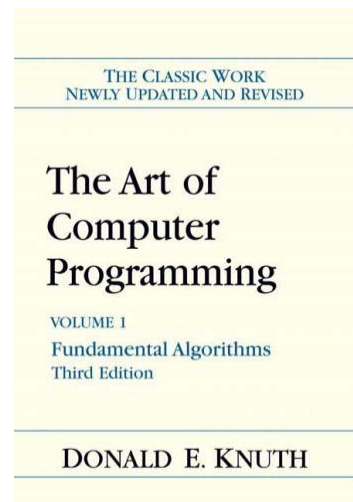


Abbildung 16: Knuth: The Art of Computer Programming.

In einem anderen Zusammenhang steht Informatik mit Kunst dort, wo sich *Künstler* als Informatiker oder Programmierer betätigen. Diese haben die Methoden der Informatik und die Möglichkeiten, die sich daraus ergeben, für sich entdeckt und wenden sie zum Erschaffen von Kunstwerken an (vgl. Maeda 2004; Trogemann und Viehoff 2005). Im Vordergrund stehen hierbei nicht Effizienz, Eleganz oder Zuverlässigkeit der Software, sondern die Weiterentwicklung von Ideen und das ästhetische Gelingen der Arbeit.

Einen zusätzlichen Beitrag für die Erschaffung von Kunstwerken und für die künstlerische Betätigung leistet die Informatik, indem sie Werkzeuge bereitstellt, die künstlerisch genutzt werden

<sup>49</sup> Allerdings verhält es sich mit vielen Kunstwerken so, dass ein Kriterium der herausgearbeiteten Definition von Kreativität verletzt wird: die Nützlichkeit. Der Nutzen eines Kunstwerks liegt oft in der subjektiven Betrachtung und am Genuss des Kunstwerks per se, ohne weiteren Nutzen darüber hinaus.

können. Hierzu zählt unterschiedlichste Software für Bild-, Musik- und Videoverarbeitung bis hin zu interaktiven Komponenten für Live-Installationen. Mit diesen Hilfsmitteln gelingt es immer mehr Menschen, sich künstlerisch kreativ zu betätigen.

Darüber hinaus arbeiten in verschiedenen Projekten Künstler mit Informatikern gemeinsam, in denen interdisziplinär beide Seiten profitieren. In der informatischen Bildung werden interdisziplinäre Projekte verwendet, um das Fach Informatik interessanter zu gestalten, die Motivation der Schüler und Studenten zu erhöhen und um das Fach für Frauen interessanter zu gestalten (vgl. Barker, Garvin-Doxas und Roberts 2005).

Tatsächlich wurden in den letzten Jahren sogar Auszeichnungen auf New-Media-Art-Festivals für „artistic software“ eingeführt. Auch lassen sich Museen mit Softwareausstellungen finden.

### 3.3.1 Informatiker als Künstler

#### Die Kunst des Programmierens

Programmieren wird von verschiedenen Informatikern als Kunst angesehen (vgl. Pomberger und Dobler 2008; Aupperle 2002; Graham 2004). Einen wesentlichen Beitrag zu dieser Sichtweise leistete Donald Knuth (1974) mit seinem Vortrag zur Verleihung des Turing Award. Wie Knuth schrieb, stand diese Auffassung allerdings in ständiger Konkurrenz mit dem Bestreben der Informatik, als Wissenschaft angesehen zu werden. In diesem Kontext zitiert er aus den *Communications of the ACM*:

*If computer programming is supposed to become an important part of computer research and development, a transition of programming from an art to a disciplined science must be effected.* (Bauer, Juncosa und Perlis 1959; zitiert nach Knuth 1974, S. 667)

Offenbar beinhaltet die Betrachtung als Kunst etwas Unerwünschtes, das für einige Wissenschaftler nicht zum Bild der Informatik passt. Dem tritt Knuth allerdings entschieden entgegen:

*[...] Computer science is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty. A programmer who subconsciously views himself as an artist will enjoy what he does and will do it better.* (Knuth 1974, S. 673)

Weiterhin führt Knuth aus, dass sich Kunst und Wissenschaft nicht gegenseitig ausschließen müssen, sondern im Programmieren gut ergänzen.

Mit der Betrachtung von Programmieren als Kunst hat sich bei Programmierern ein ästhetisches Empfinden dafür durchgesetzt, was „schöner Code“ und was ein „ansprechender“ Softwareentwurf ist und was nicht (Bond 2005). Solche Programmierästhetik bzw. solches Schönheitsempfinden kann sich auf den Akt des Programmierens als auch auf das Programm selbst beziehen. So berichtet Knuth (2001) über Programmtexte, welche Eleganz oder sogar Witz besitzen und wiederum von anderen, deren Lesbarkeit ein Graus ist.

Nach Bond (2005) gibt es eine Übereinstimmung zwischen Softwarepraktikern, dass sich die Ästhetik eines Programms an folgenden, von Knuth 1974 postulierten Eigenschaften festmachen lässt: Korrektheit, Wartbarkeit, Lesbarkeit, Übersichtlichkeit, Anmut der Nutzerinteraktion<sup>50</sup> und Effizienz. Dieses Ergebnis von Knuths Überlegungen zur Ästhetik von Programmen führte gar zur Philosophie des „Literate Programming“, welches eine gute Lesbarkeit für Programme fordert. Gute Lesbarkeit als Teil eines guten Programmierstils findet sich als Lernziel auch im Informatikunterricht wieder.

Nicht zu vernachlässigen ist der motivationale Einfluss der künstlerischen Betätigung. Littler (2005) befragte verschiedene erfahrene Programmierer, inwieweit sie ihr Tun als Kunst bezeichnen würden und was diese Betrachtungsweise für sie bedeutet:

<sup>50</sup> Eng.: „grace of interaction with users“.

Rossum<sup>51</sup> bejaht die Frage: „*If there was no art in it, it wouldn't be any fun, and then I wouldn't still be doing it after 30 years.*“ Stroustrup<sup>52</sup> unterstreicht vor allem die Gemeinsamkeit von Kunst und Handwerk, wobei er Programmieren eher als Handwerk auffasst. Graham<sup>53</sup> betont, dass er auf ähnliche Art und Weise Programme entwickelt, wie es Schriftsteller, Maler und Architekten tun: indem sie zuerst ihr Werk skizzieren und dann nach und nach vervollständigen. Diesem Ansatz folgte er von Anfang an bis heute, auch wenn es nicht dem Vorgehen entspricht, wie er es im College gelernt hat. Auch Stallman<sup>54</sup> beschreibt Programmieren als Kunst, differenziert aber zwischen Kunst, die allein Schönheit zum Ziel hat, und Kunst, die nützlich ist. Programmieren ordnet er – bis auf wenige Ausnahmen – letzterem zu.

Auch bei der Betrachtung von Programmieren als Kunst spielt die persönliche Auffassung vom dem, was man unter Kunst versteht – ähnlich wie bei der Auffassung zur Kreativität – eine wichtige Rolle. So erläutert Erik de Castro Lopo<sup>55</sup>, dass er unter Kunst das bewusste Brechen von Regeln und Konventionen versteht, die dem Kunstwerk ein Element der Überraschung verleihen. Da Programmieren einer Menge Konventionen und Standards unterliegt, die eingehalten werden müssen, erachtet er Programmieren nicht als Kunst.

Hunt und Thomas (2001) vergleichen das Entwickeln von Software mit dem Malen eines Portraits durch einen Künstler. In diesem Prozess stoßen Softwareentwickler genauso wie der Maler auf vergleichbare Hürden: Wie fange ich an? Wann bin ich fertig? Wie erfülle ich die Wünsche des Auftraggebers? Durch ihre interdisziplinäre Betrachtungsweise empfehlen sie Lösungsmöglichkeiten, die der Kunst-Perspektive entstammen, aber in der Softwareentwicklung hilfreich sein können.

Auch Brooks (1995) beschreibt Programmieren als Kunst und zieht Parallelen zwischen Programmierern und Schriftstellern, wobei Software als einfach zu handhabendes Material für den Künstler beschrieben wird:

*The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures. (Brooks 1995, S. 7)*

Lippman (2006) vergleicht seine Programmierfähigkeit mit seinem langjährigen Hobby des Schreibens von Prosa. In Anlehnung an John Markoffs Feststellung<sup>56</sup>, dass das Ziel der ersten Informatikvisionäre – eines Tages sollte jede Person einen Computer besitzen – inzwischen erreicht sei, führt er aus, dass dies nicht genügen kann.

*Das Ziel wurde nur halb erreicht. Was für einen Nutzen hat all diese Leistungskraft von Computern für eine Person, wenn diese Person einen Computer nicht selbst programmieren kann? Der Traum ist noch nicht wahr geworden, weil wir noch nicht herausgefunden haben, wie wir Programmieren zu einer Kunst machen können, so dass alle daran teilhaben können. (Lippman 2006)*

Dies lässt sich auch als Aufforderung an die Informatikdidaktik verstehen.

### **Programmier-Kunst**

Eine weitere Sichtweise, Programmieren als Kunst zu betrachten, ist das Programmieren, um *Kunstwerke* zu erschaffen. Ziel solcher Kunstwerke ist es, vergleichbar z. B. mit Michelangelos Sixtinischer Kapelle, eine künstlerische Herausforderung zu meistern (vgl. Knuth 2001). Bspw.

---

<sup>51</sup> Guido van Rossum ist Entwickler der Programmiersprache Python.

<sup>52</sup> Bjarne Stroustrup ist Entwickler der Programmiersprache C++.

<sup>53</sup> Paul Graham ist Lisp-Programmierer, Gründer verschiedener Softwareunternehmen und Autor des Lisp-Standardlehrwerks ANSI Common Lisp.

<sup>54</sup> Richard Stallman ist Gründer des GNU Projekts und der Free Software Foundation.

<sup>55</sup> Erik de Castro Lopo ist unter anderem Coauthor des Buchs “C for Linux Programming in 21 Days” und Entwickler der Linux Audio Library libsndfile.

<sup>56</sup> Vgl. Markoff (2005).

gehören dazu das Programmieren von Programm-Einzeilern, das Entwerfen von DEMOs<sup>57</sup> oder der Versuch, Programme zu schreiben, die wiederum eine Kopie ihrer selbst erzeugen. Solche Software hat in der Regel keinen bestimmten Zweck (was Software normalerweise hat), außer den des *künstlerischen Ausdrucks* oder der Demonstration technischen Könnens. Allerdings erfahren solche Meisterwerke in der Regel auch nur Anerkennung von Personen, die der entsprechenden Szene nahe stehen.

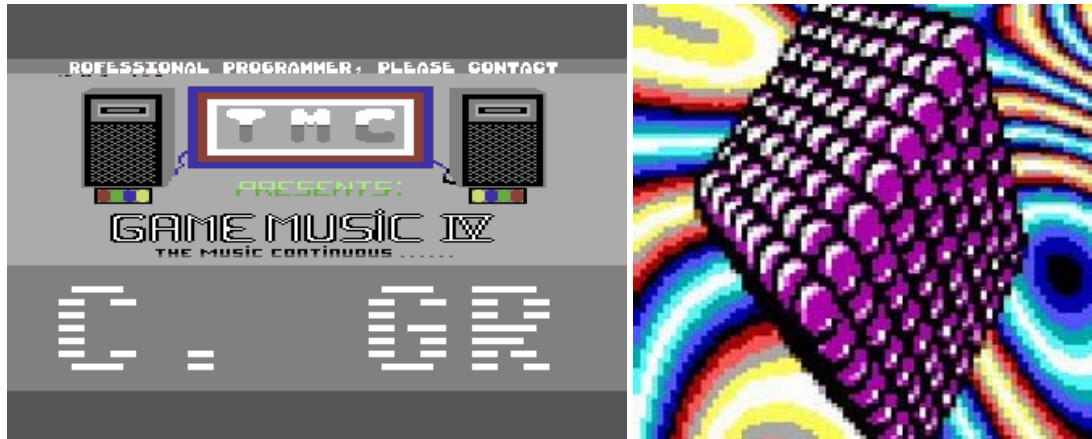


Abbildung 17: Kunstwerke aus der DEMO-Szene<sup>58</sup>.

Dass ein Programmtext auch durchaus poetisch sein kann, demonstriert ein syntaktisch korrektes Perl-Programm, welches es durch seine Poesie sogar zu Anerkennung außerhalb der Fachwelt geschafft hat (vgl. Abbildung 18).

Verschiedene Festivals bieten inzwischen Podien für Programmierkünstler, so z. B. die Transmediale Berlin oder das „Read\_ME 1.2 international media art festival“.

Kreatives Tun geschieht nicht immer zweckgebunden, sondern kann auch allein aus der Freude an der Betätigung motiviert sein. Ein Beispiel beschreiben Blackwell und Collins (2005): *Live Coding* – Programmieren von Laptopmusik als faszinierende Möglichkeit im end-user programming. Eine solche Änderung des Charakters der Programmierstätigkeit und Programmierumgebung werden als interessante kreative Perspektive für die Informatik beschrieben.

<sup>57</sup> Mit der DEMO-Szene hat sich eine interessante Programmierszene herausgebildet, die hochgradig spezielle Software entwickelt, die den Namen Kunstwerk verdient. Ziel dieser Programmierer ist es, mit möglichst wenigen Ressourcen beeindruckende Kunstwerke (Bilder, Animationen, Musik) zu erschaffen.

<sup>58</sup> Links: Game Music IV on the Commodore 64 by Charles Deenen, Rechts: PC text mode demo: Bolognese von Alpha Design. Quelle: <http://en.wikipedia.org/wiki/Demoscene>.

```

APPEAL:

listen (please, please);

open yourself, wide;
  join (you, me),
connect (us,together),

tell me.

do something if distressed;

  @dawn, dance;
  @evening, sing;
  read (books,$poems,stories) until peaceful;
  study if able;

  write me if-you-please;

sort your feelings, reset goals, seek (friends, family, anyone);

  do*not*die (like this)
  if sin abounds;

keys (hidden), open (locks, doors), tell secrets;
do not I-beg-you, close them, yet.

                                accept (yourself, changes),
                                bind (grief, despair);

require truth, goodness if-you-will, each moment;

select (always), length(of-days)

```

Abbildung 18: „Listen“ (Hopkins 1995; zitiert nach Bond 2005).

### 3.3.2 Künstler als Informatiker

In viel stärkerem Maß noch, als sich Programmierer damit beschäftigen, Kunstwerke zu erschaffen, haben Künstler die Methoden und Werkzeuge der Informatik für sich entdeckt. Die kreative Entwicklung von Kunst ist schon seit je her eng verbunden mit den verfügbaren Werkzeugen und Techniken. Neue Werkzeuge eröffnen dem kreativen Künstler neue Möglichkeiten, sich auszudrücken oder gar neue Kunstformen zu erschaffen. Ein typisches Beispiel aus der Musik ist die Erfindung des Klaviers, welches durch Johann Sebastian Bach in seinem Werk „Das Wohltemperierte Klavier“ systematisch erforscht und mit seinen Besonderheiten und Neuheiten vorgestellt wird. Ähnliche Beispiele sind auch in der Malerei und in anderen Künsten zu finden, wobei ein Werkzeug auch im immateriellen Sinne als Idee oder Technik verstanden werden kann.

Die Informatik hat in ihrer kurzen Entwicklungszeit bereits eine beachtliche Menge an Werkzeugen und Techniken hervorgebracht, die Künstler nur allzu gern aufgreifen und als künstlerisches Mittel einsetzen. Mit der Einführung von 3D-Druckern wird bald auch digitales Bildhauen möglich sein.

Beachtlich ist in diesem Zusammenhang, dass die neuen, digitalen Werkzeuge nicht nur für etablierte Künstler oder Informatiker ein neues Betätigungsfeld darstellen. Darüber hinaus – ja sogar in viel stärkerem Maße – wurde vielen neuen Menschen eine Betätigung in den digitalen Künsten ermöglicht, die nahezu alle etablierten Künste berührt und auch Amateuren neue Gebiete eröffnet (vgl. Wands 2002). Hierzu gehören Digitale Fotokunst/Fotografie, Webdesign, Musik-Bearbeitung/Sampling/Mixing, das Entwerfen von Kleidung im Internet, Videoproduktion, Audioproduktion, 2D- und 3D-Animation und Design und Typographie, um nur einige Beispiele zu nennen.

Maeda (2004) sieht Computer und Computertechnologie nicht nur als Werkzeug, sondern vielmehr als „neues Material“, welches künstlerisch verarbeitet werden kann. Dieses „neue Material“ wird von Künstlern und Designern begeistert aufgenommen und zur Schaffung neuer und besonderer Werke eingesetzt. Inzwischen hat sich in diesem Bereich eine Vielzahl von Strömungen entwickelt, die unter Bezeichnungen wie Digitale oder Digitale Mediale Kunst vereint werden.



Abbildung 19: Interaktive Kunst nach Pollock (vgl. Manetas 2003).

Neben den neuen Medien und Werkzeugen, die Künstlern – dank Informatik – zur Verfügung stehen, bedienen Sie sich auch verschiedener Techniken der Informatik (vgl. Maeda 2004; Trogemann und Viehoff 2005). Vor allem durch Programmier Techniken lassen sich in Kunst und Musik Werke erschaffen, die sonst nicht denkbar wären. Dreifus (2002) beschreibt ein solches Beispiel:

*In another experiment, John opened up one of his sugar packets and wrote a computer program that was able to count exactly how many crystals there were in one sweet little sack. The number was in the neighbourhood of 70,000, something he found “amazing.” After magnifying and arranging 70,000 sugar grains, Maeda created “Tea for Two,” a piece that can’t be termed “abstract,” because it is reality. (Dreifus 2002, S. 7)*

Eine Kategorisierung von Digitaler Kunst ist schwierig, wenn nicht unmöglich. Vielmehr überschreiten Künstler oft mit ihren Arbeiten die Grenzen verschiedener Medien und Verbinden unterschiedlichste Elemente miteinander (Paul 2003; Klütsch 2007). Ein Versuch, dennoch Kategorien zu bilden, die zudem einen interessanten Überblick über die vielfältigen Erscheinungsformen digitaler Kunst bieten, findet sich in der deutschsprachigen Wikipedia. So ist demnach innerhalb der Digitalen Kunst zu unterscheiden in Computergenerierte Digitale Kunst und Mediale Digitale Kunst (vgl. Abbildung 20). Die Widersprüchlichkeit des Versuchs zeigt sich allerdings an der langen und kontrovers geführten Diskussion zum Artikel.

<b>Computergenerierte Digitale Kunst</b>	<b>Mediale Digitale Kunst</b>
<ul style="list-style-type: none"> <li>• Bildkunst (z. B. Digitales Malen, Fotomanipulation, 3D-Kunst, Mathematische Kunst)</li> <li>• Grafik</li> <li>• Poesie</li> <li>• Musik</li> </ul>	<ul style="list-style-type: none"> <li>• Netzkunst</li> <li>• Softwarekunst</li> <li>• Code Poetry/Net Poetry</li> <li>• Interaktive Installationen</li> <li>• Virtuelle Realität</li> </ul>

Abbildung 20: Ausprägungen Digitaler Kunst laut Wikipedia (2008).

Vor allem der Bereich der Computergraphik ist ein Themenbereich, der in der Schulinformatik bereits häufig thematisiert wird, meist im Zusammenhang mit Algorithmik. Ich denke, die inzwischen riesige Palette von Ideen und Betätigungsmöglichkeiten in der Digitalen Kunst bietet vielfältige motivierende Anknüpfungspunkte, um im Zusammenhang mit Kompetenzen, die im Informatikunterricht vermittelt werden sollen, thematisiert zu werden.

### 3.3.3 Symbiose von Informatik und Kunst

Zunehmend entdecken Kunsthochschulen, dass Methoden und Kenntnisse der Informatik für Künstler nützlich, wenn nicht unabdingbar sind (vgl. Davis und Kundert-Gibbs 2006). Andererseits wird der Aspekt der „künstlerisch-kreativen Informatik“ teilweise auch in der informatischen Bildung genutzt, z. B. durch künstlerische Aufgaben oder das spezielle Einbeziehen von Kunst beim Programmieren (Ursyn und Sung 2007; Ursyn und Scott 2007). Ursyn und Scott (2007) integrierten Informatik und Kunst in einen Informatik-Anfangskurs, um sowohl das Fachverständnis zum Programmieren als auch das Verständnis von Kunst zu verbessern. Schwerpunkte waren die Anwendung von Kunst im Programmierkurs und ein interdisziplinäres Arbeiten der Studenten. Das Ziel, ein besseres Verständnis für Programmieren durch die Beschäftigung mit Kunst und durch expressives Denken zu erreichen, wurde in diesem Kurs erreicht. Zu den Aufgaben des Kurses gehörten u. a. die Konzeption und Erstellung eines „Roboter Makers“, der nach veränderbaren Variablen der Nutzer roboterartige Gebilde konstruiert, und die künstlerische Visualisierung einer abstrakten Klasse (siehe Abbildung 21).

Turner, Weakley et al. (2005) berichten von einer fruchtbaren Zusammenarbeit von Programmierern und Künstlern bei der Erschaffung von Kunstwerken, in welcher Programmierer die Schnittstelle zwischen Künstler und Computer übernehmen. Sowohl Künstler als auch Programmierer profitierten von diesem Projekt gleichermaßen.

Im Kontext der Schulinformatik sind aus dieser Perspektive verschiedenste Projekte denkbar, die Informatik und Kunst vereinen.

### 3.3.4 Kritik

Rechenberg schreibt: „*Der Rechner ist ein viel zu kompliziertes Gebilde, der Zugang zu ihm ist viel zu umständlich und indirekt, und er ist schließlich bei aller Flexibilität viel zu starr, um als Material für Kunstwerke dienen zu können*“ (Rechenberg 2000, S. 287).

Personen, die den mystischen Charakter in ihrem Kunstverständnis nachgeben, äußern sich kritisch über den Beitrag von Informatik für die Kunst. „*Es fehlt ihnen der Mensch, der dahinter steht, die Schöpfungskraft des Künstlers, die uns ein Kunstwerk bewundern und immer wieder zu ihm zurückkehren lässt, auf dass wir neue Kraft, Belehrung und Anregung zu eigenem Tun von ihm empfangen*“ (Rechenberg 2000, S. 287).

Gerade hierbei scheint uns die Entwicklung eines besseren zu belehren. Ist auch die Entwicklung (noch? Vgl. Abschnitt 3.5) nicht so weit, selbständig kreative Leistungen hervorzubringen, so können Informatiksysteme den Menschen in vielfältigster Weise bei kreativen Tätigkeiten unterstützen und dienen Künstlern durchaus als ergiebiges Material. Diese Sichtweise auf Kreativität und Informatik werde ich im Folgenden darstellen.









```
#include <iostream.h>
class  {
private:
 ;
 ;
public:
 () {
void  (int) ;
void  () ;
int  () ;
int  () ;
};
```

Abbildung 21: Visualisierung der Klasse „Cow“ (Ursyn und Scott 2007).

### 3.4 Technologie-Perspektive<sup>59</sup>

Computer werden von Wissenschaftlern und Künstlern zunehmend als wertvolle Werkzeuge für kreative Tätigkeiten angesehen. Auch ein großer Teil der Forschung im Spannungsbereich von Kreativität und Informatik beschäftigt sich mit den Möglichkeiten zur Förderung von Kreativität durch IKT in professionellen Bereichen und Lernumgebungen (Shneiderman, Fischer et al. 2005). Diese beschränken sich nicht allein auf Möglichkeiten der Förderung von Kreativität innerhalb der Informatik, sondern erstrecken sich auf unzählige Gebiete, auf denen Software kreativitätsunterstützend eingesetzt werden kann. So sieht Trogemann (2001) die Aufgabe und Herausforderung der Informatik insbesondere darin, neue Kreativ-Werkzeuge bereitzustellen. Shneiderman (2007) prophezeit:

*The current and forthcoming generations of programming, simulation, information visualization, and other tools are empowering engineers and scientists just as animation and music composition tools have invigorated filmmakers and musicians. (S. 20)*

Das impliziert eine steigende Bedeutung kreativitätsunterstützender Software und bedeutet, dass Erfindungen und Innovationen, angeregt durch kreativitätsunterstützende Werkzeuge, in noch größerem Maß als zuvor zunehmen werden. Sie ermöglichen selbst Anfängern, wie Experten zu arbeiten, und Experten, immer beeindruckendere Ergebnisse in vielfältigen Bereichen zu produzieren.

Shneiderman (2002) postuliert in einem Kreativitätsframework vier Aktivitäten, die durch IKT unterstützt werden können: das Sammeln und Darstellen von Information, das in Beziehung setzen zu vorhandenem Wissen, z. B. durch Unterstützung der Kommunikation, das Schaffen und Evaluieren neuer Lösungen sowie das Verteilen der Ergebnisse (vgl. Abbildung 22). Während diese Aktivitäten die potentiellen Möglichkeiten von IKT noch relativ abstrakt beschreiben, konkretisiert Shneiderman (2002) die Anforderungen durch Kriterien für kreativitätsunterstützende Softwarewerkzeuge.<sup>60</sup>

*IT should support*

- *pain-free exploration and experimentation*
- *immediate and useful feedback for one's actions*
- *no big penalties for mistakes, meaningful reward for success*
- *easy way to undo and redo*
- *visualizing data processes*
- *searching for knowledge and inspiration*
- *composing a work step by step*
- *disseminating results to gain recognition.*

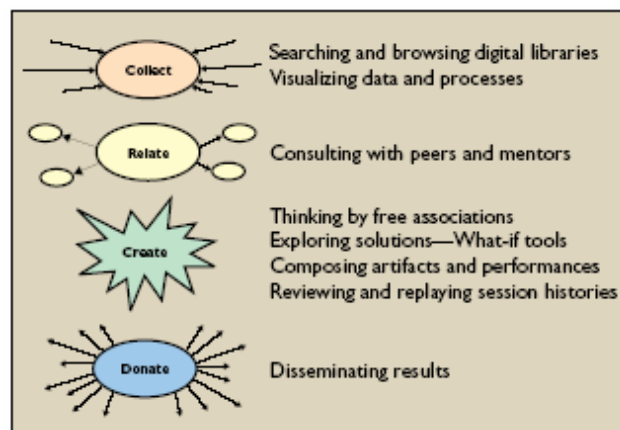


Abbildung 22: Vier Aktivitäten und Aufgaben im Kreativitäts-Framework (Shneiderman 2002).

Diese Anforderungen lassen sich direkt auf Software beziehen, die in der Informatik eingesetzt wird. Computerbasiertes Experimentieren und Explorieren kann zu neuen Ideen verhelfen und sollte Designen, Simulation, auch Überraschungen und die Erzeugung von Artefakten ermöglichen (Mitchell, Inouye et al. 2003). Programmierer nutzen diese Möglichkeiten des Experimentierens, die ihnen Programmierumgebungen eröffnen, indem sie Ideen und mögliche Lösungswege austesten. Nach Glass (2006) stellt das Anwenden von selektivem Trial und Error einen wesentlichen Teil der Softwareentwicklung dar. Solche „Was passiert wenn“-Szenarien können schnell auspro-

<sup>59</sup> Technologien meint in diesem Zusammenhang Informations- und Kommunikationstechnologien.

<sup>60</sup> Vergleichbare Anforderungen stellen (Greene 2002; Resnick, Myers et al. 2005; Loveless 2007; Lubart 2005).

biert und variiert werden und sind nur durch die Unterstützung der Entwicklungsumgebung möglich. Programmierumgebungen für Programmieranfänger, wie z. B. Scratch, machen Explorieren sogar noch leichter; Operationen können einfach auf Objekten ausgeführt und das Ergebnis in Echtzeit beobachtet werden. Offensichtlich beinhaltet auch das Programmieren-Lernen das Erkunden der Programmierumgebung („Was sind meine Möglichkeiten?“) und das Experimentieren mit den Objekten und Konstrukten, die anwendbar sind („Was passiert, wenn ich dieses so verwende?“).

Ein sofortiges und nützliches Feedback auf ausgeführte Aktionen durch den Compiler oder Interpreter unterstützt das Experimentieren und ist eine Besonderheit der Softwareentwicklung. In anderen Bereichen geben die Werkzeuge, mit denen gearbeitet wird, entweder kein qualitatives Feedback automatisch oder können nicht auf Fehler hinweisen.<sup>61</sup> Bei vielen Schülern kann man beobachten, wie solches handlungsbasiertes Feedback<sup>62</sup> ein Gefühl von eigener Kontrolle und selbstreguliertem Lernen ermöglicht.<sup>63</sup>

Die von Shneiderman geforderte „Straffreiheit“ wird in Softwareentwicklungsumgebungen insofern realisiert, als dass das Fehlerfeedback die größte „Strafe“ darstellt. Im Gegensatz zu anderen Unterrichtsfächern, wo Feedback nur durch den Lehrer oder durch andere Schüler möglich ist, wird dieses beim Programmieren durch eine Maschine in einer nicht einschüchternden Art und Weise übernommen. Ein funktionierendes Programm wartet als Belohnung für einen erfolgreichen Programmierversuch; in Scratch wäre dies bspw. eine Animation oder ein Spiel. Unterstützend wirkt sich beim Experimentieren die Sicherheit aus, dass nichts wirklich „kaputt“ gemacht werden kann. So ermöglicht Scratch auf einfachem Weg, letzte Änderungen rückgängig zu machen oder wiederherzustellen. Während diese genannten Punkte kreativen Arbeitens in anderen Fächern wenn dann nur an Simulationen möglich sind, wird beim Programmieren am Produkt experimentiert.

Viele Programmier(lern)umgebungen erfüllen auch weitere von Shneidermans Kriterien, z. B. die Visualisierung von Prozessen und Daten, welche dabei hilft, Ideen und Fakten zu organisieren und Beziehungen zu entdecken. Der Trend zu visueller Programmierung spiegelt diesen kreativitätsrelevanten Aspekt wider. In Scratch werden Programmierkonstrukte (z. B. Schleifen, Operationen, Botschaften) durch bunte Bausteine dargestellt, welche die unterschiedlichen Bausteintypen durch verschiedene Farben kategorisieren. In modellierungsorientierten Ansätzen werden Datenstrukturen und Prozesse als Diagramme mit ihren Beziehungen modelliert und können automatisch in Programmtext übersetzt werden und vice versa (z. B. mit Fujaba). Der Trend zu diesem visuellen, abstrakteren Level der Programmierung verspricht, dass dies in Zukunft wichtiger, und damit auch Kreativität noch besser unterstützt wird.

In einer kreativitätsunterstützenden Umgebung ist es wichtig, Wissen einfach zugänglich zu machen und für Inspiration zu sorgen. Programmierumgebungen ermöglichen das Suchen in einem Hilfesystem und beinhalten häufig eine Dokumentation mit Kommentaren und Beispielen. Scratch bringt zusätzlich inspirierende Beispielprogramme mit, die von anderen Schülern programmiert wurden, und zusätzliche Materialien, an welchen Programmierkonstrukte erklärt und Experimentieren mit diesen Konstrukten angeregt werden.

Schrittweises Komponieren bedeutet ein langsames Herangehen an ein Problem, während noch viele Entscheidungen offen gelassen werden. Für kreatives Arbeiten ist diese Bottom-Up-Vorgehensweise wichtig, da ein Ergebnis häufig nicht klar von Anfang an bestimmt ist und sich erst später konkretisiert. In Scratch ist diese Möglichkeit gegeben, indem ein Programm zu jedem Zeitpunkt ausgeführt und überprüft werden kann.

Ein Verbreiten der Ergebnisse kann sich positiv auf die Motivation und die eigene Reputation auswirken. Ein Klassenraum bietet zwar nur wenige Möglichkeiten, eigene Leistungen zu präsen-

---

<sup>61</sup> So müssen bspw. musikalische oder Kunstexperimente durch andere bewertet werden. Nicht-Softwarebasierte Experimente in Physik oder Chemie geben im Fall des Misserfolgs nicht an, wo die Probleme liegen.

<sup>62</sup> Wie in Kapitel 2 festgestellt wurde, sollte Feedback nur informierenden, nicht kontrollierenden Charakter haben.

<sup>63</sup> Programmieren wird häufig sogar autodidaktisch gelernt.

tieren, durch das Internet können diese aber einem breiten Publikum vorgestellt werden. Scratch besitzt eine Funktion, mit der Scratch-Programme einfach auf einen Internet-Server geladen werden können. Dort können sie von anderen Lernenden eingesehen und bewertet werden und dienen ebenso zur Inspiration für andere.<sup>64</sup>

Abschließend betrachtet erfüllt Scratch, so wie viele andere Programmierlernumgebungen, Shneidermans Kriterien für kreativitätsunterstützende Software.

Software kann durchaus auch un kreativ eingesetzt werden, was bei der Betrachtung insbesondere älterer „Lern-Software“, welche vor allem einem Drill-and-Practice Ansatz folgte, deutlich wird. So warnt Hartmann (2004), dass Computer und Internet allein im Unterricht kaum zu mehr Kreativität und Effizienz führen werden:

*Gefragt ist eine effiziente und effektive Nutzung dieser Werkzeuge, für eine kreative und inspirierende Schulumgebung bleiben aber weiterhin die Lehrpersonen verantwortlich.*

IKT fördern Kreativität in der Informatik so, wie Musikinstrumente Kreativität in der Musik beflügeln haben. Dabei sind IKT im Unterricht nicht als „Wundermittel“ zu verstehen, deren alleinige Existenz für die Entwicklung von Kreativität garantiert. Die sorgfältige Auswahl des Werkzeugs und die passenden Aufgabenstellungen sind hierfür grundlegend. Dies gilt insbesondere für andere Unterrichtsfächer. Werden vom Informatiklehrer die kreativitätsrelevanten Anforderungen an die Werkzeuge berücksichtigt, können diese allerdings gerade eine Teil der kreativen und inspirierenden Schulumgebung ausmachen. Nicht zuletzt weil Computer einen festen Bestandteil der Lernumgebung von Informatikunterricht darstellen, sollte auf einen kreativen Einsatz der Technik geachtet werden.

### 3.5 Artificielle Kreativität

Ziel verschiedener informatischer Bestrebungen ist es, Systeme zu konstruieren, welche Kreativität nachbilden. Die Vorstellung, dass Computer eigenständig etwas Schöpferisches hervorbringen, mag manchem absurd vorkommen: So kann ein Computer doch nur das hervorbringen, was ihm zuvor einprogrammiert wurde. Es mag umstritten sein, ob Computerleistungen als kreativ oder schöpferisch zu bewerten sind. Zumindest können Computer Dinge tun, die schöpferisch wirken.<sup>65</sup> Boden stellt in diesem Zusammenhang vier zentrale Fragen (vgl. Boden 1995, S. 18):

1. Können computerwissenschaftliche Vorstellungen beim *Verständnis* menschlicher Kreativität helfen?
2. Können Computer jemals etwas zu tun, das schöpferisch *wirkt*?
3. Kann ein Computer menschliche Kreativität *erkennen*?
4. Kann ein Computer jemals *tatsächlich kreativ* sein (im Gegensatz zu reiner Produktion scheinbar kreativer Leistungen, deren Originalität ausschließlich auf den menschlichen Programmierer zurückzuführen ist)?

Die ersten drei Fragen können inzwischen sicherlich mit „ja“ beantwortet werden, bleiben aber in ihrer Bedeutung nicht ohne Brisanz. Für die Diskussion der vierten Frage können verschiedenartigste Projekte herangezogen werden. Eines der bekanntesten und erfolgreichsten Projekte im Bereich artifizierlicher Kreativität ist das Programm Aaron (Cohen 1995). Die Software generiert originäre komplexe Malereien und Zeichnungen, welche durchaus mit Werken versierter Künstler vergleichbar und in verschiedenen Kunstausstellungen zu sehen sind. Dargestellt werden häufig menschliche Figuren, die vor verschiedenen Hintergründen bei Aktivitäten zu sehen sind. Hofstadter (1996) charakterisiert die Zeichnungen gar als „*oftmals lustig und von einer bezaubernden*

<sup>64</sup> So wie die einfache Verbreitungsmöglichkeit von Amateur-Geschichten zu einer steigenden Anzahl kreativer Literatur im Internet geführt hat, kann auch das Verbreiten von Software durch das Internet zu kreativem Tun anregen.

<sup>65</sup> Nach der Feststellung, dass es inzwischen Computer gibt, die ihre eigenen Programmierer in Schach oder Dame schlagen können, resümieren Lefrancois und Leppmann (1994): „Es ist also möglich, Computer kreativ zu programmieren“ (S. 142). Dass Computer etwas Neues schaffen und erfinden, bezeichnen sie als Herausforderung der Künstlichen Intelligenz.

*Naivität, die ein unverkennbares Stilmerkmal darstellt*“ (S. 522). Cohen selbst bezeichnet das Programm allerdings nicht als kreativ, zumal er mit dem Wort sehr vorsichtig umgeht (vgl. Cohen 1999a, b).

Bringsjord und Ferrucci (1999) berichten über den Versuch, einen Computer in den urkreativen Bereich menschlicher Kreativität vordringen zu lassen: dem Geschichtschreiben, welches zu einem gewissen Grad heute bereits möglich ist. So hat das Computerprogramm Racter (vgl. Racter 1984) bereits „eigene“ Bücher geschrieben, wobei einzuschränken ist, dass in nicht unerheblichem Maße hierbei von Menschenhand „nachgeholfen“ wurde. In akademischen Kreisen wurde der Paper-Generator populär, welcher eigenständig pseudowissenschaftliche Artikel generiert, von denen nach Berichten sogar verschiedene Artikel auf Konferenzen oder in Journals akzeptiert wurden (vgl. Stribling, Krohn und Aguayo 2005). Das Programm „AM-Automated Mathematician“ ist ein Beispiel für wissenschaftliche Entdeckungen, die durch Software geleistet wurden.<sup>66</sup> Weitere Betrachtungen zur Entwicklung kreativer Systeme finden sich u. a. bei VanLangen, Wijngaards et al. (2004).

In der Schulinformatik eröffnet die Diskussion von Bodens Fragen Anreize, über die Grenzen der Computerisierung nachzudenken oder sogar Versuche zu unternehmen, z. B. durch geschicktes Einbinden von Zufallselementen, Ansätze artifizierlicher Musik oder Malerei aktiv nachzuvollziehen. Dies liefert zudem auch Anstöße, über die Strukturen menschlicher Kreativität nachzudenken.

### **3.6 Situierung der Sichtweisen im Informatikunterricht**

Die dargestellten Sichtweisen verdeutlichen, dass Kreativität in der Informatik nicht nur existent, sondern sogar für das Fachverständnis, für den Softwareentwicklungsprozess und für Informatiker selbst bedeutsam ist. Darüber hinaus eröffnet die Informatik durch ihre Methoden, Werkzeuge und durch die aus ihr hervorgegangenen Produkte einem breiten Spektrum an Anwendern kreative Betätigungsmöglichkeiten.

Die Sichtweisen können helfen, die Diskussion um Kreativität in der Informatik zu präzisieren und damit produktiver zu gestalten. So können bspw. die Beiträge im LOG IN-Themenheft „Computer, Kreativität und Ästhetik“ (LOG IN 1995) der Systematisierung entsprechend der Kunst-Perspektive zugeordnet werden.

Für einen kreativen Informatikunterricht besitzen die Sichtweisen sowohl thematische als auch inhaltliche Relevanz. Ziel kreativen Lernens im Informatikunterricht sollte es sein, Kreativität in der Informatik aus der Fach-Perspektive inhaltlich zu verdeutlichen. Damit erfolgt kreatives Lernen im Informatikunterricht nicht nur nah am Fach, sondern nutzt auch die immanenten Kreativitätsfaktoren, welche die Informatik bietet. Zu prüfen ist, ob sich die in der Mensch-Perspektive dargestellten motivationalen Vorteile ebenfalls bei Schülern erreichen lassen und ob sich eine kreative Beschäftigung mit Informatik im Unterricht zusätzlich positiv auf ihr Bild von der Informatik auswirkt.

Anwenden der in der Technologie-Perspektive genannten Aspekte ist eine methodische, Anwenden der Themen der Kunst-Perspektive und der Perspektive der artifizierlichen Kreativität sind thematische Möglichkeiten, sich der Fach-Perspektive anzunähern. Gegenüber der systematischen, theoretischen, technischen oder historischen Betrachtung ist die Beschäftigung mit den kreativen Seiten der Informatik für die Schulinformatik äußerst attraktiv. Ausgehend vom Konsens, dass Kreativität eine Eigenschaft ist, die im Informatikunterricht sowohl förderungswürdig ist als auch die Attraktivität des Fachs erhöht, ergeben sich hieraus zahlreiche Anknüpfungsmöglichkeiten, anhand derer Unterrichtsinhalte thematisch verankert werden können und damit Kreativität in der Informatik erfahrbar gemacht werden kann.

Die Charakterisierung erhebt keinen Anspruch auf Vollständigkeit – weitaus mehr Sichtweisen

---

<sup>66</sup> AM wieder-entdeckte die Goldbachsche Vermutung.

können eingenommen werden, dieses spannende Interaktionsfeld zu betrachten und anzuwenden.<sup>67</sup> Für die weitere Diskussion um Kreativität in der Informatik können sie durchaus eine Grundlage legen, die verschiedenen Verständnisse von Kreativität in Äußerungen und Argumenten einzuordnen und zu bewerten.

In dieser Arbeit wird Kreativität in der Informatik im Folgenden von der Fach-Perspektive betrachtet.

Zusammenfassend bezeichnet Kreativität aus der Fachsicht der Informatik in dieser Arbeit damit sich *aus informatischen Prozessen* (wie Programmieren, Modellieren) *ergebene* Leistungen, die zu persönlich neuen und verwendbaren Ideen, Produkten, Lösungen oder Erkenntnissen führen.

---

<sup>67</sup> Z. B. „Hacking“ als kreativer Umgang mit Technik und Informatik (vgl. Graham 2004; Schell und Martin 2006) oder Creative Gaming. Selbst eine theologische Betrachtung von Informatik und Kreativität ist möglich (vgl. Garner 2005).

## 4 Kreativität in der informatischen Bildung

Während im vorherigen Kapitel ein auf die Fach-Perspektive bezogenes Kreativitätsverständnis konkretisiert wurde, werden in diesem Kapitel die Chancen, die sich aus der Berücksichtigung dieser Kreativität in der informatischen Bildung ergeben, im Hinblick auf den Informatikunterricht betrachtet. Während in Deutschland nur wenige Arbeiten der Fachdidaktik Kreativität thematisieren, verdeutlichen verschiedene Forschungsergebnisse der internationalen fachdidaktischen Diskussion das Potential, das sich durch die Berücksichtigung von Kreativität ergibt.

Als erstes wird die Problematik der Schülerwahrnehmung von Kreativität im Fach Informatik erörtert. Anschließend werden Aussagen und Forschungsergebnisse der internationalen Forschung sowie kreativitätsverwandte Projekte in der informatischen Bildung analysiert und vor dem Hintergrund der bis hierher geführten Diskussion bewertet. Das Kapitel schließt mit Überlegungen zu Ursachen der mangelnden Berücksichtigung von Kreativität im Informatikunterricht und einem Vorschlag zur Lösung dieses Problems.

### 4.1 Zur Wahrnehmung von Kreativität in der Informatik durch Schüler

Die Sichtweise auf ein Schulfach prägt die Erwartungen, den Anspruch, aber auch die Bemühungen, die einem Fach entgegengebracht werden. Informatik wird in der Öffentlichkeit nicht als kreatives Fach wahrgenommen (vgl. Kapitel 3, Abschnitt 3.1.1), und auch viele Schüler vertreten die Auffassung, dass Informatik per se „nicht kreativ“ sei (bspw. in Antonitsch, Kramer et al. 2007). Maaß und Wiesner (2006) stellen fest, dass sich Informatik für viele Schüler auf „Programmieren, Mathe und ein bisschen Hardware“ zu beschränken scheint. Dabei wird dieses Bild weder den kreativen Möglichkeiten der Informatik, dem Informatikstudium noch den Tätigkeiten eines Informatikers gerecht. Gefordert wird, ein facettenreiches Bild der Informatik zu vermitteln (vgl. Hengl 2008).

Offensichtlich bleibt vielen Schülern, denen es nicht vergönnt ist, Informatikunterricht zu besuchen, das kreative Potential der Informatik verborgen. Bei Schülern, die Informatikunterricht besuchen, ist die Wahrnehmung des Fachs offenbar recht unterschiedlich. Ermutigende Feststellungen machte Modrow (2002). In einer Umfrage bat er die Informatikschüler, ihre eigene Kreativität in und außerhalb des Informatikunterrichts einzuschätzen.<sup>68</sup> Im Vergleich mit den Sprachkursen schätzen die Informatikschüler sich selbst, ebenso wie die „Naturwissenschaftler“, als deutlich weniger kreativ ein. Dieses Ergebnis kehrt sich allerdings bei der Betrachtung der Kreativität im Fach um. 81% der Schüler bezeichnen das, was sie im Fach Informatik tun, als kreativ. Offenbar bietet Informatikunterricht Schülern, die sich sonst nicht als kreativ einschätzen, die Möglichkeit, sich im Fach Informatik kreativ zu betätigen. Sollte dem so sein, wäre dies ein Beitrag, den Informatikunterricht zur Kreativitätsförderung liefern könnte (vgl. auch Kapitel 6).

Die Frage, welche Wahrnehmung Schüler von Kreativität im Informatikunterricht haben bzw. ob und inwiefern sich diese durch einen kreativen Informatikunterricht beeinflussen lässt, werden in den Kapiteln 6 und 9 als Gegenstand zweier empirischer Studien erneut aufgegriffen.

	IN	MA	PH	DE	EN
Kreativität allgemein	65 %	69 %	57 %	74 %	71 %
Kreativität im Fach	82 %	46 %	62 %	79 %	57 %
<b>Differenz: Fach – allgemein</b>	<b>18 %</b>	<b>-23 %</b>	<b>5 %</b>	<b>5 %</b>	<b>-14 %</b>

Abbildung 23: Einschätzung eigener Kreativität von Schülern (zitiert nach Modrow 2002, S. 111).

<sup>68</sup> Auch wenn es sich nicht um eine repräsentative Umfrage handelt, stellt sie doch ein positives Beispiel für die Wahrnehmung kreativen Tuns im Informatikunterricht dar.

## 4.2 Kreativität in der Forschung zur informatischen Bildung

In der internationalen Forschung<sup>69</sup> gibt es mehr oder weniger zaghafte Versuche, die Kreativität von Schülern und Studenten zu instrumentalisieren. Dennoch bleibt es erstaunlich, dass Kreativität auch international bisher kaum reflektiert ist. Sucht man nach dem Stichwort „*creativity*“ in der *ACM Digital Library* erhält man nur wenige Artikel mit Bezug zu Bildung und Didaktik. Diese Artikel lassen sich wenigen Gruppen in den Bereichen Problemlösen und Problemfinden, Motivation und Verbesserung der Unterrichtsqualität sowie der Untersuchung von Informatiksystemen zur Unterstützung kreativen Arbeitens zuordnen. Damit spiegeln die Gruppen die drei für die informatische Bildung wesentlichen Perspektiven von Kreativität wieder: Fach-, Mensch- und Technologie-Perspektive.

### Aussagen und Ergebnisse zur Fach-Perspektive

Scragg, Baldwin et al. (1994) argumentieren, dass Informatik grundsätzlich ein kreatives Unterfangen ist und Lernende demzufolge ermutigt werden sollten, Einsichten in die kreativen Problemlöseprozesse der Informatik zu erhalten. Hill (1998) beschreibt Problemlösen mit offenem Ende und Designprozesse in der informatischen Bildung als kreative Prozesse, die Exploration erfordern. Sie empfiehlt die Entwicklung lebensweltnaher Prototypen, statt auf der Modellebene stehen zu bleiben. Diese Vorschläge stehen im Gegensatz zu populären deutschen Ansätzen für die Schulinformatik, in welchen explizit nur das Entwerfen von Modellen, aber nicht deren Implementierung vorgeschlagen wird (vgl. Hubwieser 2000).

Einige Autoren richten ihr Augenmerk auf das Gebiet des Problemfindens bzw. Identifizierens von Problemen. Diese Tätigkeiten erfordern Kreativität und sind wichtig in der Informatik. Im Unterrichtskontext beschränkt sich das nicht auf das Finden komplett neuer Probleme, sondern beinhaltet auch das Umformulieren und Anpassen von gegebenen Problemen (vgl. Lewis, Petrina und Hile 1998; Kaasbøll 1998). Sutinen und Tarhio (2001) schlagen in diesem Zusammenhang vor, besser von „Problem-Management“ statt von Problemlösen zu sprechen, da das Erkennen und Formulieren von Problemen zu den grundlegenden Fertigkeiten in der Informatik zählen.

Gu und Tong (2004) fanden in einer empirischen Studie heraus, dass in Softwareentwicklungskursen Design und Programmierfähigkeiten von den Studenten als kreativ empfunden und diese Phasen bevorzugt wurden. Aus vergleichbaren Gründen setzen einige Autoren Kreativität in Informatikkursen bewusst als Mittel ein, Motivation und Interesse zu steigern. Dies wurde z. B. erreicht

- durch eine Veränderung der Lernumgebung, so dass „hands-on-learning“ und Exploration ermöglicht wurden (Lewandowski, Johnson und Goldweber 2005),
- indem die Studenten ihre eigenen Probleme auswählen und bearbeiten durften (Meisalo, Sutinen und Tarhio 1997) und
- indem Programmieren als Mittel persönlichen kreativen Gestaltens vorgestellt und motiviert wurde (z. B. Peppler und Kafai 2005; Resnick 2002).

Einige Wissenschaftler berichten darüber hinaus von positiven Auswirkungen auf die Leistungen von Lernenden, wenn Kreativitätstechniken in Informatikkursen angewandt wurden (Epstein 2006) oder wenn kreative Methoden beim Unterrichten verwendet wurden (Chaytor und Leung 2003).<sup>70</sup> So integriert Brunner (1991) kreativitätsunterstützende Aktivitäten in einem Programmierkurs mit dem Ergebnis, dass sich Stolz auf die eigene Arbeit, Ausdauer und Fleiß verbesserten.

<sup>69</sup> Vor allem in der Forschung im Bereich *computer science education*.

<sup>70</sup> Kreative Lehrmethoden implizieren nicht notwendigerweise eine Berücksichtigung der Kreativität auf der Schülerseite. Wie auch Biggs (1996) in seiner Theorie des „Constructive alignment“ anschaulich erläutert, hängt Lernerfolg vor allem von den Aktivitäten der Lernenden ab. Entsprechendes gilt für Kreativität. Aus diesem Grund wird das *kreative Unterrichten* der Informatik hier nicht weiter über die Tatsache hinaus betrachtet, dass die Berücksichtigung und Förderung von Kreativität beim Lehren auch als kreatives Unterrichten verstanden werden kann.

Neben diesen Versuchen, in denen Kreativität erfolgreich zur Verbesserung des Lernerfolgs eingesetzt wurde, fordern einige Wissenschaftler mehr Kreativität in der informatischen Bildung, da

- Absolventen häufig zu wenig Kreativität und Problemlösefähigkeiten besitzen (Mittermeir 2000),
- Kreativität im Curriculum unterrepräsentiert ist (Sweeney 2003),
- Frauen ihre Informatikausbildung abbrechen, weil sie in Informatikkursen nicht genug Raum für individuelle Kreativität finden (Guzdial und Soloway 2002) und
- Kreativität als die höchste Form von Bildung angesehen wird, auch im Bezug auf Computer Literacy (Van Dyke 1987).

### **Aussagen und Ergebnisse zur Technologie-Perspektive**

IKT werden als hilfreiche Werkzeuge zur Unterstützung von Kreativität angesehen. Viele wissenschaftliche Arbeiten beschäftigen sich entsprechend damit, was IKT zur Unterstützung kreativen Arbeitens leisten können. Dennoch gibt es nur wenig Artikel, welche die Möglichkeiten von IKT bezogen auf die Schulinformatik thematisieren. Wheeler, Waite et al. (2002) untersuchten den Einfluss von IKT auf kreatives Lernen in einer englischen Grundschule. Sie fanden heraus, dass Lehrer durch die Einbeziehung von Computern im Unterricht mehr offene Aufgaben stellten und diese variabelere Lernstile zuließen. Ebenso konnten unterschiedlichere Leistungsniveaus besser berücksichtigt werden. Clements (1995b) unterstreicht, dass IKT zwar zunehmend als Kreativitätswerkzeuge erfolgreich eingesetzt werden, diese aber genauso uncreative Lernansätze unterstützen können. Die Förderung von Kreativität hängt von der Art der eingesetzten Software und den Rahmenbedingungen, z. B. dem Curriculum ab.

Ein Verständnis von Konzepten der Informatik ist grundlegend wichtig für eine kreative Nutzung von IKT, auch in anderen Fächern (vgl. Computer Science and Telecommunications Board 1999). Im Bezug auf Programmierung und Kreativität wurde mehrfach der Einsatz von LOGO untersucht. Es konnte gezeigt werden, dass LOGO elementare Prozesse kreativer Leistungen unterstützt (Papert 1993; Clements 1995a).

### **Aussagen und Ergebnisse zur Mensch-Perspektive**

In den Betrachtungen zu Kreativität und Informatik konnte bereits festgestellt werden, dass intrinsische Motivation und Kreativität in einem sich wechselseitig bedingenden Verhältnis stehen und dass dies in Studien mit Programmierern bestätigt wurde. In einer Fallstudie zur Spieleprogrammierung in der informatischen Bildung wurde von Long (2007) die Motivation der Lernenden untersucht. „Die Möglichkeit, Probleme selbst zu lösen“ und „die Möglichkeit, kreativ zu sein“ waren in dieser Studie die wichtigsten Faktoren, welche intrinsische Motivation beeinflussten. Dies kann als Anhaltspunkt dafür dienen, dass auch in der informatischen Bildung das Gefühl und die Möglichkeit von Kreativität intrinsische Motivation fördern.

Da Grundstudiumskurse der Informatik, zumeist Einführungskurse in die Programmierung, oftmals nur schlechte Ergebnisse liefern, wurden bereits viele Untersuchungen durchgeführt, welche die Gründe für das schlechte Abschneiden der Studenten herauszufinden versuchten. Von folgenden Ergebnissen wird bzgl. Motivation berichtet:

- Hoch-intrinsisch motivierte Studenten brachten bessere Leistungen (Bergin und Reilly 2005)
- Viele Studenten waren an sich nicht grundsätzlich am Programmieren interessiert (Curzon und Rix 1998; Mamone 1992)
- Motivation konnte durch die Verwendung von bedeutsamen Aufgaben und Übungen gesteigert werden (Rich, Perry und Guzdial 2004; Tharp 1981; Feldgen und Clua 2003)
- Motivation konnte durch persönlich herausfordernde Aufgaben verbessert werden (Lawrence 2004)

Auch in deutschen Schulen wird Informatik oftmals als kompliziertes, technisches Fach aufgefasst. Auch hier führt die Programmierung als Unterrichtsinhalt oftmals zu Desinteresse, Motivationsverlust und schlechten Schülerleistungen. Bisher wurde offensichtlich noch nicht der optimale Weg gefunden, Schüler und Studenten an die Informatik heranzuführen. Ein Teil des Schlüssels zum Erfolg könnte Kreativität sein: Während im Unterrichtszusammenhang Programmieren oft nicht als motivierend und kreativ erlebt wird, lassen sich außerhalb des Unterrichts auch bei Schülern interessante Beobachtungen machen: Das Entwickeln von Software kann auch auf Informatikschüler bzw. -studenten einen faszinierenden Einfluss ausüben, ähnlich wie bei Open-Source-Programmierern (vgl. Kapitel 2, Abschnitt 2.3). Auch Schüler und Studenten investieren mitunter einen enormen Teil ihrer Freizeit zum Entwickeln von Software und lernen Programmiersprachen sogar autodidaktisch. Die von Lakhani und Wolf (2005) festgestellten kreativitätsverwandten Faktoren, die für die intrinsische Motivation und folglich für die Beteiligung an Softwareprojekten verantwortlich waren (beabsichtigter Gebrauch, Reputation, Identifizierung mit der Gruppe, Lernen), lassen auch für die Förderung von Motivation in der Schulinformatik Implikationen zu.

#### **Berücksichtigung des Gebrauchs:**

Ein offensichtlicher Grund, eine Software zu entwickeln, ist der Wunsch, diese später auch zu verwenden. Wenn ein Schüler ein Problem hat, welches er nicht mit verfügbarer Software lösen kann, so muss er eine solche Software selbst entwickeln oder verfügbare Software seinen Wünschen entsprechend anpassen. Für den Unterricht bedeutet dies, dass Aufgaben für Schüler eine Bedeutung besitzen und sinnvoll sein sollten, idealerweise so, dass ein Produkt später auch verwendet werden kann. Eine „Pseudo-Problemorientierung“, wie bspw. das Modellieren einer Taschenlampe auf dem Computer, erfüllt diese Anforderung nicht.

#### **Breites Feedback ermöglichen:**

Normalerweise erhalten Schüler Feedback nur durch den Lehrer. Motivation kann dadurch gesteigert werden, dass Schüler ihre Ergebnisse zum einen vor der Klasse präsentieren, zum anderen aber auch aus dem Klassenraum hinaustragen, z. B. durch Präsentation im Internet, wodurch auch Eltern und Freunde erreicht werden.

#### **Identifikation mit der Gruppe:**

Gruppenprojekte gehören bereits zu den typischen Arbeitsformen im Informatikunterricht. Längerfristige Gruppenprojekte können zur Identifizierung mit den Gruppenzielen führen und gemeinschaftliches kreatives Arbeiten anregen. Dieses Motiv kann bspw. in Gruppenprogrammierprojekten angewandt werden.

#### **Sinnvolles Lernen:**

Während sich Open-Source-Programmierer auch kreativ betätigen, um dabei zu lernen, ist Lernen im Schulkontext häufig vordergründig extrinsisch motiviert. Schafft es ein Lehrer allerdings, Schülern den Nutzen von Wissen und Konzepten zu vermitteln und finden die Schüler das Entwickeln von Software interessant, kann das Ziel, hierin besser zu werden, motivierend sein.<sup>71</sup>

Alle diese Beiträge zeigen ein breites Spektrum an Beispielen, bei welchen sich Kreativität positiv ausgewirkt hat und wo Kreativität erfolgreich zur Verbesserung des Lernerfolgs angewandt wurde. Vor diesem Hintergrund ist es erstaunlich, dass die Möglichkeiten, die Kreativität bietet, nicht häufiger in der informatischen Bildung angewandt werden.

---

<sup>71</sup> Dies kann bei besonders motivierten Informatikschülern beobachtet werden, die Programmieren als Hobby entdeckt haben und den Lehrer nach weiteren Informationen, die weit über den Unterrichtsinhalt hinausgehen, befragen.

### 4.3 Verwandte Ansätze und Projekte kreativen Lernens

#### Programmieren als Ausdrucksmöglichkeit persönlicher Ideen

Bereits 1977 prognostizierten Kay und Goldberg (1977) eine nachhaltige Beeinflussung des Lernens durch Multimedia. In ihren Erfahrungen mit dem *Dynabook* spielte die Möglichkeit, selbst zu programmieren, für Kinder eine wesentliche Rolle:

*[...] the kids love it! The interactive nature of the dialogue, the fact that they are in control, the feeling that they are doing real things rather than playing with toys or working out "assigned" problems, the pictorial and auditory nature of their results, all contribute to a tremendous sense of accomplishment to their experience. Their attention spans are measured in hours rather than minutes.*

Erfahrungen zeigen, dass dies bei heutigen, an Medien gewöhnten Schülern nicht anders sein muss. Aus eigenen Erfahrungen kann ich von Schülern berichten, die nach der Einführung in die Programmierung begeistert feststellen: „*Der Computer macht das erste mal genau das, was ich will!*“

Auch Peppler und Kafai (2005) sehen Programmierkenntnisse als Voraussetzung, um sich in digitalen Medien kreativ ausdrücken zu können und basieren ihre Aussage auf Erfahrungen in einem Digital Design Studio. Dabei warnen sie allerdings davor, Programmierprojekte im mathematischen oder wissenschaftlichen Zusammenhang anzusiedeln, sondern empfehlen Medien einzusetzen, die sich als Hauptinteresse der Schüler herausgestellt haben: Grafik, Musik und Video.

Kynigos (1995) beschreibt Programmieren als eine Möglichkeit für Kinder und Schüler, Ideen auszudrücken und daran zu lernen. Programmierbarkeit identifiziert er als wichtiges Merkmal explorativer Lernsoftware:

*Programming thus allows a merging of intuitive and reflective thinking. Furthermore, just as in language, programming allows the pupil to create, extend and enrich a domain-specific vocabulary [...] and, unlike spoken and written language, it enables pupils to design and construct objects.*

Diese Eigenschaft wurde von Papert und Harel (1991) in der Lerntheorie des Konstruktivismus verdeutlicht. Aufgrund der Relevanz für den Informatikunterricht soll hier das Konzept des Konstruktivismus noch einmal aufgegriffen werden.

#### Kreatives Lernen als konstruktivistisches Lernen

Konstruktivismus teilt die Vorstellung des Konstruktivismus, in welchem Lernen als aktiver Prozess verstanden wird, bei dem Wissensstrukturen konstruiert werden. Dem wird allerdings ein Aspekt hinzugefügt:

*[Learning] happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe. (Papert und Harel 1991)*

Die Konstruktion geschieht damit nicht nur im Kopf des Lernenden, sondern als Konstruktionsprozess in der realen oder virtuellen Welt. Ein solches Produkt kann ausprobiert, gezeigt, diskutiert und analysiert und auch bewundert werden. Der Lernende wird hierbei „eins“ mit dem Phänomen betrachtet werden soll, statt es nur „von außen“ zu betrachten.

Vor diesem Hintergrund stellt sich kreatives Lernen im Informatikunterricht, welches immer das Erschaffen eines Produkts oder Artefakts zum Ziel hat, als konstruktivistisches Lernen dar. Kreatives Lernen fügt dem den Aspekt des Problemfindens und Situierens des zu erschaffenen Produkts vor den Gegebenheiten und eigenen Möglichkeiten hinzu.

### Kreatives Lernen für die kreative Gesellschaft

Resnick (2007b) unterstreicht die Rolle von IKT zur Vorbereitung der Schüler auf die *Kreative Gesellschaft*<sup>72</sup>. Während IKT zum einen die Entwicklung in allen Bereichen des Lebens beschleunigen und damit kreatives Denken erforderlich machen, können diese Technologien – wenn sie entsprechend ausgewählt und eingesetzt werden – Schüler bei der Entwicklung zu kreativen Denkern unterstützen. Seiner Meinung nach geschieht dies am besten, wenn der Lernprozess einer „creative thinking spiral“ folgt (vgl. Abbildung 24).

*In this process, people imagine what they want to do, create a project based on their ideas, play with their creations, share their ideas and creations with others, and reflect on their experiences—all of which leads them to imagine new ideas and new projects. As students go through this process, over and over, they learn to develop their own ideas, try them out, test the boundaries, experiment with alternatives, get input from others, and generate new ideas based on their experiences. (Resnick 2007b, S. 18)*

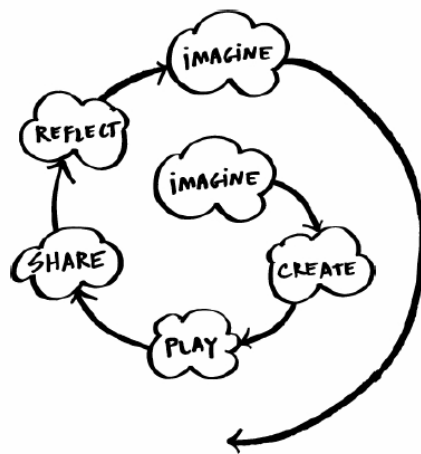


Abbildung 24: Creative thinking spiral (nach Resnick 2007a).

### Computer Club Houses

Resnick (2007a) berichtet darüber, wie in Computer-Clubhäusern eine kreative Verwendung von Computern gelernt wird. Das Projekt der „Computer Clubhouses“ (vgl. Resnick und Rusk 1996) verfolgt das Ziel, Schülern ein kreatives Lernen mit Computern zu ermöglichen. Der Clubhouse Lernansatz wurde 1993 in Zusammenarbeit mit dem MIT Medialab in den USA entwickelt und expandiert mit Ablegern inzwischen weltweit. Dem Lernansatz liegen vier Leitprinzipien zugrunde: Lernen durch Design, Folgen eigener Interessen, Bilden einer Gemeinschaft und eine Umgebung des Respekts und Vertrauens. Auch wenn informatische Bildung nicht das primäre Ziel des Clubhouse-Lernansatzes ist, sondern vielmehr interkulturelle Verständigung und ein tolerantes Miteinander angestrebt werden, so lernen die Teilnehmer hier quasi „nebenbei“ grundlegende Verfahren, Konzepte und Anwendungen der Informatik.

### Fazit

Die wissenschaftlichen Untersuchungen zur Kreativität in der informatischen Bildung lassen sich drei Kreativitätsperspektiven zuordnen: Die Bemühungen zur Steigerung von Motivation und Interesse der Schüler und Studenten versuchen die Chancen der Mensch-Perspektive wahrzunehmen; Studien zu kreativem Problemfinden und Problemlösen beziehen sich auf informatische

<sup>72</sup> Die Bezeichnung *Kreative Gesellschaft* nutzt Resnick als logische Konsequenz aus der Diskussion um die Informationsgesellschaft oder Wissensgesellschaft. Seiner Meinung nach befinden wir uns in einer *Kreativen Gesellschaft*, in welcher sich kreative Denkfertigkeiten als ausschlaggebend für Erfolg und Zufriedenheit darstellen.

Prozesse, also die Fach-Perspektive; Untersuchungen zu Informatiksystemen zur Förderung von Kreativität sind der Technologie-Perspektive zuzuordnen. Alle genannten Studien heben verschiedene Chancen hervor, die sich aus Kreativität ergeben (z. B. Motivation, Interesse, Förderung persönlicher Kreativität, Einblick in kreative Prozesse der Informatik) und verschiedene Instrumente, um diese zu erreichen (z. B. offene Aufgaben, Prototypen, Bearbeitung eigener und herausfordernder Probleme).

Auf den Grundideen der dargestellten kreativitätsverwandten Ansätze und Projekte basieren verschiedene pädagogische Werkzeuge und Programmiersprachen, wie LOGO, Smalltalk, Squeak und Scratch, welche im Informatikunterrichts bereits<sup>73</sup> eingesetzt werden und in Zukunft vermutlich noch stärkeren Einfluss nehmen werden. Problematisch ist, wenn diese Entwicklungen von den zugrunde liegenden Ideen isoliert verwendet werden. Dann würde nicht nur viel Potential verschenkt; das Ignorieren oder gar Unterbinden der kreativen Möglichkeiten dieser Werkzeuge könnte gar bei den Schülern zur Frustration führen. Insbesondere bei der Betrachtung von Resnicks „creative thinking spiral“ wird deutlich, dass sich Informatikunterricht bisher häufig nur auf die Aspekte des „create“ (wenn überhaupt!) und „reflect“ zu beschränken scheint. Ursachen und eine mögliche Lösung für diese Problematik werden im folgenden Abschnitt diskutiert.

#### **4.4 Herausforderungen als kreativer Zugang zu Problemen<sup>74</sup>**

In der Praxis des Informatikunterrichts und der Informatikdidaktik haben sich verschiedene Vorgehensweisen etabliert, die für kreatives Arbeiten problematisch sind. Im Folgenden sollen diese vor dem Hintergrund kreativen Lernens problematisiert werden und ein alternatives Vorgehensmodell für die Gestaltung kreativer Unterrichtssequenzen vorgeschlagen und begründet werden.

##### **4.4.1 Informatik-Lernen hat sich verändert**

Ein wesentlicher Teil des Erfolgs der dargestellten Projekte basiert auf der Berücksichtigung der geänderten Umstände und Lernvoraussetzungen der Schüler. Diese wachsen in einer technisierten Welt auf und verstehen IKT zunehmend als Medien, die in der Freizeit verwendet werden, Spaß machen und bei der Selbstverwirklichung helfen können. Folgt man der Begrifflichkeit der Digital-Natives vs. Digital-Immigrants-Argumentation (vgl. Prensky 2001) wird gar die groteske Situation deutlich, dass im Informatikunterricht Digital-Natives von Digital-Immigrants unterrichtet werden, was mitunter an überholten Unterrichtsgegenständen deutlich wird. Vor dem Hintergrund der immer intuitiver werdenden Bedienbarkeit von IKT und Benutzerschnittstellen hat sich der Fokus des Informatikunterrichts weg von der Nutzungsschulung hin zu Kerninhalten der Informatik entwickelt. Hierbei hat sich „Problemlösen“ als zentraler Unterrichtsinhalt etabliert. Fraglich ist, ob diese Beschäftigung mit „vorgesetzten“ Problemen bei den heutigen Schülern die richtige Motivation hervorruft oder ob sich ihre Bedürfnisse nicht besser berücksichtigen lassen (vgl. Guzdial und Soloway 2002).

Wie die Lernforschung zeigt, ist Lernen nicht nur ein rationaler Akt, sondern passiert auch in emotionalen Zusammenhängen (vgl. Lewis, Petrina und Hile 1998; Hascher 2005). Das bedeutet, dass nicht allein das Aneignen von Wissen ausreicht, um ein guter Problemlöser zu werden. Es ist genauso wichtig, wie ein Lernender über das, was gelernt werden soll, fühlt und welche Bedeutung der Lerngegenstand für den Lernenden hat. Für den Informatikunterricht impliziert das, den Computer nicht mehr nur als „Rechenmaschine“ wahrzunehmen, sondern auch als Medium persönlichen Ausdrucks.

Mit der Änderung der Sichtweise auf den Computer ist auch eine Änderung der Unterrichtsmethodik für den Informatikunterricht notwendig. So ist in der Informatikdidaktik das Problem der

---

<sup>73</sup> Bzw. wie im Fall von LOGO sogar schon lange.

<sup>74</sup> Dieser Abschnitts wurden bereits zum Teil sinngemäß und zum Teil wortwörtlich in (Romeike 2008f) veröffentlicht.

prozeduralen Unterrichtsmethodik seit langem bekannt: Statt schrittweiser Instruktion wird postuliert, zugrunde liegende Konzepte zu betonen. Dennoch scheint sich dies nur bedingt erfolgreich durchzusetzen. Ein Grund hierfür könnte sein, dass die Aufgaben gleich blieben und damit auch der Ansatz des Unterrichts. Ist eine Änderung der Methodik gewünscht, müssen neue Unterrichtskonzepte auch mit neuen Aufgaben realisiert werden. Bisher wird häufig ein Problemlöseansatz verwendet, wie er im professionellen Informatikbereich vermutet wird: Schüler erhalten den Auftrag, Probleme eines „Kunden“ (dem Lehrer) zu lösen. Die Brisanz dieser Problematik wird im Folgenden anhand verschiedener Beispiele verdeutlicht.

#### 4.4.2 Probleme des Problemlöseansatzes in der Praxis

##### Probleme sind oftmals nicht Probleme (sondern Aufgaben)

Ein *Problem* kann definiert werden als ein Hindernis, das zwischen einer Person und einem angestrebten Ziel steht (vgl. Häcker und Stapf 1998; Cambridge 2008; Wordsmyth 2007). Diese Definition steht im Gegensatz zum Verständnis von „*etwas, das getan werden muss*“, was eine Aufgabe bezeichnet.<sup>75</sup> Dieser Unterschied wird von Lehrern häufig übersehen, so dass viele „Probleme“ im Wesentlichen darin bestehen, das Rezept zur Lösung einer Aufgabe zu finden, woraufhin das Problem trivial und in wenigen Schritten lösbar wird; im Programmierkontext typischerweise durch Implementieren von Code. Schüler müssen in solchen Situationen ihr Vorgehen nicht reflektieren und brauchen sich über das Konzept hinter dem Problem keine weiteren Gedanken machen. Stattdessen suchen sie nach einer Möglichkeit, wie sie schnellstmöglich vom Ursprung des Problems zu seiner „Lösung“ kommen.<sup>76</sup>

Ein Schulbuch-Beispiel illustriert dieses Problem<sup>77</sup>:

[Gegeben ist ein Programm, welches zwei Zeiten (Minuten und Sekunden) addiert.]

*Erweitern Sie das Programm so, dass auch Tage eingegeben und verarbeitet werden können!* (Fothe 2002, S. 8)

Ein typischer Schüleransatz wäre, die dargestellten Programmzeilen zu kopieren und den Umrechnungsfaktor anzupassen. Dies ist in der Tat eine Lösung für die gegebene Aufgabe. Das Lernergebnis bleibt allerdings fraglich. Der Schüler hat sein mathematisches Wissen über die Umrechnung von Tagen in Stunden aufgefrischt, aber er hat keine Informatikkonzepte gelernt oder gefestigt. Dies führt zu einem weiteren Problem.

##### Probleme sind oft keine Informatikprobleme, sondern mathematische Probleme

Der Umstand, dass Computer (zu deutsch: Rechner) ursprünglich vor allem für mathematische Berechnungen verwendet wurden, scheint bis heute einen erstaunlichen Einfluss auf die informatische Bildung auszuüben: Viele Probleme entstammen der Mathematik. Fraglich ist, ob Schüler im Informatikunterricht nicht besser lernen sollten, Probleme der Informatik mit Methoden der Informatik zu lösen, da sich das Fach sonst seiner Legitimation beraubt. Viele Beispiele illustrieren dieses Problem:

[Gegeben ist ein Python-Programm zum Addieren zweier gemeiner Brüche.]

*Erweitern Sie ihr Programm um das Subtrahieren, Multiplizieren und Dividieren zweier gemeiner Brüche!* (Fothe 2002, S. 15)

<sup>75</sup> Aufgabe lässt sich definieren als „Arbeit die zu erledigen ist“ oder als „geistige Anforderung, für deren Bewältigung Methoden bekannt sind“ (Häcker und Stapf 1998; Oxford 2007).

<sup>76</sup> Bereits Ausubel, Novak et al. (1982) stellen diesbezüglich fest: „*Im mathematischen und naturwissenschaftlichen Unterricht ist das Lösen von Standardproblemen meistens kaum etwas anderes als mechanisches Memorieren und mechanisches Anwenden von Formeln, mechanisches Manipulieren mit Symbolen und der Gebrauch von eigentlich irrelevanten Schlüsseln für die Identifizierung von Problemen als Angehörige einer Klasse von Problemen*“ (S. 652).

Informatiklehrer machen sich diesen Ansatz offensichtlich zu eigen.

<sup>77</sup> Problematisch ist meiner Ansicht nach weniger das Verwenden einfacher Übungsaufgaben zur Verdeutlichung der Lerninhalte als vielmehr die Bezeichnung solcher Aufgaben als „Problemlösen“, wodurch auch ein fragliches Verständnis von Problemlösen geprägt wird.

Weitere Beispiele finden sich z. B. in den populären Programmieraufgaben zum GGT.<sup>78</sup> Diese Beispiele haben gemeinsam, dass Schüler eine *mathematische* Lösung für ein gegebenes Problem finden, lösen und anschließend in Programmtext implementieren sollen. In diesen Aufgaben ist kein algorithmisches Problemlösen involviert.

Auch in der informatikdidaktischen Forschung setzt sich die Problematik fort, dass mathematisches Problemlösen mit algorithmischem Problemlösen gleichgesetzt wird. Im Artikel „Misleading Intuition in Algorithmic Problem Solving“ (Ginat 2001) demonstriert der Autor, wie Intuition auf der Suche nach einer Problemlösung eine Hürde darstellen kann.

*A long railway is divided into many equal sections. The distance between every two adjacent sections is 1. N wagons are placed in N different sections which are not necessarily consecutive. How can the railroad workers move all the wagons to one of the sections such that the sum of all movements is minimal?*

Das Problem in dieser Aufgabenstellung (Mittelwert oder Median?) ist allerdings mathematischen Ursprungs und stammt nicht aus der Informatik. Wie berichtet, wählten die Studenten eine falsche *mathematische* Lösung, keinen falschen Algorithmus.

Zusätzlich kritisch bei der Anwendung mathematischer Probleme ist der motivationale Einfluss dieses recht unpopulären Fachs für die Schüler.<sup>79</sup>

### **Probleme haben oft keinen Bezug zum Schüler**

Die Lernforschung unterstreicht, dass Lernende bessere Leistungen bringen und dann motivierter sind, wenn eine Aufgabe für sie eine Bedeutung besitzt. Diese Sichtweise kann auf das Problemlösen übertragen werden. Ein Problem ist relevant und bedeutungsvoll für einen Schüler, wenn es Bezug zu seiner Lebenswelt aufweist, wenn er es wirklich als Hürde auffasst und wenn er das angestrebte Ziel auch wirklich erreichen will. Häcker und Stapf (1998) charakterisieren ein Problem anhand zweier weiterer Komponenten: einem unerwünschten Ausgangszustand und einem erwünschten Endzustand. Stellt ein Problem sich nicht als Problem für den Schüler dar, z. B. weil es ihm nur vom Lehrer auferlegt wurde oder es sich darüber hinaus nur um eine Aufgabe handelt, zu welcher der Lehrer die Lösung bereits kennt, sind die Bedingungen an ein Problem in diesem Verständnis nicht erfüllt – es fehlt die innere Motivation, das Problem zu lösen.

Diese Faktoren spielen typischerweise auch in den oben beschriebenen Szenarien eine Rolle.

#### **4.4.3 Ansätze zur Bewältigung der Probleme des Problemlösens**

Im Kontext des Problemlösens wird von einigen Forschern der Blick auf einen oft übersehenen Aspekt des Problemlösens erweitert, welcher aber in der Informatik und für Kreativität eine wichtige Rolle spielt: das Erkennen von Problemen.<sup>80</sup> In empirischen Studien zur informatischen Bildung wirkte sich die Berücksichtigung dieses Aspekts positiv auf die Motivation und Leistungen der Lernenden aus. Dazu gehört das Bestimmen eigener Aufgaben (vgl. Runco und Chand 1995; Fasko 2000), Lehren von Problemfindfertigkeiten (Kaasbøll 1998; Eastman 2003) und Verwenden „schlecht-definierter“ Probleme (Reed 2002).

Nach Meinung einiger Autoren ist die Entwicklung von Fertigkeiten im Problemfinden offenbar genauso wichtig, wie die Fähigkeit, Probleme zu lösen (vgl. Fasko 2000).

Im vorherigen Abschnitt wurde bereits auf die Notwendigkeit der Berücksichtigung persönlicher Aspekte der Lernenden hingewiesen. Auch Kontextualisierung, Personalisierung und Wahlmög-

<sup>78</sup> Papert (2000) verwendet ein ähnliches Beispiel zur Demonstration mangelnder persönlicher Relevanz abstrakter Unterrichtsinhalte: „*Learning how to find the common denominator of a bunch of fractions is boring for [the student] because he is not able to use it in any exciting way. It supports neither flights of the mind nor “hands-on” projects*“ (S. 722).

<sup>79</sup> Durchaus haben mathematische Probleme den Vorteil, dass sie exakt definierbar sind. Dieses ist allerdings untypisch für die Informatik und bringt zusätzlich die Nachteile, dass nur eine richtige Lösung möglich und das Problem meist lebensweltfern ist.

<sup>80</sup> Auch Problemfinden, Problemformulieren oder Problemmanagement.

lichkeiten können sich positiv auf Motivation und Leistungen auswirken. Schüler sind Individuen und bringen ihre eigenen Ideen, Wünsche und Bedürfnisse mit. Diese können angesprochen werden, so dass die Schüler ermutigt werden, im Unterrichtskontext ihre eigenen Ziele zu entwickeln. Wenn sie daran arbeiten, diese Ziele zu realisieren, stoßen sie auf Probleme, denen sie gern entgegentreten und dann motiviert lösen. Das impliziert die Anforderung an Lehrer, Lernsituationen in einer Weise zu beeinflussen, dass die Schüler inspiriert werden, eigene Ideen zu entwickeln und damit eigene Herausforderungen finden. So lösen sie Probleme, die für sie eine Bedeutung besitzen. Für Kreativität ist eine solche Herausforderung eine wichtige Voraussetzung (vgl. Kapitel 2).

Die Kernideen des situierten Lernens, entdeckenden Lernens und Konstruktivismus unterstützen diese Auffassungen und fügen zwei Aspekte hinzu: Demnach sind die Lernsituation und das Generieren von Ideen im aktiven Lernprozess zu berücksichtigen, im aktuellen Kontext offensichtlich schon ehe die Phasen des Problemfindens bzw. Problemlösens beginnen. Bereits bei der Betrachtung verschiedener Modelle des kreativen Prozesses konnte festgestellt werden, dass das Finden eines Ziels, das sich als interessante Herausforderung darstellt, und die Berücksichtigung des Umfelds wesentliche Bestandteile von kreativen Prozessen sind (vgl. Kapitel 2, Abschnitt 2.1.2), und dass deren Berücksichtigung zu den wirksamsten Mitteln der Kreativitätsförderung gehört (vgl. Kapitel 2, Abschnitt 2.4).

#### 4.4.4 Der Challenge-Cycle

Die Ergebnisse der bisherigen Überlegungen finden integriert in einem Vorschlag für ein Unterrichtsframework Anwendung, welches für kreatives Lernen im Informatikunterricht eine methodische Grundlage bildet. In diesem Framework werden „traditionelle“ Problemlöseaufgaben durch *Challenges*<sup>81</sup> ersetzt. Als Challenge wird im Folgenden ein Problem bezeichnet, das für den Schüler relevant ist, ein meist offenes Ergebnis besitzt und welches vorzugsweise vom Schüler selbst definiert wurde. Auf diese Weise werden Schüler ermutigt, kreativ zu sein, und lösen Probleme, die für sie eine Bedeutung besitzen. Schüler folgen dann eigenen Interessen, werden aber durch den Rahmen des Unterrichtskontexts geleitet. So lernen sie individuell an einem Thema, welches sie selbst mitgeprägt haben. Lernen nach diesen Prinzipien soll die Schlüsselfaktoren, die für das Interesse von Schülern verantwortlich gezeichnet werden, unterstützen: Motivation, bedeutungsvolle und persönlich herausfordernde Aufgaben, Relevanz, Kreativität, Anwendbarkeit, Beistuerbarkeit von Ideen, Experimentieren und die kreative Anwendung von IKT.

Die Phasen des Challenge-Cycles sind in Abbildung 25 dargestellt und werden in einer Unterrichtssequenz mehrere Male iteriert. Ein Eintritt kann mit jeder *Lernsituation* erfolgen, in welcher neue Konzepte eingeführt werden. In einer *Challenge-Phase* stellt der Lehrer sicher, dass alle für die Challenge grundlegenden Wissens-Bausteine von den Schülern verstanden und/oder verfügbar sind. Die Schüler werden nun vorbereitet und inspiriert, eigene Ideen zu entwickeln und eine persönliche Challenge zu bestimmen. Anschließend wird die Challenge gelöst, implementiert und präsentiert. Im Folgenden werden die einzelnen Phasen in einem (idealisierten) Durchlauf näher erläutert.

#### Challenge-Phase

Die Situation zum Zeitpunkt des Eintritts in den Challenge-Cycle wird maßgeblich durch individuelle Vorerfahrungen, Wissen, der Lernsituation und der persönlichen Situation der Schüler geprägt. Die Lernsituation stellt sich damit als maßgebliche, durch den Lehrer beeinflussbare, Variable des Lernprozesses und der Motivation der Schüler dar. Vom Lehrer ist sicherzustellen, dass Wissen und Konzepte, die zur Bewältigung einer bestimmten Herausforderung notwendig sind, gut eingeführt wurden oder von den Schülern selbst erarbeitet werden können. Des Weiteren sollte

<sup>81</sup> Eine Challenge (auch Herausforderung) bezeichnet ein „interessantes Problem oder eine schwierige Aufgabe“ (Wordsmyth 2007) und „etwas, das [...] Anstrengung bedarf, um erfolgreich bewältigt zu werden“ (Cambridge 2008). Wie in Kapitel 4.4.2 dargestellt, erfüllen Problemlöseaufgaben in der Informatik diese Anforderung häufig nicht.

zu Beginn einer Challenge der Betätigungsraum durch eine ungefähre Zielvorgabe (offene Aufgabe) insofern eingeschränkt werden, dass die Schüler sich nicht in der Beliebigkeit „verirren“ und eine Herausforderung wählen, die in der zur Verfügung stehenden Zeit gelöst werden kann.

Entsprechend eigenen Beobachtungen entwickeln Schüler eine Menge Ideen, nachdem sie die Möglichkeiten, die sich aus neuen Konzepten für sie ergeben, entdeckt haben. Aus diesen Ideen heraus wählen sie eine Challenge, die sie realisieren wollen. Das Finden der persönlichen Challenge beinhaltet auch das Evaluieren dieser bezüglich der zur Verfügung stehender Zeit, eigenem Wissen, voraussichtlicher Schwierigkeit und, wenn nötig, ein Anpassen der gestellten Aufgabe. Typische Merkmale solcher Challenges sind – im Gegensatz zu vielen Problemlöseaufgaben – Offenheit in der möglichen Bearbeitungstiefe, Relevanz, Originalität sowie die Identifikation des Schülers mit *seiner* Herausforderung.

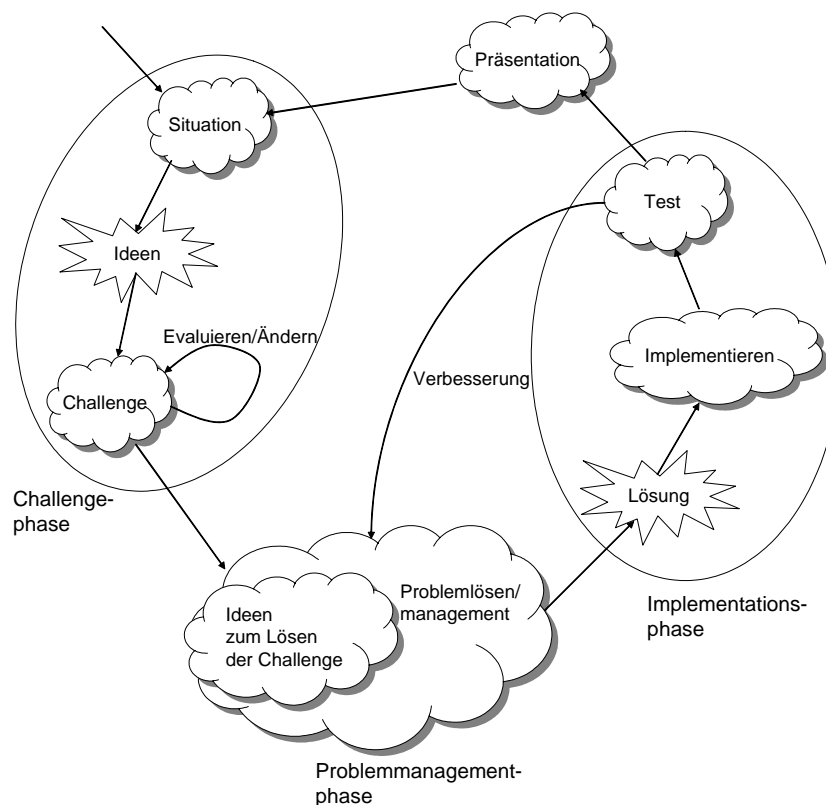


Abbildung 25: Challenge-Cycle.

### Problemmanagementphase

Nachdem eine Challenge präzisiert wurde, beginnt die Phase des Problemmanagements. Zu Beginn dieser Phase werden Ideen generiert, um das gestellte Ziel zu erreichen. Hier ist das Beherrschen von Kreativitätstechniken, wie z. B. Brainstorming, hilfreich. Diese Phase beinhaltet Problemlösestrategien mit dem Zusatz, dass die Schüler sehr motiviert sind, ihre persönliche Challenge zu bewältigen.

### Implementationsphase

Sobald eine Strategie und Lösung für die Challenge gefunden wurde, kann das Ergebnis implementiert und getestet werden.<sup>82</sup> Dadurch erhalten die Schüler die Möglichkeit, ihre Lösung auszu-

<sup>82</sup> Problemmanagement- und Implementationsphase verschmelzen häufig bei der Entwicklung von Teillösungen und Prototypen und werden mehrfach iteriert.

probieren. Oftmals findet sich Potential, das Produkt noch zu verbessern, was dann in der übrigen Zeit getan werden kann. Das können zusätzliche Features sein, eine bessere Bedienbarkeit oder eine schönere Gestaltung.

Nachdem die Schüler ihr (Mini-)Projekt beendet haben, bietet eine Präsentationsphase den Schülern die Möglichkeit, sich gegenseitig ihre Ergebnisse vorzustellen. Fragen, Entdeckungen, aufgekommene Probleme und deren Lösungen werden mit allen Schülern besprochen und diskutiert. Durch den Lernerfolg beim selbständigen Lösen der Probleme und durch die Inspirationen, welche die Schüler von einander erhalten haben, sind die Schüler jetzt bereit, die Grundlagen für weitere Herausforderungen zu lernen.

#### **4.4.5 Schlussfolgerungen**

IKT besitzen das Potential, als Werkzeuge für eine bessere Lebensqualität, zur eigenen Selbstdarstellung und kreativen Selbstverwirklichung gesehen zu werden. Da diese Werkzeuge sowohl inhaltlich als auch in ihrer Anwendung den Informatikunterricht prägen, kann diese Perspektive dafür genutzt werden, dass Schüler die Motivation entwickeln und behalten, sich tiefgründiger mit Informatik zu beschäftigen. Mit dieser neuen Sichtweise wird eine neue Unterrichtsmethodik notwendig, wie bereits beim Versuch des Wechsels von softwarenutzungsorientiertem Unterrichten zum Lehren fundamentaler Konzepte festgestellt werden konnte: Das Ziel wird nur schwer erreicht werden, wenn die Methoden und Aufgaben die gleichen bleiben. Der Challenge-Cycle verspricht Potential zu haben, die traditionelle Problemlösemethodik zu ersetzen. Er legt eine methodische Grundlage für die Gestaltung kreativen Informatikunterrichts und wird im Kapitel 8 als Basis für die Entwicklung eines Unterrichtsbeispiels kreativen Informatikunterrichts dienen.

## 5 Faktorenmodell kreativen Lernens im Informatikunterricht

Wie im vorherigen Kapitel verdeutlicht wurde, existieren verschiedenen Ansätze und Möglichkeiten, Kreativität in der informatischen Bildung zu berücksichtigen. Bisher geschieht das allerdings kaum. Um eine Grundlage für die Planung und Durchführung kreativen Informatikunterrichts zu legen, werden in diesem Kapitel die in der Betrachtung von Kreativität, Informatik und der informatischen Bildung herausgearbeiteten Faktoren informatikunterrichtsspezifisch in einem Faktorenmodell expliziert. Mit diesem Modell wird das Ziel verfolgt, das Phänomen der Kreativität im Informatikunterricht zugänglich, greifbar und überprüfbar<sup>83</sup> zu machen. Das Faktorenmodell wird darüber hinaus als Grundlage dienen, Kreativität im Informatikunterricht empirisch zu untersuchen.

Zuerst werden die wesentlichen Faktoren, die in den Kapiteln 2 und 3 dargestellt wurden, zusammengefasst und in spezifischen Modellen abstrahiert, um ausgehend von diesen Modellen, unter Berücksichtigung der in Kapitel 4 dargestellten Forschungserkenntnisse, ein Faktorenmodell kreativen Lernens für den Informatikunterricht aufzustellen.

Wie in den bisherigen Darstellungen deutlich geworden ist, manifestiert sich Kreativität in der Informatik besonders im Bereich der Softwareentwicklung. Vergleichbar mit der Orientierung der Fundamentalen Ideen der Informatik am Softwareentwicklungsprozess (vgl. Schwill 1993) soll die Betrachtung von Kreativität im Informatikunterricht auch hier auf diesen Bereich der Informatik beschränkt werden. Es ist zu vermuten, dass dies auch der Bereich ist, der Schülern in erster Linie als kreative Seite der Informatik bewusst wird. Weitere thematische Anknüpfungspunkte für Kreativität in der Informatik wurden im Kapitel 3 charakterisiert.

### 5.1 Faktoren aus der Perspektive der Kreativitätsforschung

In Kapitel 2 wurden im Bezug auf den Informatikunterricht wesentliche Faktoren, die in der Kreativitätsforschung diskutiert werden, herausgearbeitet. Als grundlegende Aussage konnte festgestellt werden, dass Kreativität im Bildungskontext aus einer persönlichen Sichtweise (P-Kreativität) zu betrachten ist und dass eine Förderung von Kreativität möglich und damit eine Berücksichtigung von Kreativität in der Schule sinnvoll ist.

Als wesentliche Faktoren stellen sich in der Diskussion folgende Aspekte dar:

#### **Die Betrachtung des Individuums (Mensch)**

Hierzu zählen insbesondere motivationale Faktoren: Kreativität und intrinsische Motivation bedingen sich gegenseitig. Die Untersuchung von Persönlichkeitsmerkmalen sowie die Messung von Kreativität haben in einer allgemeinbildenden Schule dagegen weniger Bedeutung. Wie verschiedene Ansätze suggerieren, kann kreatives Tun auch als inhärentes Bedürfnis des Menschen, sich selbst zu verwirklichen, verstanden werden.

#### **Die Betrachtung des Prozesses**

Sowohl im Unterrichtskontext als auch hinsichtlich informatischer Prozesse ist eine strukturorientierte Sichtweise des kreativen Prozesses opportun. Auch wenn Illuminationen den kreativen Prozess voranbringen, so lassen sie sich doch nicht erzwingen. Vor dem Hintergrund einer notwendigen Vorbereitung und Auseinandersetzung mit einem Problem entspricht die strukturorientierte Sichtweise besser den Möglichkeiten und Vorgehensweisen der Informatik. Amabiles Komponentenmodell (vgl. Abbildung 3) stellt wesentliche Faktoren dar, die den kreativen Prozess beeinflussen: Umfeld, Motivation, Wissen und Fertigkeiten.

---

<sup>83</sup> Überprüfbar im Sinne der Unterrichtsevaluation und Analyse informatischer Lernprozesse, nicht zur Überprüfung von Schülerleistungen.

### Die Betrachtung des Umfelds

Die Gestaltung der Umgebungsfaktoren des kreativen Prozesses hat einen erheblichen Einfluss auf die Entfaltungsmöglichkeiten von Kreativität. Diese sind im Unterricht durch den Lehrer positiv gestaltbar, z. B. durch das Stellen und Ermöglichen von Herausforderungen, Freiheit, Ressourcen, Unterstützung und Ermutigung.

### Die Betrachtung des Produkts

Während sich ein Teil der Kreativitätsforschung mit der Bestimmung gezeigter Kreativität und von Kreativitätsmerkmalen anhand von Produkten beschäftigt, ist dies in einer allgemeinbildenden Schule mit Blick auf die in der situationsorientierten Sichtweise dargestellten Problematik nicht empfehlenswert. Vielmehr sollten bei der Betrachtung des Produkts die motivationalen Vorteile für den einzelnen Schüler sowie die Vorteile des konkreten Lernens (Konstruktivismus) durch die Möglichkeit der Untersuchung von Produkten im Informatikunterricht genutzt werden.

Die genannten Faktoren beeinflussen sich gegenseitig (vgl. Abbildung 26). Chancen, die sich durch Kreativität ergeben, sind zum einen die Förderung intrinsischer Motivation durch die kreative Betätigung, zum anderen kann durch den Wechsel von konkretem und abstraktem Denken ein besseres Verständnis für die Lerninhalte erwartet werden. Nicht zuletzt wird hierdurch das Erreichen der höchsten kognitiven Lernzielstufe (vgl. Kapitel 2, Abschnitt 2.5) angestrebt.

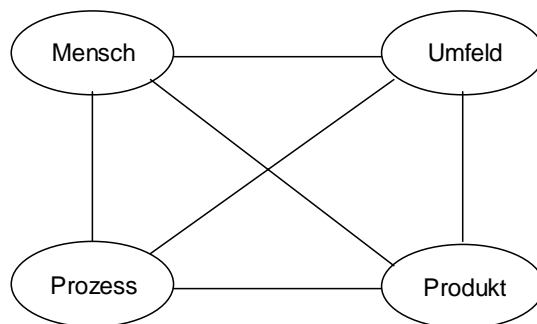


Abbildung 26: Modell der Einflussfaktoren von Kreativität.

## 5.2 Faktoren aus der Perspektive der Informatik

So, wie sich in der Betrachtung der Kreativitätsforschung unterschiedliche Sichtweisen herauskristallisiert haben, ist auch bei der Betrachtung der Informatik das Einnehmen verschiedener Perspektiven notwendig, um die Facetten der Kreativität zu erfassen. Während die Betrachtungen zur Mensch-Perspektive, Technologie-Perspektive und Fach-Perspektive unmittelbar für den Informatikunterricht relevant sind, stellen die Kunst-Perspektive und die Perspektive der artifiziellen Kreativität vor allem thematische Zugänge zur Kreativität in der Informatik dar. Diese eignen sich zwar für eine thematische Verankerung informatischer Inhalte, werden aber aufgrund der mangelnden methodischen Relevanz im Folgenden vernachlässigt.

Ein zentraler Stellenwert wird entsprechend der Grundannahme dem Softwareentwicklungsprozess eingeräumt, wie auch am Modell des kreativen Softwareentwicklungsprozesses verdeutlicht wurde (vgl. Abbildung 14).

### Betrachtungen aus der Mensch-Perspektive

Das Gestalten von Softwareartefakten kann motivieren und Spaß machen. Hierbei ist es vor allem das Gefühl der kreativen Betätigung, welches einen maßgeblichen Einfluss auf die Motivation ausübt. Ob ein Produkt nun tatsächlich eine kreative Leistung nach den Kriterien der Kreativitätsdiagnostik darstellt, stellte sich in den im Kapitel 3, Abschnitt 3.2 angegebenen Studien als

erstaunlich irrelevant heraus. Versucht man diese Eigenschaft der Softwareentwicklung im Informatikunterricht zu nutzen, impliziert diese Feststellung, dass die Einschätzung oder Bewertung eines kreativen Produkts unterbleiben sollte.<sup>84</sup> Wie die Studien mit professionellen Softwareentwicklern zeigen, sind es vor allem die den eigenen Bedürfnissen entsprechenden Herausforderungen, die für den Spaß und die Motivation hierbei sorgen. Offensichtlich lassen sich die in der Kreativitätsforschung beschriebenen motivationalen Bestrebungen des Menschen, sich durch Kreativität selbst zu verwirklichen, auch in der Informatik belegen. Die in Kapitel 4 zusammengefassten Studien legen einen vergleichbaren Effekt in der informatischen Bildung nahe.

### Betrachtungen aus der Fach-Perspektive

Softwareentwicklung stellt sich in der Informatik als kreativer Prozess dar, der sich an der Erstellung eines (häufig kreativen) Produkts orientiert und von Informatikern als kreativ empfunden wird. Dies unterstreicht den zentralen Stellenwert von Kreativität im Fach. Fachcharakteristika vereinfachen kreatives Tun: Das Verständnis grundlegender Informatikkonzepte als Bausteine, deren gute Dokumentation und damit Verständlichkeit und Anwendbarkeit. Die Relevanz der Baustein-Metapher in der informatischen Bildung ist an verschiedenen Stellen dokumentiert (z. B. Bruce, Buckingham et al. 2004; Duraes und Madeira 2006; Begel 1996).

### Betrachtungen aus der Technologie-Perspektive

Das Umfeld informatischer Prozesse wird maßgeblich durch auf IKT basierende Werkzeuge geprägt. Aus Sicht der Technologie-Perspektive stellen sich verschiedene Faktoren als begünstigend für Kreativität dar. Zum einen bringt Informatik IKT hervor, die Kreativität unterstützen oder in gewissen Bereichen oder für bestimmte Personengruppen erst ermöglichen. Zum anderen werden in der Informatik selbst kreativitätsunterstützende IKT angewandt.<sup>85</sup> Dazu zählen viele Entwicklungsumgebungen, Modellierungstools oder Visualisierungswerkzeuge. Des Weiteren gehört die Erforschung und Ausreizung der Möglichkeiten der vorhandenen Technologien zu den wichtigsten Förderern von Kreativität in der Informatik.

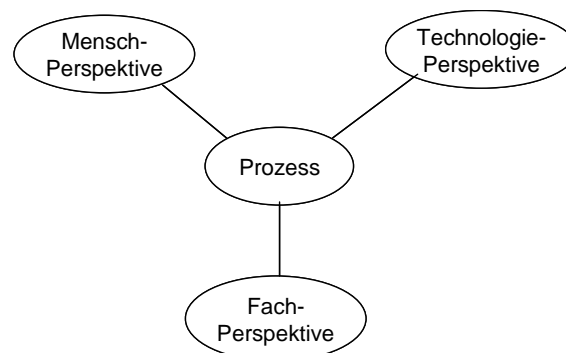


Abbildung 27: Relevante Kreativitätsperspektiven aus Sicht der Informatik.

## 5.3 Entwicklung des Faktorenmodells

Die herausgearbeiteten Faktoren werden nun unter Berücksichtigung der Anforderungen des Informatikunterrichts in einem Faktorenmodell zusammengefasst, welches die für kreatives Lernen im Informatikunterricht relevanten Faktoren darstellt. Im Folgenden werden die Entwicklung des Modells erläutert und die den einzelnen Faktorendimensionen zugrunde liegenden Hypothesen herausgestellt.

<sup>84</sup> Im Gegensatz zur Ermutigung und Belohnung, wenn Schüler kreative Bemühungen tätigen, eigene Ideen einbringen und eigene Wege gehen.

<sup>85</sup> Wie in Kapitel 3.4 dargestellt, erfüllen viele Werkzeuge der Informatik zumindest einige der Kriterien für kreative Softwaretools nach Shneiderman.

Die Grundlage für die Entwicklung des Faktorenmodells bilden die für den Informatikunterricht relevanten Kreativitätsperspektiven aus Sicht der Informatik (vgl. Abbildung 27), zu welchen die Einflussfaktoren von Kreativität in Beziehung gesetzt werden. Lernprozesse stellen sich in einem kreativen Informatikunterricht als (kreative) informatische Prozesse dar und bilden vor dem Hintergrund, dass kreatives Lernen immer auch konstruktionistisches Lernen bedeutet, das zentrale Element des Modells.

Die *Mensch-Perspektive* berücksichtigt subjektbezogene Merkmale, von denen sich vor allem Motivation und Interesse als Faktoren herausgestellt haben und die im Faktorenmodell in der Schüler-Dimension dargestellt werden.

**Hypothese 1: Im Informatikunterricht stimuliert kreatives Arbeiten Motivation und Interesse der Schüler.**

Die *Fach-Perspektive* verdeutlicht Fachcharakteristika, die für die Kreativität in der Informatik als ausschlaggebend angesehen werden können. Hierzu zählen die *Produktorientierung* sowie das Bausteinprinzip als Verständnis von Konzepten der Informatik.

**Hypothese 2: Im Informatikunterricht wird Kreativität aufgrund der kreativen Charakteristika des Fachs Informatik gefördert.**

Die *Technologie-Perspektive* unterstreicht die kreativitätsfördernde Funktion von IKT, die, als Besonderheit des Informatikunterrichts, maßgeblich das *Umfeld* prägen. Diese Dimension hebt das Gestaltungs- und Konstruktionspotential hervor, das IKT im Informatikunterricht zur Kreativitätsförderung bieten. Zusätzlich profitiert die Anwendung von IKT im Informatikunterricht davon, dass diese nicht nur Werkzeug, sondern zugleich auch Unterrichtsgegenstand sind. Konzepte der Informatik werden nicht nur an IKT verdeutlicht, sondern ein Konzeptverständnis ist auch elementar für deren adäquate und effiziente Nutzung.

**Hypothese 3: IKT fördern Kreativität im Informatikunterricht.**<sup>86</sup>

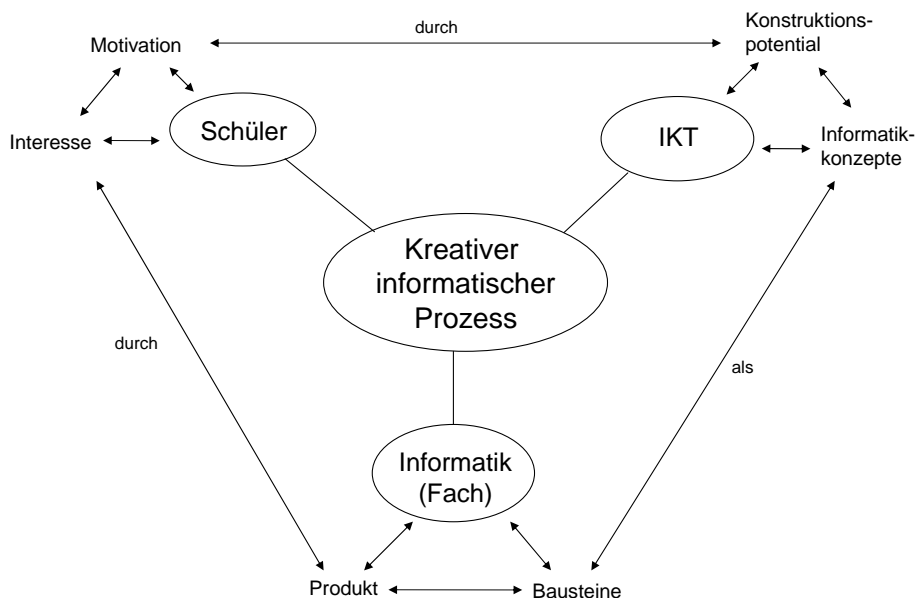


Abbildung 28: Faktorenmodell eines kreativen Informatikunterrichts.

<sup>86</sup> Unter der Voraussetzung, dass bei der Auswahl der Werkzeuge Shneidermans Kriterien (vgl. Kapitel 3.4) berücksichtigt wurden.

### Beschreibung des Modells

Die Faktoren des Modells stehen in einer untereinander vernetzten Beziehung (vgl. Abbildung 28). *Interesse* stimuliert *Motivation* und vice versa; Gestaltungsprozesse sind prädestiniert, die Interessen der Schüler anzusprechen. Kreativität in der *Informatik* wird vor allem an (Software-)Designprozessen deutlich, welche als Ergebnis des kreativen Prozesses zu einem *Produkt* führen. Im Informatikunterricht stellt ein solches Produkt das Ziel eines konstruktionistischen Lernprozesses dar. Das *Produkt* selbst ist Resultat der Anwendung fachspezifischen Wissens respektive der adäquaten Komposition der als *Bausteine* auffassbaren Fachkonzepte. Die Aussicht auf ein *Produkt* als Umsetzung eigener Ideen erweckt potentiell *Interesse* bei Schülern. *Bausteine* repräsentieren *Informatikkonzepte*, die wiederum grundlegend für ein Verständnis und den adäquaten und effizienten Einsatz von *IKT* und das daraus resultierende *Konstruktionspotential* sind. Das *Konstruktionspotential* von *IKT* als Möglichkeit kreativen Tuns *motiviert* Schüler, sich auf informatische Prozesse einzulassen.

Zur Planung und Durchführung kreativen Informatikunterrichts sind die Faktoren in ihrer Gesamtheit zu berücksichtigen. Die alleinige Beachtung der Fach-Dimension spiegelt nur den traditionellen Unterricht wider. Die alleinige Berücksichtigung der IKT-Dimension spiegelt die (gescheiterten) Versuche der Informationstechnischen Grundbildung wider. Die alleinige Betrachtung der Schüler-Dimension reflektiert schülerorientierte Lerntheorien ohne Berücksichtigung des Fachs. Abbildung 29 stellt die informatikspezifischen Implikationen für die Gestaltung kreativen Informatikunterrichts im Bezug auf die Dimensionen des Faktorenmodells dar.

### 5.4 Fazit

Die Anwendbarkeit des Faktorenmodells auf den Informatikunterricht wird aus folgenden Gründen erwartet:

1. Die Faktoren haben sich nicht nur im professionellen Umfeld, sondern auch in informatischen Bildungsprozessen als relevant erwiesen.
2. Vergleichbare Beobachtungen lassen sich teilweise im Informatikunterricht und in Freizeitbeschäftigungen von Schülern machen.

Eine Überprüfung des Modells, die weitere Zusammenhänge verdeutlicht, wird in Kapitel 6 vorgenommen. Hierzu werden die vorstehend aufgestellten Hypothesen herangezogen:

1. Im Informatikunterricht stimuliert kreatives Arbeiten Motivation und Interesse der Schüler.
2. Im Informatikunterricht wird Kreativität aufgrund des kreativen Charakters des Fachs Informatik gefördert.
3. IKT fördern Kreativität im Informatikunterricht.

Dargestellt werden im Modell nur jene Faktoren, die für den Informatikunterricht markant sind. Weitere Einflussfaktoren, insbesondere der kreativen Umgebung (z. B. Zeit, Freiheit, Lehrerrolle) und der Mensch-Dimension (z. B. Selbstsicherheit, Denkstil), werden nicht dargestellt.

Im Vergleich mit anderen Unterrichtsfächern sind die dargestellten Faktoren im Informatikunterricht etwas Besonderes, weil diese nur hier durch die Charakteristika des Fachs vernetzt auftreten und im Fachverständnis fest verankert sind. Letztendlich hat Informatikunterricht im Vergleich mit anderen Unterrichtsfächern den Vorteil, sich nicht auf „Nacherfinden“ und „Nachentdecken“ beschränken zu müssen (wie vorgeschlagen für die Fächer Physik und Chemie, vgl. Kapitel 2, Abschnitt 2.7.2 und 2.7.3) oder sich mit dem Erfinden gedanklicher Gebilde abfinden zu müssen (vgl. Abschnitt 2.7.1). Stattdessen können Methoden und Konzepte der Informatik mit Hilfe von IKT zum Erstellen von konkreten und bedeutungsvollen Softwareartefakten angewendet werden.

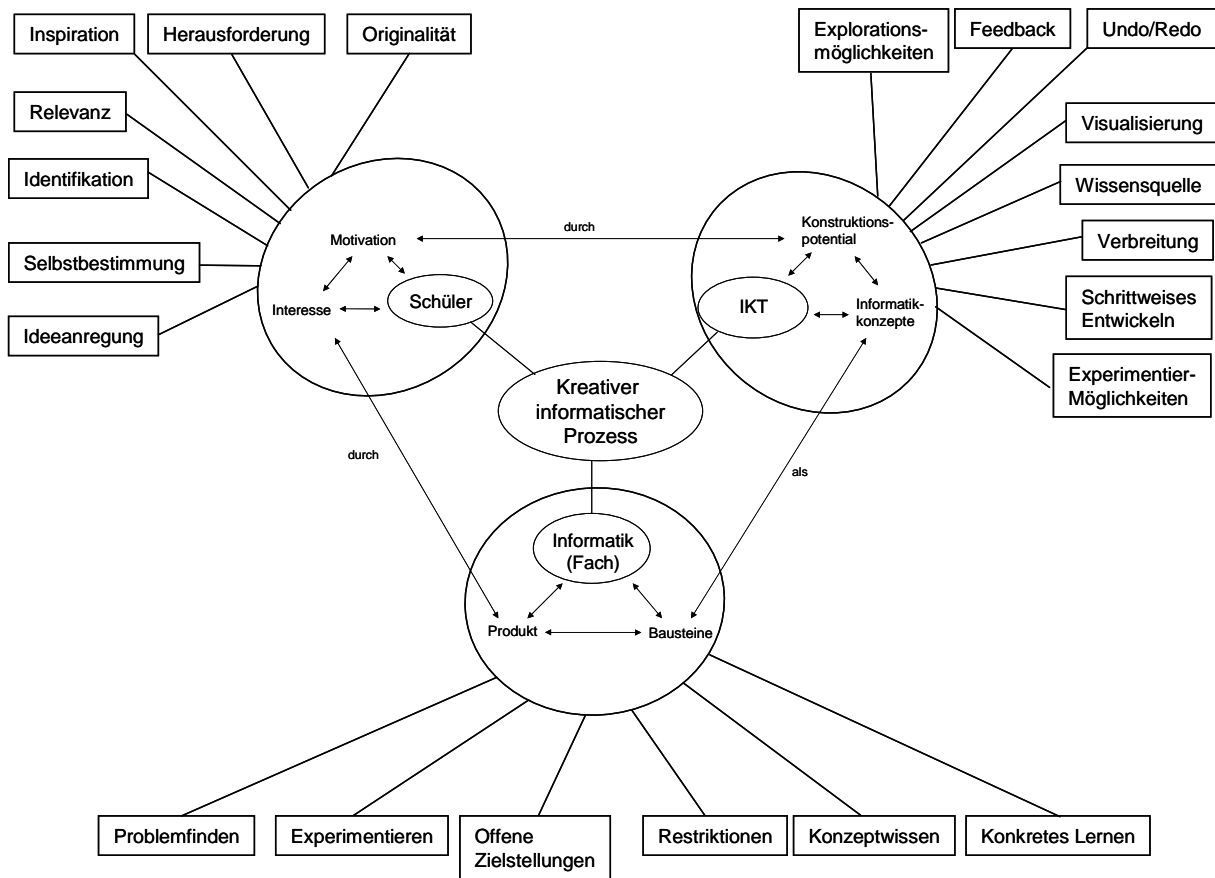


Abbildung 29: Implikationen für die Gestaltung kreativen Unterrichts im Bezug auf die Dimensionen des Faktorenmodells.

## 6 Untersuchungen zum Faktorenmodell

Die im Faktorenmodell dargestellten Dimensionen stützen sich bisher auf theoriebasierte Überlegungen aus Kreativität, Informatik und Lerntheorien. In diesem Kapitel sollen das Faktorenmodell und die aufgestellten Hypothesen überprüft werden.

Die Untersuchung des Faktorenmodells stützt sich auf verschiedene qualitative Studien und eine quantitative Studie, welche die Aussagen des Faktorenmodells hinsichtlich der aufgestellten Hypothesen stärken oder einschränken könnten. Ein Vorteil qualitativer Studien ist es, dass (neben der Überprüfung der Faktoren), die Faktoren durch zusätzliche Aussagen (Erfahrungen, Beispiele, Ansichten) untermauert werden können.

Der Versuch einer quantitativen empirischen Überprüfung aller Faktoren ist vor dem Hintergrund der Kreativitätsdiagnostik-Problematik unangebracht und würde aufgrund des Umfangs der Faktoren den Rahmen dieser Arbeit bei Weitem übersteigen.

Im Folgenden werden jeweils wesentliche Aspekte der Untersuchungsansätze dargestellt, die Ergebnisse beschrieben und eine Einordnung der Ergebnisse vorgenommen.

Zur Überprüfung des Modells werden folgende Ansätze gewählt:

1. In einer Analyse retrospektiv geschriebener Computernutzungsbiographien wird die Rolle der Kreativität in den Biographien von Studienanfängern der Informatik untersucht.
2. Informatikstudenten werden nach ihrem Verständnis von Kreativität und Erfahrungen mit Kreativität im selbst erlebten Informatikunterricht befragt.
3. Ein Interview mit einem Informatikschüler, der durch kreatives Arbeiten in außerhalb des Informatikunterrichts auffiel, wird als Einzelfallstudie ausgewertet. Hiervon erwarte ich detaillierte Hinweise zu seinem Verständnis von Informatik und seiner Motivation, sich kreativ mit Informatik zu beschäftigen.
4. Studienreferendare und Lehramtsstudenten der Informatik werden zu ihrer Sicht und ihren Erfahrungen bezüglich des Potentials und der Möglichkeiten, Kreativität im Informatikunterricht zu nutzen und zu fördern, befragt. Hieraus erwarte ich praxisorientierte Aussagen zum Faktorenmodell.

### 6.1 Kreativität in Biographien von Studienanfängern der Informatik<sup>87</sup>

Grundannahme dieser Studie ist es, dass sich die Faktoren des Kreativitätsmodells in Biographien von (ehemaligen) Schülern, die Informatikunterricht besuchten, widerspiegeln, falls Kreativität in ihrer Entwicklung eine Rolle spielte. Knobelsdorf und Schulte versuchen, anhand von Computerbiographien von Studienanfängern der Informatik herauszufinden, wie Schüler Informatik lernen und welches Verständnis sie von Informatik haben (Knobelsdorf und Schulte 2005, 2007; Schulte und Knobelsdorf 2007). Ziel dieses biographischen Ansatzes ist es, didaktische Interventionen für den Informatikunterricht zu entwickeln. Eine Grundannahme des Ansatzes ist, dass Vorstellungen von der Informatik durch die Interaktion mit Informatikartefakten geprägt werden. In einer ersten Analyse durch Knobelsdorf und Schulte wurde deutlich, dass Aspekte der Kreativität in den Computerbiographien eine Rolle zu spielen scheinen. Ziel dieser erneuten Analyse ist es herauszufinden, inwieweit sich die im Faktorenmodell eines kreativen Informatikunterrichts dargestellten Faktoren in den Biographien widerspiegeln, d. h. welche Rolle Kreativität in der Entwicklung zukünftiger Informatikstudenten spielt.

#### 6.1.1 Biographien der Computernutzung

Eine Computerbiographie ist eine retrospektive Erzählung, in welcher eine Person ihre Erfahrungen und Erlebnisse mit Computern subjektiv darstellt. Typischerweise enthält eine solche Erzäh-

---

<sup>87</sup> Teile dieses Abschnitts wurden bereits zum Teil sinngemäß und zum Teil wortwörtlich in (Knobelsdorf und Romeike 2008) veröffentlicht.

lung nur jene Aspekte, die vom Autor für die Darstellung „seiner“ Computergeschichte als wichtig erachtet werden. Studienanfänger der Informatik bemühen sich bei dieser Aufgabe meist darum darzustellen, wie sie zu dem Entschluss gekommen sind, Informatik als Studienfach zu wählen. In der Regel folgen die Texte einem typischen Muster und sind schlüssig aufgebaut. Dazwischen lassen sich wichtige Erfahrungen finden, welche die Entwicklung des Studenten begünstigt oder behindert haben. Da Computernutzungserfahrungen und Informatik – insbesondere für Anfänger – eng miteinander verbunden sind, offenbaren die Computerbiographien damit auch individuelle Vorstellungen und Verständnisse der Informatik (Knobelsdorf und Schulte 2007).

### 6.1.2 Forschungsfragen der Untersuchung

Die Bedeutung von Kreativität in Lernsituationen sowohl an Schulen als auch an Universitäten wurde in Kapitel 4 aufgezeigt. Auch in Computernutzungsbiographien von Studienanfängern der Informatik finden sich teilweise Lernprozesse und Motivationen wieder, die im Zusammenhang mit kreativen Tätigkeiten stehen. Vor dem Hintergrund der Frage, ob Kreativität einen Zugang zur Informatik darstellt und der Überprüfung der einzelnen Aspekte des Faktorenmodells kreativen Informatikunterrichts, werden die Forschungsfragen für diese Untersuchung wie folgt detailliert formuliert:

Zur Dimension Person<sup>88</sup>:

Steht die Motivation der Studenten, sich mit Informatik zu beschäftigen, im Zusammenhang mit Kreativität?

Zur Dimension Fach:

Fassen die Studenten Informatik, und damit die Aktivitäten im Zusammenhang mit Informatik, als ein kreatives Fachgebiet auf?

Zur Dimension IKT (Umfeld):

Wurde Kreativität durch IKT gefördert? Welche Faktoren haben darüber hinaus Einfluss auf eine kreative Beschäftigung mit Informatik genommen?

Darüber hinaus soll herausgefunden werden, inwieweit schulischer Informatikunterricht zu einem kreativen Informatikbild beiträgt. Da Informatikunterricht inzwischen an der Mehrzahl der allgemeinbildenden Schulen angeboten wird, haben Schüler die Möglichkeit, frühzeitig Informatik zu belegen. Schüler, die bereits in ihrer Freizeit Erfahrungen mit Informatik gemacht haben, sollten in der Schulformatik die Möglichkeit bekommen, zusätzliche Anregungen und Wissen für den kreativen Umgang mit Informatik zu erhalten.

### 6.1.3 Analyseverfahren

Analysiert wurden 135 Computernutzungsbiographien von Studienanfängern der Informatik (72 Männer, 16 Frauen) und der Bioinformatik (25 Männer, 22 Frauen).

Die Auswertung der Biographien stützt sich methodisch auf die *strukturierende Qualitative Inhaltsanalyse* nach Mayring (2007).

*Strukturierende Inhaltsanalyse will bestimmte Aspekte aus dem Material herausfiltern, will unter vorher festgelegten Ordnungskriterien einen Querschnitt durch das Material legen oder das Material unter bestimmten Kriterien einschätzen.* (Flick, Kardorff und Steinke 2007, S. 473)

### Strukturierende Qualitative Inhaltsanalyse

Die strukturierende Qualitative Inhaltsanalyse unterteilt sich in verschiedene aufeinander folgende Schritte. Zuerst werden Strukturdimensionen in Hinsicht auf die Forschungsfragen und den theoretischen Hintergrund bestimmt. Die Strukturdimensionen werden in Kategorien eingeteilt,

<sup>88</sup> Die Dimension *Person* wurde im Faktorenmodell durch *Schüler* konkretisiert. Während *Person* sich ein Subjekt in der informatischen Bildung bezieht, präzisiert *Schüler* das Subjekt im Informatikunterricht.

welche, zusammen mit den Kodierungsregeln und charakterisierenden Textpassagen, ein Kategoriensystem bilden. Nach einem ersten Kodierprozess mit ca. 10-50% der Daten wird das Kategoriensystem überarbeitet. Dem schließt sich der finale Kodierungsprozess an. Nachdem die Daten genau strukturiert wurden, werden die Kodierungsregeln angewandt. Der letzte Schritt kann mit verschiedenen Intentionen angewandt werden. In diesem Fall wurde eine *Strukturierung* verwendet, welche typische Aspekte der Strukturdimension beschreibt; in diesem Fall die Rolle von Kreativität in den Computernutzungsbiographien.

Die Strukturdimensionen basieren auf dem theoretisch entwickelten Faktorenmodell des kreativen Informatikunterrichts und unterteilen sich damit in Person, Fach und IKT (Umfeld). Um die Inter-Coder-Reliabilität zu gewährleisten, wurde das Material in einem Peer-Coding-Prozess von zwei Wissenschaftlern kodiert. Im Folgenden wird das Kategoriensystem detailliert beschrieben (vgl. Abbildung 30).

Das Kategoriensystem unterteilt sich in drei Hauptkategorien: *Person*, *Fach* und *IKT (Umfeld)*. Im Folgenden werden die zugehörigen Unterkategorien mit den korrespondierenden Codes erläutert.

Die Hauptkategorie *Person* unterteilt sich in zwei Subkategorien: *Motivation* und *Verhaltensweisen*<sup>90</sup>. Motivation enthält dabei die folgenden Codes (kursiv und fett), welche sich auf motivationale Aspekte der Studenten beziehen: Etwas tun, weil es eine *Herausforderung* darstellt, *Spaß* macht und/oder *interessant* ist; *Schaffen zum Gebrauch* (etwas tun, um es später zu benutzen); Motivation durch die *Identifizierung mit einer Gruppe* oder *mit einem Artefakt*; es kann motivieren etwas *Originales* zu tun, eine *Reputation* aufzubauen, etwas persönlich *Relevantes* zu tun oder anderen zu helfen (*Altruismus*).

Die Subkategorie *Verhaltensweisen* enthält die folgenden Codes (kursiv und fett), welche sich auf Aspekte des Verhaltens beim kreativen Tun beziehen: Kreatives Tun kann von *kritischem Denken* oder *evaluativem Denken* ausgehen, der Fähigkeit, *Probleme aufzuspüren* oder dem Bedürfnis zu handeln, die Lücke zwischen Wissen und Verstehen zu schließen und im *Sehen von Möglichkeiten*, neue Produkte und Verhaltensweisen zu entwickeln, Lernstrategien beinhalten die Fähigkeit zu *erforschen* und zu *experimentieren* und sich *aktiv innerhalb* und *außerhalb der Schule* zu betätigen.

Die Hauptkategorie *Fach* bezieht sich auf Kreativitätscharakteristika der Informatik als Fach. Hierzu gehören die folgenden Codes (kursiv und fett), welche diese Charakteristika aus der Perspektive der Studenten beschreiben: Informatik ist ein *kreatives Fach*, es werden Sachen gemacht, die mit *Kunst* vergleichbar sind,<sup>91</sup> *Problemlösen* erfordert Kreativität, Ziel einer kreativen Tätigkeit ist in der Regel ein *Produkt*, das *Bausteinprinzip* der Informatik ist förderlich für

Codesystem (Ausschnitt)	Codes
<b>Person</b>	
Intrinsische Motivation	14
Extrinsische Motivation	3
Schaffen zum Gebrauch	23
Herausforderung	6
Spaß	19
Interesse	45
Reputation	9
Relevanz	8
Stolz	6
<b>Umfeld</b>	
Möglichkeiten	17
Kreativitätsunterstützung: Internet	11
Bezugsperson	17
<b>Fach</b>	
kreative Natur	2
Kunstperspektive	1
Problemlösen	4
Produktorientierung	22
Konzeptwissen	6
Lsg. duch Exp.	4
Lernstrategien: Erforschen	22

Abbildung 30: Hauptkategorien der Strukturdimensionen und häufige identifizierte Codes.<sup>89</sup>

<sup>89</sup> Es bleibt zu beachten, dass der Forschungsansatz keine quantitativen Rückschlüsse zulässt.

<sup>90</sup> *Verhaltensweisen* bezieht sich auf Aktivitäten im kreativen Prozess.

<sup>91</sup> Dieser Code gehört zwar nicht zur Fach-Perspektive von Kreativität, da aber bei den Studenten nicht ein entsprechendes Kreativitätsverständnis vorausgesetzt werden kann, wurden die Daten auch auf das Vorkommen kunstbezogener Aussagen analysiert.

Kreativität und unterstützt kreatives Tun, kreative Prozesse in der Informatik folgen einem *Anlass/Stimulus*, basieren auf *Konzeptwissen*, müssen sich an *Einschränkungen* halten und beinhalten Lösungen durch *Experimentieren*.

Die Hauptkategorie *IKT (Umfeld)* bezieht sich auf die Frage, wie Computer als Arbeitsumgebung gesehen werden. Die Kategorie wird durch folgende Codes (kursiv und fett) beschrieben: Software und Entwicklungsumgebungen eröffnen vielfältige *Möglichkeiten* für eine kreative Betätigung, ein *Klima der Vielfalt* kann inspirierend sein (so ergibt sich z. B. in einem Computerclub die Möglichkeit, Ideen auszutauschen und sich gegenseitig zu unterstützen), und Software kann auch direkt *Kreativität fördern* und unterstützen.

Die Anwendung des Kategoriensystem in der Kodierung erfolgte mit Hilfe der Kodierungssoftware MAXQDA (2008).

#### 6.1.4 Ergebnisse

Die Studenten beschreiben generell eine starke Faszination für die Beschäftigung mit dem Computer und ein großes Interesse am Erforschen und Lernen über Computer und Informatik, insbesondere im Zusammenhang mit der Erstellung von Software. Die genannten Charakteristika sind typisch für kreative Prozesse. Im Folgenden werden die Ergebnisse analog dem Analyseprozess dargestellt. Hierzu werden zuerst die Ausprägungen der Aspekte des Kategoriensystems dargestellt, wie sie in den Biographien gefunden wurden. Anschließend werden die Ergebnisse im Bezug auf die formulierten Forschungsfragen analysiert.

#### Steht die Motivation der Studenten, sich mit Informatik zu beschäftigen, im Zusammenhang mit Kreativität?

Bezüglich der Hauptkategorie Person bestand das Interesse darin, motivationale Aspekte und Verhaltensweisen zu finden. In etwa einem Drittel der Biographien konnten Aussagen gefunden werden, die für Kreativität charakteristische Motivationen und Verhaltensweisen beschreiben.

Das Interesse für Computer, das am meisten genannte motivationale Merkmal, erscheint in verschiedenen Verbindungen: Faszination und Neugier für den Computer und seiner Funktionen im weitesten Sinn und Spaß und Enthusiasmus für das Arbeiten mit dem Computer.

*Dieser erste Personalcomputer lief noch nicht sonderlich schnell und auch nicht sehr stabil, dennoch war ich damals schon von den schier endlosen Möglichkeiten dieser Apparate fasziniert und versuchte so viel es ging herauszufinden und auszuprobieren [...]. [058I1987mU7]<sup>92</sup>*

Auch wenn es alleingegenommen nicht direkt ein Indikator für Kreativität ist, so steht *Interesse* doch wesentlich deutlicher in einer Beziehung zu Kreativität, wenn man den Kontext der Aussagen mit einbezieht. Interesse, Faszination und Spaß erscheinen immer wieder im Zusammenhang mit Aktivitäten, die kreatives Arbeiten beinhalten, wie Webseitenerstellung, Basteln mit Hardware und Programmieren. Zusätzlich bietet Programmieren Möglichkeiten, Artefakte zu entwerfen, verschiedene Ideen zu explorieren und unabhängig zu arbeiten, wie im folgenden Beispiel beschrieben wird.

*[Informatikunterricht] war ja auch etwas anderes. Man musste nicht dasitzen und dem Lehrer zuhören, aufschreiben usw. Man konnte selber tätig werden und Programme schreiben. [127B1987wU7]*

Alle Aussagen, die *Interesse* verdeutlichen, beziehen sich außerdem auch auf Lernen und Arbeitsweisen: Unabhängiges Arbeiten, eigenständiges Erschaffen eines Artefakts sowie aktives und unabhängiges Explorieren sind die am häufigsten genannten Aspekte der Kategorie Verhaltensweisen.

<sup>92</sup> Der Code identifiziert eine Biographie. Die erste Nummer steht für die Nummer der Biographie, der Großbuchstabe kennzeichnet die gewählte Studienrichtung (B steht für Bioinformatik, I für Informatik), die zweite Nummer ist das Geburtsjahr, der Kleinbuchstabe kennzeichnet das Geschlecht, U7 bezieht sich auf die Studie.

Auch wenn viele Studenten Interesse an Computern äußern, so wollten einige doch die Funktionalität von Computern aus einem bestimmten Grund verstehen, und zwar, um ein bestimmtes Problem zu lösen. Entsprechend lassen sich zwei Gruppen identifizieren: Eine Gruppe interessiert sich für Computer und ist von den Möglichkeiten, die Computer bieten, fasziniert. Für sie sind Originalität, Identifikation mit originalen Arbeiten und den geschaffenen Artefakten bedeutsam. Diese Gruppe experimentiert gern und erforscht den Computer unabhängig. Die andere Gruppe interessiert sich ebenfalls für Computer, allerdings vor allem, um damit bestimmte Aufgaben zu erledigen. Für einige Studenten ist die Reputation, die sie durch die Anerkennung ihrer Arbeit und Expertise bei Freunden und Familie erhalten haben, ein wichtiger Aspekt, was auch an einer Art von Stolz deutlich wird.

Kritisches Denken, Evaluieren und Problemfinden werden fast gar nicht erwähnt, dabei sind diese auch wichtige Charakteristika von Kreativität. Die Ursache hierfür kann allerdings an der Form der Datenerhebung liegen. Die Studenten wurden nicht explizit nach diesen Aspekten gefragt.

Alle negativen Kommentare oder Schilderungen von Enttäuschungen beziehen sich auf erlebten Informatikunterricht, welcher in der Regel negativ attribuiert wird. Studenten, die der zweiten Gruppe zuzuordnen sind, kritisieren überwiegend mangelndes Wissen des Informatiklehrers und Kommunikationsdefizite. Im Gegensatz dazu wurde der Unterricht von Studenten der ersten Gruppe positiv erwähnt, wenn der Lehrer selbst motiviert und engagiert war und für eine unabhängigkeitunterstützende Unterrichts Atmosphäre sorgte.

### **Fassen die Studenten Informatik, und damit die Aktivitäten im Zusammenhang mit Informatik, als ein kreatives Fachgebiet auf?**

Die Aktivitäten der Studenten entwickeln sich mit ihrer Biographie (Knobelsdorf und Schulte 2007). Diese Aktivitäten beginnen meist mit Spielen, gefolgt vom Erkunden von Anwendungsprogrammen und deren Möglichkeiten, Experimentieren mit dem Computer, dem Sammeln von Wissen und der Internetnutzung. Währenddessen entdecken sie Programmieren, welches meist im Kontext mit der Erstellung von Internetseiten geschieht. Zu diesem Zeitpunkt belegen einige Studenten Informatik als Schulfach, wo häufig ebenfalls Programmieren im Vordergrund steht. Entsprechend enthalten viele Biographien einen erheblichen Teil einer Programmier-Biographie:

*Als ich in der 11. Klasse Java programmieren lernte, kam die große Wende, weil ich jetzt Programme nach meinen Wünschen erstellen konnte[...]. [026I1986mU7]*

Während die erstgenannten Aktivitäten sich kaum auf Informatik beziehen, so gehört doch Programmieren zur Informatik dazu und beeinflusst das studentische Bild von der Informatik. Entsprechend wurde das Produzieren von Produkten (Artefakten) sehr häufig erwähnt. Andere kreativitätsrelevante Eigenschaften der Informatik, wie z. B. das Bausteinprinzip, wurden dagegen nicht erwähnt.

Problemlösen spielt in der Informatik eine zentrale Rolle und wird in der informatischen Bildung in Schulen und Hochschulen regelmäßig betont. Ebenso ist Problemlösen ein wichtiger Aspekt der Kreativität. Dennoch wurde dieser Aspekt in keiner Biographie explizit erwähnt<sup>93</sup>. Implizit wurden Problemlöseaktivitäten allerdings im Zusammenhang mit Programmiererfahrungen beschrieben. Dabei werden Probleme als Herausforderungen beschrieben, die gemeistert werden müssen, um ein bestimmtes Ziel zu erreichen. Diese Herausforderungen sind für die Studenten sehr motivierend, im Gegensatz zu Problemen, die auf Verlangen des Lehrers gelöst werden müssen um gute Noten zu erhalten.

Informatikunterricht trifft die Erwartungen der Studenten, wenn der Unterricht Programmieren und Projektarbeit beinhaltet. Unterricht, der sich allein auf die Anwendung von Computern bezieht, wird dagegen sehr negativ beschrieben.

---

<sup>93</sup> Eine Möglichkeit hätte sein können: „Weil mir das Lösen von Problemen mit dem Computer so viel Spaß macht, habe ich mich entschieden Informatik zu studieren.“

### **Wurde Kreativität durch IKT gefördert? Welche Faktoren haben darüber hinaus Einfluss auf eine kreative Beschäftigung mit Informatik genommen?**

Einige Studenten stellen fest, dass sie mit ihrem ersten Computerzugang die Möglichkeit erhielten, Dinge zu tun, zu denen sie vorher nicht in der Lage waren:

*Die Möglichkeiten, die der PC anbietet, sind kaum zu umfassen. [092B1988mU7]*

*Mit diesem [Computer] machte ich meine ersten Erfahrungen mit einer faszinierenden elektronischen Welt. [...] [Es gab] eine schier nie endende Anzahl von Möglichkeiten mit diesem Gerät. [101B1986mU7]*

Der Beginn einer „Karriere“ in der Informatik beginnt damit offensichtlich für viele mit dem ersten Computerkontakt. Bald werden die Möglichkeiten des Computers erkundet und schließlich wird er dazu genutzt, eigene Software zu erstellen. Zu diesem Zeitpunkt wird der Computer als ein kreatives Werkzeug angesehen, wie von Shneiderman (2000) beschrieben: Er unterstützt den Anwender bei der Beschaffung von relevantem Wissen (immer wieder wurde in dieser Hinsicht das Internet von den Studenten genannt), bildet eine Plattform zum Explorieren und Experimentieren, gibt sofortiges Feedback (dieser Aspekt bezieht sich hauptsächlich auf Programmiersprachen und/oder Webdesign) und erlaubt es, die Ergebnisse an andere zu verteilen, z. B. durch das Publizieren einer Internetseite online. Das Internet dient offenbar als Katalysator: Es ist Informationsquelle, sorgt für einen Anlass und Inspiration:

*Mit meinem ersten Internetanschluss [...] konnte [ich] mein Wissen stetig erweitern und im Basteln von Internetseiten unter Beweis stellen. [002I1985mU7]*

Nur in wenigen Biographien wird eine Bezugsperson als relevant erwähnt: Ein Elternteil aus dem Informatikfach, ein Freund mit gleichen Interessen oder ein inspirierender Lehrer.

#### **6.1.5 Schlussfolgerungen**

Die Ergebnisse dieser Untersuchung zeigen, dass sich in Computernutzungsbiographien von Informatikstudienanfängern zahlreiche Charakteristika von Kreativität wiederfinden. Diese Studenten beschreiben Informatik als etwas das Spaß macht, kreativ ist und Unabhängigkeit fördert. Dieses wird typischerweise im Kontext von Programmieren beschrieben.

Das Streben nach einer gut funktionierenden Software konnte als wichtigste Motivation für das Beschäftigen mit der Programmierung identifiziert werden. In den meisten der beschriebenen Programmierprozesse werden Aufgaben beschrieben, die für die Studenten eine persönliche Bedeutung besitzen, sich aber als überraschend irrelevant als Produkt darstellen.<sup>94</sup> In diesen Prozessen ist die Aktivität (häufig Programmieren) am wichtigsten, was für kreative Prozesse typisch und von Künstlern bekannt ist. Diese Gruppe von Studenten ist fasziniert und interessiert an den Möglichkeiten, die ihnen der Computer und die Informatik bieten. Sie drücken ein starkes Verlangen nach weiterem Wissen, Exploration und Verständnis aus. Die Aktivitäten dieser Gruppe können als hochkreativ bezeichnet werden.

Eine andere Gruppe von Studenten zeigt ebenfalls Faszination und Interesse für den Computer, aber auf eine pragmatische Art. Für diese Studenten steht die effiziente Nutzung des Computers im Mittelpunkt. Wissenserwerb wird als notwendig zur Computernutzung und zur effizienten Lösung von Problemen verstanden. Während dieser zweite Ansatz zu dem Ansatz passt, der häufig im Informatikunterricht zu finden ist, beschreiben Studenten, die sich gern kreativ mit Computern auseinandersetzen, ihren Unterricht als enttäuschend. Offensichtlich ist eine stärkere Betonung von Kreativität im Informatikunterricht notwendig, um diese Studenten nicht zu verlieren. Möglicherweise lassen sich sogar mehr Studenten für die Informatik gewinnen, wenn sie die kreative Seite der Informatik entdecken.

<sup>94</sup> Viele der beschriebenen Produkte sind als kommerzielle oder professionell erstellte Produkte in überlegender Qualität verfügbar (z. B. Spiele).

Sind die produzierten Softwareartefakte nützlich, wie es die Definition von Kreativität verlangt? Die Nützlichkeit muss hier ebenfalls von einer persönlichen Sichtweise betrachtet werden. Dennoch, auch wenn ein gewisser Nutzen für Spiele und Internetseiten existiert, ist kommerzielle Software in der Regel wesentlich ausgefeilter und umfangreicher. Dieser Aspekt scheint für die Studenten allerdings nicht wichtig zu sein. Der Prozess, bedeutungsvolle Artefakte zu produzieren, wird als kreativ aufgefasst. Es scheint mit dem Nutzen von Software gleichwichtig zu sein, wenn nicht sogar wichtiger. Die Motivation, Faszination, Ausdauer bei der Beschaffung von Wissen und Fertigkeiten und der Enthusiasmus der Studenten im Bezug auf ihre kreativen Tätigkeiten sind beeindruckend.

Zusammenfassend kann festgestellt werden, dass sich die Elemente des Faktorenmodells in Computerbiographien von Studenten, deren Lern- und Entwicklungsprozess als kreativ zu bezeichnen ist, nachweisen lassen.

## 6.2 Kreativitätsverständnis und Erfahrungen von Informatikstudenten

Die Frage nach dem Verständnis von Kreativität in der Informatik und die Erfahrungen im selbst erfahrenen Informatikunterricht waren Gegenstand einer Pilotstudie mit Studenten an der Universität Potsdam, die ich explizit aufforderte, Kreativität in ihrem erlebten Informatikunterricht zu reflektieren. Aufgrund der expliziten Fragestellung geben die Antworten der Studenten einen konkreteren Einblick in deren Kreativitätsverständnis und Wertung von Kreativität im Informatikunterricht. Befragt wurden 13 Studenten des zweiten und dritten Studienjahres in einer Hausaufgabe zur Vorlesung Didaktik der Informatik.

Die Fragestellung lautete:

*Diskutieren Sie, ob und ggf. wie Informatikunterricht Kreativität fördern kann. Als Einstieg können Sie folgende Fragen beantworten: Empfinden Sie Ihren Informatikunterricht als kreativ? Erleben Sie Ihre Beschäftigung mit Informatik als kreativ?*

Im Folgenden werden wesentliche Aussagen bzgl. dem Kreativitätsverständnis und den Erfahrungen der Studenten dargestellt. Die Auswertung der Vorstellungen der Studenten, wie Informatikunterricht Kreativität fördern könne, unterbleibt an dieser Stelle, da diese Aussagen wenig fundiert sein dürften.

### Verständnis

Die Studenten geben ein Verständnis von Kreativität an, welches im Wesentlichen der herausgearbeiteten Definition von Kreativität und dem Faktorenmodell entspricht. Neben wenigen Studenten, die zwischen künstlerischer und logischer bzw. wissenschaftlicher Kreativität unterscheiden, sieht die Mehrzahl der Studenten Kreativität in jedem Fachgebiet als möglich an. Kreativität wird demnach gezeigt wenn

*Wissen und/oder zur Verfügung stehende Möglichkeiten neu kombiniert und angewendet werden, um etwas Neues bzw. Unbekanntes zu erschaffen. [Antwort 1]*

Eine andere Definition stellt das Kombinieren noch deutlicher in den Vordergrund:

*Kreativität ist weitgehend die Fähigkeit, vorhandene „Teile“ so zu kombinieren, dass etwas Neues entsteht. Diese „Teile“ können Materialien, Melodien, Farben, Worte, Bewegungen, Wissensselemente oder Sonstiges sein. [Antwort 4]*

Diese Auffassung spiegelt im Allgemeinbezug wider, was im Faktorenmodell als Bausteinprinzip bezeichnet wird.

Als Beispiele werden u. a. Algorithmenentwurf, Problemlösen und Programmieren angegeben. Offensichtlich sehen die Studenten damit Informatik als kreatives Fachgebiet an.

### Erfahrungen

Die Studenten besuchten in den letzten drei bis fünf Jahren ihrer Schulbildung Informatikunterricht. Die Hälfte der Studenten gibt an, dass sie ihren Informatikunterricht als kreativ erlebt hat,

während sich die andere Hälfte nicht an kreative Aktivitäten und Aufgaben im Unterricht erinnern kann. Stattdessen berichten sie über Aufgaben, die normalerweise das Lernen einer Programmiersprache und das Nachvollziehen von Lehrerbeispielen umfassten. Problemlösungen wurden mit der Aufgabenstellung im Detail besprochen und/oder konnten durch Anwenden einfacher Muster gefunden werden. Kreative Ideen und eigene Ansätze für Problemlösungen wurden vom Lehrer häufig als „falsch“ gestoppt; stattdessen wurde darauf hingearbeitet, dass alle einen gleichen Ansatz wählen. Die Anwendung von Software wurde in der Regel in einem „How-to“-Stil unterrichtet, unterstützt durch konvergente Instruktionen, wie einfache Aufgaben zu lösen sind.

Die Studenten, die einen kreativen Unterricht erlebten, berichten vom Gegenteil: Hier hatte jeder Schüler die Möglichkeit, Probleme auf eine individuelle Art und Weise zu lösen und genügend Zeit zum Experimentieren und Umsetzen eigener Ideen. Diese Studenten geben an, dass es für sie motivierend war, wenn sie ein einführendes Framework für eine Aufgabe erhielten. Kreatives Arbeiten in Projekten machte den Studenten Spaß, war motivierend und interessant und erlaubte selbstbestimmtes Arbeiten.

Ihre derzeitigen Lernaktivitäten in Informatikkursen an der Universität fassen die Studenten einstimmig als nichtkreativ auf. Kreatives Arbeiten erfolgt derzeit nur in ihrer Freizeit in privaten Projekten. Die Kurse werden als theoretisch empfunden und scheinen nur dem Zweck zu dienen, sich Wissen und Fertigkeiten auf Vorrat anzueignen, ohne Anforderungen oder Möglichkeiten, diese kreativ anzuwenden. Hausaufgaben beziehen sich oft auf theoretische Probleme oder Beweise. Deshalb fühlen sich einige Studenten demotiviert. Allerdings erwarten die meisten Studenten auch, dass die Aufgaben in höheren Semestern kreativer werden und sie dann das, was sie gelernt haben, anwenden können. Die Möglichkeit, in der Informatik kreativ sein zu können, motiviert die Studenten.

### 6.3 Kreativität in der Schulformatik aus Schülersicht<sup>95</sup>

Die beiden Studien, welche retrospektive Betrachtungen informatischer Lernprozesse analysieren, bestätigen die wesentlichen Faktoren des Modells aus Sicht des Subjekts. Zusätzlich verdeutlichen sie, dass Defizite in der Unterrichtspraxis bei der Berücksichtigung von Kreativität auftreten. Anhand der Aussagen wird offensichtlich, dass einige Schüler auch unabhängig von der Unterrichtsgestaltung eine kreative Herangehensweise an Informatik entwickeln. Um einen detaillierten Einblick in dieses Verständnis für Informatik und dort präferierte Herangehensweisen und Sichtweisen zu erhalten, wird eine Einzelfallstudie mit einem herausragend kreativen Informatikschüler durchgeführt.<sup>96</sup> Ein Vorteil der konkreten Befragung in einem Interview ist der, dass gezielt auf Details eingegangen werden und bei offenen Fragen nachgehakt werden kann.

#### 6.3.1 Ziel der Untersuchung

Eine zentrale Frage bei der Untersuchung des Faktorenmodells ist die, ob die Faktoren vergleichbar im Erleben der Schüler reflektiert werden. Sichtweisen, Einstellungen und Arbeitsweisen von Schülern, welche sich kreativ in der Informatik engagieren, können Aufschluss darüber geben, welche Aspekte der Informatik sie ansprechen und wie ein möglicher kreativer Zugang zur Informatik aussehen kann.

Zentrale Fragen dieser Untersuchung sind:

Welches Verständnis hat der Schüler von Kreativität (in der Informatik)?

Welche Rolle spielt Kreativität für das eigene Interesse/die Beschäftigung mit Informatik?

Welches Informatikbild entwickelt der kreative Schüler?

Welche Methoden und Arbeitsweisen werden verwendet?

<sup>95</sup> Teile dieses Abschnitts wurden bereits zum Teil sinngemäß und zum Teil wortwörtlich in (Romeike 2006) veröffentlicht.

<sup>96</sup> Eine Hinzunahme weiterer durch Kreativität hervortretender Schüler zur Untersuchung würde den Aussagen dieser Studie möglicherweise mehr Gewicht verschaffen. Dies ist im Rahmen dieser Arbeit leider nicht zu bewältigen.

### 6.3.2 Methode

Um das Modell aus Schülersicht exemplarisch zu untersuchen, wird in einer Einzelfallstudie ein Interview mit einem Schüler analysiert, der durch besonders kreative informatische Leistungen inner- und außerhalb des Unterrichts aufgefallen ist. Anhand der Aussagen des Schülers kann das Modell subjektiv untermauert bzw. widerlegt werden.<sup>97</sup> Ggf. lassen sich hierdurch mögliche Ausprägungen der Faktoren beispielhaft präzisieren und illustrieren.

*In einer Einzelfallstudie wird eine einzelne Untersuchungseinheit (Person [...]) genau erforscht und beschrieben, [...]. Die qualitative Einzelfallbeobachtung hilft dabei, Fragestellungen über individuelle Prozesse und Verläufe zu beantworten. [...] Im Unterschied zu breit angelegten Stichprobenuntersuchungen, die tendenziell viele Untersuchungsobjekte ausschnitthaft betrachten (extensive Forschung), wird in der Einzelfallstudie die Komplexität eines Falles möglichst umfassend und detailliert erfasst (intensive Forschung).*

(Bortz und Döring 2002, S. 323)

Für die Studie wurde ein Schüler aufgrund von Unterrichtsbeobachtungen ausgewählt. Das Interview wurde in Form eines Leitfadenterviews durchgeführt, kurz nachdem der Schüler sein Abitur beendet hatte.

*Das Leitfadenterview ist die gängigste Form qualitativer Befragungen. [...] Es [lässt] genügend Spielraum, spontan aus der Interviewsituation heraus neue Fragen und Themen einzubeziehen oder bei der Interviewauswertung auch Themen herauszufiltern, die bei der Leitfadent-Konzeption nicht antizipiert wurden.* (Bortz und Döring 2002, S. 315)

Während des Interviews stellte sich heraus, dass das Hauptinteresse des Schülers dem Entwickeln von Software galt, weshalb dies das Hauptthema des Interviews darstellte.

### 6.3.3 Ergebnisse

#### Interesse und Motivation

*Vor allem wenn man etwas im Internet macht, kann man mit relativ wenig Aufwand viel erreichen. Das fand ich besonders interessant. [T1S1Z47]<sup>98</sup>*

Praktisches Arbeiten, insbesondere Programmieren, hat den Schüler am Informatikunterricht besonders interessiert. Am praktischen Arbeiten gefällt ihm, dass er dabei kreativ sein kann und mit wenig Aufwand viel erreicht:

*Programmieren ist für mich auch eine kreative Aufgabe, es ist etwas wo man etwas schafft. [T1S2Z35]*

*[Bei anderen Sachen] da fehlt mir so ein bisschen der kreative Aspekt. [T1S2Z31]*

*Dieses Schöpferische finde ich schon sehr wichtig beim Programmieren, dass man eben was schafft. Man bastelt irgendwas zusammen, und am Ende hat man ein Programm. [T1S2Z44]*

Es ist wichtig für ihn, dass er am Ende etwas „in seiner Hand hält“. Die Hauptmotivation ist für ihn die Möglichkeit, sich „ein bisschen selbst zu verwirklichen“. Dieses empfindet er als Kreativität und endet immer in einem Produkt.

#### Verständnis von Kreativität in der Informatik

Für den Schüler ist Programmieren eine kreative Aufgabe. Kreativität beschreibt er als etwas Individuelles.

---

<sup>97</sup> Das Interview wurde bereits durchgeführt, als das Modell noch nicht feststand, weshalb in den Fragen nicht auf alle Aspekte des Modells eingegangen wurde.

<sup>98</sup> Der Code identifiziert ein Zitat. T bezieht sich auf die Nummer des Transkripts, S auf die Seite und Z auf die Zeile des Zitas im Transkript des Interviews.

*Für mich ist eigentlich alles kreativ, was um etwas zu schöpfen seinen eigenen Weg geht. Also, man hat eben dieses Ziel, und man überlegt sich, wie komm' ich da hin, da man sich ja nicht, um kreativ zu sein, das Endziel setzt. [T1S4Z43]*

Dieses Verständnis korrespondiert mit der Definition von P-Kreativität und Problemlösen. Dem Problemlösen fügt es den Aspekt hinzu, dass eine Idee bezüglich des Ziels zwar existiert, dies aber nur eine Richtungsvorgabe und unscharf definiert ist. Dieser kreative Prozess lässt sich nach Erfahrung des Schülers unterteilen.

*Man kann ja auch ein kleines Modul nehmen oder einen kleinen Teil, und am Ende muss dieser Teil das können. Und dann überlegt man sich eben, wie man das hinkriegen kann, und so ist eigentlich jeder einzelne dieser kleinen Schritte schon ein kreativer Schritt. [T1S4Z47]*

Kreativität ist seiner Auffassung nach sehr wichtig in der Informatik. Im Vergleich zu anderen Schulfächern führt er aus, dass Informatik vergleichbar kreativ wie Musik oder Kunst ist, dies aber nicht von der Mehrheit der Mitschüler so gesehen wird.

*Ich [sehe] schon große Parallelen. Dass eben Leute eher Informatik machen, weil sie dort irgendwie kreativ sein können, was sie einfach in anderen Fächern so nicht sein können. Mir z. B. fehlt so die musikalische Kreativität, Kunst sowieso, das krieg ich nicht hin, aber um etwas zu schöpfen und auf die Reihe zu kriegen, das kann ich einfach im Informatikkurs. [T1S7Z33]*

*Es ist natürlich kein musikalisches Fach in dem Sinne, es ist eine naturwissenschaftliche Kreativität, also es ist nicht dieser eng gefasste Kreativitätsbegriff, dass man ein Lied schreibt oder ein Bild malt, aber es ist eine andere Art von Kreativität. [T1S7Z42]*

Es wird deutlich, dass er damit vom traditionellen kunstorientierten Begriff der Kreativität abrückt und ein eigenes Verständnis entwickelt hat, welches Informatik einschließt.

### **Konzepte und Strategien**

*Programmieren ist für mich auch eine kreative Aufgabe, es ist etwas wo man etwas schafft, wo man etwas mit ein paar Bausteinen, die man hat, die man intelligent oder weniger intelligent zusammen baut (...), dann kommt am Ende irgendwas bei raus. Und es gibt eben nicht nur eine Möglichkeit etwas zu implementieren, es gibt meistens hunderte Möglichkeiten; die einen machen das so, die anderen so, das eine ist eleganter, das andere ist eher schneller. [T1S2Z35]*

Bei der Beschreibung von Kreativität in der Informatik weist der Schüler auf das Bausteinprinzip in der Informatik hin. Unter Bausteinen versteht er die grundlegenden Modelle und Konzepte der Informatik.

*Wir haben ja mit Delphi programmiert, mit diesen verschiedenen Modellen, diese Schleifen, die man benutzen kann, Variablen, diese ganzen Standarddatentypen, einfach, die Befehle auf der einen Seite, die man hat und die Variablen und so auf der anderen Seite, den Input, der dann eben verwendet wird. Und eben auch die Konzepte, die dahinter stehen, was man damit machen kann, was so 'ne Schleife bewirkt, das sind eben die Bausteine, mit denen man was machen kann. [T1S3Z5]*

### **Umgebungsfaktoren**

*Vor allem wenn man etwas im Internet macht, kann man mit relativ wenig Aufwand viel erreichen. [T1S1Z47]*

*Man kann sich da auch sehr viel selbst beibringen, [...] allein durch die ganzen Informationen, die man im Internet findet, kann man eigentlich schon eine ganze Menge selbst auf die Reihe kriegen. Ich bin der Meinung, dass man Programmieren sich auch selbst beibringen kann, mehr oder weniger. [T1S2Z1]*

Eine wichtige Rolle spielt für den Schüler die Möglichkeit, Ideen und Lösungsansätze mit Hilfe des Compilers ausprobieren zu können.

*Oftmals ist es einfach nur Ausprobieren. Einfach mal was hinschreiben, so einen Abschnitt, einfach mal reinton und dann ausprobieren und dann gucken was hakt, [...]. [T1S3Z39]*

Statt einem deduktiven strukturierten Vorgehen berichtet der Schüler, dass das Ausprobieren von (Lösungs-)Ideen und anschließendes Nachvollziehen von Debuggerinformationen einen wichtigen Teil seines Problemlösens ausmacht. Planen und strukturierteres Vorgehen findet er nur bei größeren Projekten notwendig und sinnvoll.

Ausführlich äußert er sich über die Problematik der begrenzten Zeit in der Schule:

*Was ich allerdings ein bisschen störend finde ist, dass man in der Schule nur so kurze Zeitabstände hat. Also wenn ich programmiere, dann brauche ich einfach mehr Zeit. Weil so viele Sachen (...), also wenn's ein bisschen komplexer wird (...), die kann man einfach nicht in einer dreiviertel Stunde abarbeiten. Und dann, wenn man sich erst eine Woche oder 4-5 Tage später wieder ran setzt, dann hat man ziemlich viele Ideen oder Sachen wieder vergessen, man startet also nicht direkt wieder da, wo man aufgehört hat, sondern muss sich schon wieder einen Teil aufarbeiten. Mir fällt das schwer. Ich finde es einfacher, wenn ich mich am Wochenende einfach mal ein paar Stunden am Stück an so ein Programm setzen kann und dann auch irgendwas hinkriege. In der Schule kann man auch schon viel schaffen, aber das ist doch störend. Aber das lässt sich auch nicht anders machen. [T1S5Z23]*

### 6.3.4 Fazit

Der Schüler bezeichnet seine Beschäftigung mit Informatik sowie Programmieren per se als kreativ. Er unterstreicht hierbei, wie wichtig dies für ihn ist. Die Aussagen haben insbesondere vor dem Hintergrund Gewicht, dass ihm weder die Intention des Interviews bekannt war, noch ist das Wort „kreativ“ oder „schöpferisch“ vom Interviewer gefallen, ehe dieser Aspekt vom Schüler erwähnt wurde. Ebenso wenig wurde Kreativität bis dato im Unterricht thematisiert.

Der Schüler unterstreicht den zentralen Stellenwert von Produkten in seiner kreativen Tätigkeit und den positiven Einfluss dieser auf seine Motivation. Beeindruckend ist die Detailgetreue, mit welcher der Schüler die Aspekte von Kreativität in der Informatik reflektiert. Offensichtlich ist es möglich, dass diese Aspekte Schülern auch mehr oder weniger von allein bewusst werden können.

Bezüglich der Umgebungsfaktoren zeigen sich viele Gemeinsamkeiten zu den in der o. g. Studie untersuchten Biographien: Das Internet als Wissensressource und Computerfeedback ermöglichen eigenständiges Entwickeln von Projekten; die Umstände in der Schule wirken sich dagegen erstmal negativ auf die kreative Entfaltung aus. Auf diese Weise hat der Schüler in der selbständigen Beschäftigung mit Informatik die kreativen Seiten für sich entdeckt und schätzen gelernt. Diese Sichtweise überträgt sich – abhängig vom Thema – auch auf den Informatikunterricht. Tatsächlich lassen sich durch die Aussagen des Schülers alle Dimensionen des Faktorenmodells belegen.

## 6.4 Sichtweisen und Erfahrungen von Referendaren und Lehramtsstudierenden<sup>99</sup>

Informatiklehrer haben eine besondere Sicht auf Kreativität und Informatik, erleben sie doch in ihrer täglichen Praxis, wie Schüler möglicherweise kreativ, aber auch unkreativ, mit Informatik umgehen. Ziel dieser Studie ist es, die Vorstellungen von Referendaren und Lehramtsstudierenden der Informatik bezüglich Kreativität und verwandter Faktoren zu untersuchen und sie zu ihren Erfahrungen mit Kreativität, in der Rolle als Lehrer und in ihrer eigenen Entwicklung, zu befragen. Die Befragten stehen in ihrer Karriere zwischen Studium und dem Beginn der hauptberuflichen Unterrichtstätigkeit. Durch das Studium sind sie mit einem Repertoire an Unterrichtsmethoden, Ideen und Erfahrungen aus den Unterrichtspraktika ausgestattet. Zusätzlich haben sie in Schulpraktika oder im Referendariat Erfahrungen aus selbständiger Lehrtätigkeit sammeln können und

<sup>99</sup> Teile dieses Abschnitts wurden bereits zum Teil sinngemäß und zum Teil wortwörtlich in (Romeike 2008e) veröffentlicht (Artikel im Anhang).

sind damit in der Lage, Theorie und Praxis zu verbinden. Ein Einblick in ihre Sicht von Kreativität im Bezug auf Informatikunterricht verspricht, Möglichkeiten und Problematiken von Kreativität im Unterricht aufzudecken und damit, Aussagen des Faktorenmodells zu stärken oder einzuschränken. Zusätzlich erlauben sie einen punktuellen Einblick in die Unterrichtswirklichkeit und den Stellenwert, den Kreativität in der Unterrichtspraxis besitzt.

Die Untersuchung stützt sich auf zwei Studien zu Vorstellungen von Lehrern zur Kreativität.<sup>100</sup> Im Folgenden werden nur die im Bezug auf die Überprüfung des Faktorenmodells relevanten Ergebnisse der Untersuchung dargestellt und diskutiert. Die Studie kann detailliert im Anhang nachgelesen werden. Der Fragebogen wurde von 13 angehenden Informatiklehrern ausgefüllt.

#### 6.4.1 Ergebnisse

Die Mehrheit der Studenten/Referendare sieht Kreativität eng verwandt mit der Fähigkeit, eigene Ideen zu entwickeln und/oder zu realisieren (53,8%). In der Begründung der Antwort beziehen sich die Befragten häufig auf kreative Ergebnisse und/oder Produkte im weitesten Sinn (46,2%). Der kreative Prozess wird von der Mehrheit beschrieben als das Beschreiten eines persönlichen neuen Wegs, um ein persönliches Ziel zu erreichen. Diese Sichtweise ist mit der Beschreibung von P-Kreativität vergleichbar, welche Kreativität in einem persönlichen Kontext beschreibt.

Umgebungsfaktoren wurden von den meisten Befragten als entscheidend für die Entwicklung von Kreativität angesehen. Auch geben alle Befragten an, dass Kreativität durch den Lehrer gefördert werden kann und dass dies bei jedem Schüler möglich ist.

61,6% sind der Meinung, dass sich kreatives Denken vom Denken, was beim schulischen Problemlösen verlangt wird, unterscheidet. Sie führen weiterhin aus, dass Schüler häufig dazu angehalten werden, Methoden und Verfahren anzuwenden, die durch den Lehrer demonstriert wurden, um die gleichen Ergebnisse wie der Lehrer zu erreichen. Zu oft gibt es nur ein „richtig“ oder „falsch“. Aus der Perspektive der Kreativität ist dies offensichtlich problematisch.

#### Kreativität und Informatik

Alle Befragten stimmen zu, dass Kreativität und Informatik gut „zusammenpassen“. 84,6% stimmen zu, dass Kreativität in der Informatik in verschiedenen Anwendungsgebieten benötigt wird, 46,2% davon mit maximaler Zustimmung. Sie führen dazu aus, dass für eine Aufgabe in der Informatik verschiedene Lösungen möglich sind, in informatischen Prozessen das Finden von Analogien und Exploration beteiligt und Produkte der Informatik in der Regel die Ergebnisse kreativer Prozesse sind. Beispielsweise wird das Entwerfen von Algorithmen von 84,6% als kreativ bezeichnet. Entsprechend geben 76,9% an, dass man in Informatik mindestens genauso kreativ sein kann, wie in Kunst oder Musik. Die Antworten auf die Frage, welches der Fächer Informatik oder Musik/Kunst mehr Raum für Kreativität bietet, sind allerdings auf der ganzen Antwortskala verteilt.

#### Interesse/Motivation:

Aktivitäten der Informatik, die Kreativität erfordern, wurden in einer Freitext-Frage wie folgt benannt: Problemlösen (46,2%), Entwurf von Benutzerschnittstellen/GUIs (38,5%), Programmieren (30,8%), Algorithmenentwurf (30,8%), Problemfinden (15,4%), Debugging (7,7%) und *Verbinden theoretischer Konzepte mit der Wirklichkeit* (7,7%). Diese Aussagen werden durch Unterrichtserfahrungen illustriert. Dabei berichten die Studenten/Referendare davon, dass sie durch kreative Aufgaben und/oder durch Raum für Kreativität erfolgreich das Interesse der Schüler gewinnen konnten. Dennoch berichten auch einige, dass sie nur selten kreativen Unterricht bei ihren Mentoren beobachten. Vergleichbare Aussagen finden sich auch in Antworten auf die Frage nach eigenen Erfahrungen mit kreativem Informatikunterricht:

<sup>100</sup> Fryer und Collings (1991) und Diakidoy und Kanari (1999).

*Kreativer Informatikunterricht ist möglich, wird aber wenig praktiziert von den Lehrern. Meiner Erfahrung nach sind Schüler erstmal überrascht von den kreativen Ansätzen, lassen sich aber sehr gern darauf ein und erzielen mitunter ganz tolle Ergebnisse, die auf herkömmlichem Wege nicht herausgekommen wären. [Referendar 1]*

### **IKT/Umgebung**

92,3% der Befragten stimmen dem zu, dass Unterricht im Computerkabinett einen positiven Einfluss auf die Kreativität der Schüler ausübt. Die Zustimmung ist allerdings nur schwach. Auf einer Skala von 1 bis 6, mit 1 als stärkster Zustimmung, lag der Durchschnitt der Antworten bei 2,77. Möglicherweise ziehen die Befragten auch die ablenkende Wirkung von IKT in Betracht.

### **6.4.2 Diskussion**

Die Ergebnisse dieser Studie zeigen, dass Lehramtsstudenten und Studienreferendare der Informatik Kreativität sehr schätzen. Kreativität wird vor allem als eine generelle Fähigkeit gesehen, die sich in unterschiedlichen Kontexten und mit verschiedenen Ausprägungen äußert. Außerdem wird Kreativität als ein Weg gesehen, Schüler zum Lernen zu motivieren und ihr Interesse für die Informatik zu entfachen. In vielen Punkten unterstützen diese Ergebnisse das Faktorenmodell für kreativen Informatikunterricht. Dazu gehören Merkmale, an denen man Kreativität identifizieren kann und Umgebungsfaktoren, die Kreativität unterstützen. Die meisten Lehrer beurteilen Kreativität von einem starken persönlichen Blickwinkel. Dies basiert vermutlich auf der Notwendigkeit der Lehrer, Schüler bezüglich ihrer persönlichen Entwicklung beurteilen zu müssen. Vor dem Hintergrund der Förderung von Kreativität ist das ein wichtiger und förderlicher Aspekt. Die Art und Weise, wie die zukünftigen Informatiklehrer Kreativität im Unterricht identifizieren, zeigt eine Sichtweise, die möglicherweise aus der Informatik stammt. Während sich ein großer Teil der Kreativitätsforschung auf Merkmale kreativer Produkte bezieht, betonen die Lehrer eher kreative Prozesse. Typischerweise liegt im Informatikunterricht ein Schwerpunkt auf Problemlöseprozessen. Nur ein kreativer Ansatz führt dabei zu einer besonderen Lösung.

Die Vorstellung von Informatik als kreatives Fach ist eine Voraussetzung für die Berücksichtigung von Kreativität im Informatikunterricht. Alle Säulen des Faktorenmodells finden sich in den Antworten, Begründungen, Ausführungen und Beispielen wieder. Ebenso geben die Lehrer im Bezug auf ihre eigene Entwicklung an, dass Kreativität ein Grund dafür war, sich ursprünglich für Informatik zu interessieren. Das könnte ein Grund dafür sein, warum sie nun Kreativität als einen Weg verwenden wollen, ihre Schüler für Informatik zu interessieren. Für einen Lehrer, der sich nie kreativ mit Informatik beschäftigt hat, wäre solch ein Transfer unmöglich. Vor dem Hintergrund kreativitätsfeindlicher Eindrücke, von denen die Befragten im Bezug auf ihr Studium berichten, sollten Hochschullehrer ihr Bild von der Informatik, welches sie an Universitäten transferieren, überdenken.

Die Vorstellungen der Befragten davon, wie Kreativität in der Informatik gefördert werden kann, entsprechen weitgehend der Forschung. Keiner beschränkt sich ausschließlich auf herausragend kreative Schüler. Basierend auf der Annahme, dass Kreativität bei jedem gefördert werden kann, wurden Methoden genannt, die auf alle Schüler angewandt werden können. Dennoch sind die genannten Methoden wenig differenziert. Die zukünftigen Lehrer sind sich unsicher, wie sie langfristig und strategisch Kreativität einsetzen können und welchen Vorteil dies auch für den Unterricht bringen kann. Hier ist didaktische Unterstützung notwendig und kann das Faktorenmodell ansetzen.

In den Antworten finden sich eine Vielzahl von Beispielen kreativer Erlebnisse mit Schülern und kreativer Produkte. Alle Beispiele berichten gleichzeitig von einem Lernerfolg. Leider weisen die Befragten auch daraufhin, dass Kreativität eher ein seltenes Phänomen ist und dass sie nur selten Kreativität im Unterricht ihrer Mentoren beobachten. Offensichtlich lassen die schulischen Rahmenbedingungen Kreativität nur sehr bedingt zu. Allerdings ist es positiv, dass sich die Befragten davon nicht entmutigen lassen.

Das Verständnis von Kreativität, welches die zukünftigen Lehrer darstellen, ist sicher hinterfragbar und trifft nur bedingt die herausgearbeitete Definition von Kreativität. Einige kreative Ergebnisse, wie sie dargestellt wurden, sind evtl. besser als „produktiv“ zu bezeichnen, da ihnen der Aspekt der Neuheit fehlt. Allerdings bietet dieses pragmatische Verständnis von Kreativität eine niedrige Schwelle und Motivation für Schüler, sich offenen und kreativen Aufgaben zu stellen und damit die Angst vor eigenen neuen Herangehensweisen in der Schule und anderswo zu verlieren.

## 6.5 Fazit der Untersuchung des Faktorenmodells

Basierend auf den Ergebnissen der Studien konnten bereits verschiedene Aussagen getroffen werden, die im Wesentlichen alle die im Faktorenmodell identifizierten Dimensionen und Faktoren stützen. Abschließend sollen die im Kapitel 5 aufgestellten Hypothesen diskutiert werden.

### **Hypothese 1: Im Informatikunterricht stimuliert kreatives Arbeiten Motivation und Interesse der Schüler.**

Alle vier Untersuchungen stützen diese Hypothese. Während kreative Aufgaben in und außerhalb des Informatikunterrichts für Motivation und Interesse bei den Schülern sorgten, wurde teilweise sogar von einem demotivierenden Effekt nichtkreativer Aufgaben berichtet.

Motivation und Interesse gehen im Informatikunterricht in der Regel einher mit persönlichen Herausforderungen, bedeutungsvollen und relevanten Aufgaben, mit denen sich die Schüler identifizieren können, und Originalität bei der Ausführung der Tätigkeiten.

### **Hypothese 2: Im Informatikunterricht wird Kreativität aufgrund des kreativen Charakters des Fachs Informatik gefördert.**

Die Titulierung und Wahrnehmung von Informatik als kreatives Unterrichtsfach konnte in allen Studien bestätigt werden. Hierbei sind es vor allem informatische Prozesse, die Experimentieren, Exploration und unabhängiges und selbständiges Arbeiten erlauben und ein Gefühl von Kreativität vermitteln. Ebenso bestätigt sich die Betonung informatischer Produkte als Ziel kreativer Prozesse und deren positiver Einfluss auf Motivation und Interesse der Schüler. An verschiedenen Stellen wurde verdeutlicht, dass Projekte als Unterrichtsmethode kreatives Arbeiten begünstigen. Der Aspekt des Bausteinprinzips wurde nur im Schülerinterview explizit als kreativitätsförderndes Element der Informatik erwähnt und die Vorteile für kreatives Arbeiten in der Informatik ausführlich beschrieben. Dass dieser Aspekt in den anderen Studien nicht sichtbar wurde, könnte darauf zurückzuführen sein, dass in keiner der Studien explizit zu diesem Thema Daten erhoben wurden.

### **Hypothese 3: IKT fördern Kreativität im Informatikunterricht.**

Als deutlichstes kreativitätsförderndes Medium stellt sich das Internet dar. Für individuelles zielstrebiges kreatives Arbeiten kommt dem Internet als Wissensquelle eine zentrale Bedeutung zu. Mehrfach erwähnen die Befragten das Internet auch als willkommene Möglichkeit, ihre Produkte zu präsentieren und von anderen Personen Feedback zu erhalten. Darüber hinaus wurden Programmierumgebungen/Compiler als Quelle von Feedback und damit als „Partner“ beim Problemlösen charakterisiert. Diese ermöglichen den Schülern individuelles unabhängiges Experimentieren und ein schrittweises Entwickeln von Lösungen.

Der schulische Informatikunterricht wurde in der Mehrzahl der Aussagen als ein Umfeld charakterisiert, welches kreatives Arbeiten nicht unterstützt. Einige Studenten und Referendare berichten allerdings auch von positiven Erlebnissen, die für die Zukunft hoffen lassen.

Im folgenden Kapitel wird eine Operationalisierung des Faktorenmodells vorgenommen, die dabei helfen soll, Unterricht bzw. Unterrichtsphasen in einer Art und Weise zu gestalten, dass im Informatikunterricht Kreativität angeregt, gefördert und gefordert wird.

## 7 Kriterien kreativen Informatikunterrichts<sup>101</sup>

Bis hierher wurden die Faktoren, die Kreativität im Informatikunterricht maßgeblich beeinflussen, herausgestellt, analysiert und anhand verschiedener Studien untermauert und illustriert. Für die Umsetzung eines kreativen Informatikunterrichts und die Evaluation von Unterricht müssen diese Faktoren nun konkretisiert werden. Im Folgenden werden auf dem Faktorenmodell basierende Kriterien formuliert, die für einen kreativen Informatikunterricht maßgeblich sind. Die Kriterien werden jeweils begründet und teilweise mit einem für das Kriterium spezifischen Beispiel illustriert. Abschließend werden die Kriterien zu einer Analyse von in der LOG IN veröffentlichten Unterrichtsbeispielen hinsichtlich der Berücksichtigung und des Stellenwerts von Kreativität herangezogen.

### 7.1 Berücksichtigung der Fach-Dimension

In dieser Dimension werden Kriterien aufgestellt, die sich aus den Möglichkeiten, die das Fach Informatik für Kreativität bietet, ergeben.

Kreative Leistungen sind nur in eigenständiger Arbeit möglich, welche grundsätzlich durch die zentrale Stellung von Projekten im Informatikunterricht begünstigt wird. Da selbständige Arbeitsphasen im Unterrichtskontext durch den Unterrichtsverlauf bzw. den Lehrer einen Rahmen in Form von Themen und Aufgabenstellungen oder Arbeitsaufträgen bekommen, sind diese hinsichtlich ihres kreativen Potentials genauer zu analysieren.

#### Problemlösung oder Produkt

Im Mittelpunkt der Informatik steht der Prozess der Softwareentwicklung und damit auch das Erschaffen von Softwareprodukten. Natürlich kann sich nicht der gesamte Informatikunterricht auf Problemlösung und das Modellieren von Software beschränken: Das Erwerben von Wissen, vor allem von Konzeptwissen, bildet die Grundlage für jede kreative Tätigkeit. Als kreative Unterrichtsphase sollte das Gestalten eines Produkts oder einer Problemlösung allerdings explizit in der Planung berücksichtigt werden, um dem gelernten Wissen einen konkreten und kreativen Anwendungsbezug zu geben.

#### Subjektive Neuheit

Neuheit ist ein wichtiges Kriterium kreativer Leistungen. In der Schule dürfte allerdings nur selten eine absolut neue Lösung von einem Schüler entwickelt werden. Dennoch kann jeder Schüler *für sich* neue (p-kreative) Produkte und Lösungen entwerfen, wenn ihm bezüglich der Bearbeitung einer Aufgabenstellung kein Lösungsweg oder Muster bekannt ist oder vorliegt. Nicht erfüllt wird dieses Kriterium von der Aufgabe (Negativbeispiel): „Verschlüssele folgende Nachricht mit der (bekannten) Caesar-Verschlüsselung!“ Erfüllt wird dieses Kriterium durch die Aufgabenstellung (Positivbeispiel): „Denke Dir ein eigenes Verfahren zum Verschlüsseln eines Textes aus!“

#### Offenheit

Charakteristisch für kreative Prozesse sind Bestandteile des Problemfindens, Explorierens und Entdeckens. So kann bei einer offenen Aufgabenstellung eine ungefähre Zielvorstellung vorhanden sein, die aber nicht klar definiert ist und erst im Prozess festgelegt wird. Stattdessen sind im Informatikunterricht oft Rahmenbedingungen einzuhalten, welche die Richtung vorgeben und durch den Lehrer oder die Aufgabenstellung bestimmt sind. Der Schüler muss sich dann seine Möglichkeiten bewusst machen und sein Betätigungsfeld abstecken. Die Möglichkeit, die Aufgabe selbst mitzugestalten, wirkt zusätzlich motivierend. Positivbeispiel: „Wende deine Kenntnisse aus der Kryptologie bei der Erstellung eines Informatiksystems an!“ Zur Evaluation von Unterricht

---

<sup>101</sup> Dieses Kapitel wurde bereits veröffentlicht als Teil des Artikels (Romeike 2007b).

oder wenn eine offene Zielstellung nicht möglich oder unpraktikabel ist, lassen sich hinsichtlich der Offenheit Abstufungen bzgl. des Bearbeitungswegs und des Ziels vornehmen.

### **Offene Ergebniserwartung**

Auch bei festgelegtem Ziel ist es möglich, unterschiedliche Ergebnisse zu erhalten, wenn diese die Anforderungen erfüllen (divergente Aufgaben). So kann bspw. den Schülern die Wahl gelassen werden, verschiedene Parameter begründet oder nach Belieben gegenüber einem empfohlenen Lösungsweg auszuwählen oder zu variieren. Dieses Vorgehen führt zu einer begrenzten Vielfalt an Lösungen. Schülern bietet sich dadurch immerhin die Möglichkeit, auch andere Lösungen herauszufinden und den Lösungsraum zu erkunden.

### **Offener Lösungs-/Bearbeitungsweg**

Zum kreativen Prozess in der Informatik gehört das Auswählen aus verschiedenen Vorgehensweisen, das Anwenden von Algorithmen, Konzepten und Modellen und das bewusste Entscheiden, wie eine Problemlösung/ein Produkt erarbeitet werden soll – nur Aufgaben, die verschiedene Wege zulassen, ermöglichen den Schülern Gestaltungserfahrungen.<sup>102</sup>

### **Bearbeitungstiefe**

Engagieren sich Schüler kreativ, arbeiten sie selbstgesteuert. Sie entscheiden, wie umfangreich, wie lange und wie intensiv sie sich mit einem Gegenstand/einer Aufgabenstellung auseinandersetzen möchten und welche Qualität ihr Produkt besitzen soll.<sup>103</sup> Dabei kann eine intensive Beschäftigung bis hin zum Forschen auftreten. Um dieses Engagement nicht zu unterbinden oder durch mangelnde Vertiefungsmöglichkeiten abzuschneiden, sollte eine Aufgabe unterschiedliche Bearbeitungstiefen zulassen, z. B. indem das angestrebte Ziel Erweiterungen oder Veränderungen/Optimierungen zulässt. Bei Implementationen steigt die Bearbeitungstiefe z. B. mit der Optimierung von Algorithmen, der Berücksichtigung möglicher Fehleingaben oder umfangreicheren Ausgaben. So lassen sich auch unterschiedliche Leistungsniveaus berücksichtigen.

### **Konzeptwissen**

Ein solides Grundwissen im Betätigungsbereich ist Voraussetzung für jeden kreativen Prozess, da hierauf begründete Modellierungs- und Problemlöseentscheidungen basieren. Faktenwissen und Produktwissen sind in dem Zusammenhang zwar ebenfalls notwendig, aber erst das Anwenden von dahinter stehenden Zusammenhängen und Konzepten ermöglicht ein problemübergreifendes kreatives Denken. Folglich ist insbesondere Konzeptwissen zu vermitteln und den Schülern die Möglichkeit zu bieten, dieses anzuwenden. Informatiktypisch und für Kreativität hilfreich ist das Verwenden der Bausteinmetapher für die Darstellung vieler Konzepte der Informatik. Bspw. ermöglicht die Kenntnis von Substitution und Transposition bei Verschlüsselungsverfahren gegenüber alleinigem Anwendungs-/Algorithmenwissen, die jeweiligen Eigenschaften geschickt für ein eigenes Verfahren der Kryptographie anzuwenden und zu beurteilen.

### **Ideenanregung**

Einer kreativen Leistung geht zumeist ein Stimulus voraus. Art, Inhalt, Formulierung und/oder Hintergrund einer Aufgabenstellung können eine solche Anregung darstellen. Die Bedeutung hierfür wird recht schnell deutlich, wenn man z. B. Aussagen von Komponisten betrachtet: Hier ist häufig ein Gefühl, eine Begegnung, ein Erlebnis oder ein Eindruck von außen ausschlaggebend für die Inspiration. Auch im Informatikunterricht soll die Aufgabe dem Schüler reichlich Anstoß geben, Ideen zu entwickeln, z. B. durch den Kontext, in den sie eingebettet ist oder durch das Anknüpfen an die Erfahrungswelt der Schüler. Das bedeutet, dass der Schüler sich vorstellen kann,

<sup>102</sup> Auch konvergente Aufgaben können hier hinzuzählen, falls verschiedene Lösungswege möglich sind.

<sup>103</sup> "Creativity also involves quality of work" (Sternberg 2003, S. 336).

wozu bspw. seine zu entwerfende Software eingesetzt werden kann und welche „größeren“ Probleme sie löst.

## 7.2 Berücksichtigung der Schüler-Dimension

Kreative Beschäftigung kann einen Menschen begeistern, an die Aufgabe fesseln und/oder in einen Flow-Zustand (vgl. Kapitel 2, Abschnitt 2.2) versetzen, in welchem er voll in seiner Betätigung aufgeht. Ziel kreativer Unterrichtsphasen sollte es sein, einem solchen Empfinden oder einem solchen Zustand möglichst nahe zu kommen. Hierzu zählt auch, eine positive Einstellung zur Kreativität zu etablieren, welche durch folgende schülerorientierte Anforderungen begünstigt wird.

### Relevanz

Eine zentrale Voraussetzung für einen kreativen Unterricht ist ein Unterrichtsgegenstand, der für die Schüler relevant und interessant ist und ansprechend motiviert wird. Interesse und besonders intrinsische Motivation sind entscheidende Faktoren für kreative Leistungen. Es ist demnach notwendig, dass die Schüler vom Thema, mit dem sie sich beschäftigen sollen, angesprochen werden. Hierzu sollte das Thema aus der Lebenswelt der Schüler stammen und/oder so aufbereitet sein, dass der Schüler einen persönlichen Bezug dazu herstellen kann. Im Informatikunterricht sollte das leicht sein, wenn ein Ausschnitt der Realität modelliert wird. Möglichst konkret statt abstrakt bedeutet dann auch, wenn möglich auf „Zahlenbeispiele“ zugunsten von „greifbaren“ Beispielen zu verzichten.

### Identifikation

Um möglichst stark in seiner Arbeit aufzugehen, soll sich der Schüler mit seiner Beschäftigung identifizieren können. Der Unterrichtsinhalt muss damit für den Schüler eine Bedeutung besitzen oder zumindest eine Bedeutung erlangen können. Verantwortung für einen Teil eines Softwareprojekts übernehmen zu können, das Gefühl als Experte eingesetzt zu werden und die spätere Möglichkeit, seine Lösung auch präsentieren zu dürfen, können dies unterstützen.

### Originalität

Jeder Schüler ist ein eigenes Individuum mit eigenen Ansprüchen, Vorstellungen und Vorlieben. Originalität als Kriterium kreativer Leistungen bedeutet in diesem Zusammenhang, dass sich ein Schüler einen Originalitätsanspruch setzen und erfüllen kann, bspw. indem er seiner Lösung/Bearbeitung eine „eigene Note“ verleiht. Diese kann ästhetische, funktionale, gewitzte oder andere Besonderheiten ausmachen: z. B. durch die Gestaltung einer GUI, einmalige Programmfunktionen oder spezielle Anwendungsgebiete.

## 7.3 Anforderungen an die Unterrichtsumgebung<sup>104</sup>

Kreatives Arbeiten ist immer auch von der Umgebung abhängig. Ein negatives Unterrichtsklima kann ein deutlicher Kreativitätshemmer sein. Entsprechend sollte der Lehrer (wie in jedem Unterricht) darauf achten, dass die Schüler sich akzeptiert und wohl fühlen. Eine Besonderheit stellen hier die Informatikunterrichtslabore dar: Durch die Arbeit am Computer mit Entwicklungs- und Simulationsumgebungen wird den Schülern Experimentieren ermöglicht, sie erhalten direktes (Compiler-)Feedback und können ihre Ergebnisse meist direkt betrachten und analysieren. Darüber hinaus stellen kreative Unterrichtsphasen weitere Anforderungen an die Umgebung, um Kreativität zu stimulieren.

---

<sup>104</sup> Im Folgenden werden aufgrund der Praxisrelevanz auch fachübergreifende Anforderungen mit einbezogen, die sich aus der Kreativitätsforschung ergeben, aber nicht explizit im Faktorenmodell verdeutlicht sind.

## Experimentieren

Kreativ tätig sein bedeutet, mit Ideen zu experimentieren, Heuristiken anzuwenden und Lösungsmöglichkeiten zu testen. Ist das Lösungsfinden durch Versuch und Irrtum auch ein Vorgehen, das methodisch im Informatikunterricht oft nicht präferiert wird, so gehört es doch zum kreativen Prozess dazu und ermöglicht gerade in der Softwareentwicklung das Aufstellen und Testen von Hypothesen für kleinere Probleme. Experimentieren schließt hierbei nicht das „Nach-Experimentieren“ gemäß vorgegebener Versuchsanleitungen mit ein, sondern meint das selbstständige Untersuchen und Prüfen von Ideen und Hypothesen.

## Zeitlicher Raum

Kreativität ist unter Druck nur schwer realisierbar. Zur Überprüfung und Realisierung von Ideen sowie für die Illumination von Gedankenansätzen wird Zeit benötigt. Da die Schüler in kreativen Unterrichtsphasen selbstgesteuert arbeiten, teilen sie sich ihre Zeit auch selbst ein; Informatikprojekte begünstigen dieses Kriterium.

## Unterrichtsklima der Vielfalt

So, wie zeitlicher Druck negative Einflüsse haben kann, sind auch Konformitätsdruck (Gruppendenken), erwartete Perfektion (Suche nach der erwarteten Antwort), Hierarchien und frühe Evaluation Kreativitätshemmer. Stattdessen sollte der Unterricht gegenseitige Anregungen und Inspiration ermöglichen. Neue Ideen sollten willkommen sein, Misserfolge ermöglicht und vielfältige Lösungen begrüßt und respektiert werden. In kaum einem anderen Unterrichtsfach als Informatik ist es möglich, so viele verschiedene Ergebnisse zu erhalten. Diese sind vorzustellen und zu fördern, auch wenn eine Software am Ende nicht „läuft“.

## Coaching

Während im traditionellen Unterricht der Lehrer den Unterricht leitet, Wissen vermittelt und die Schüler korrigiert und bewertet, besteht in kreativen Unterrichtsphasen seine Aufgabe im Coaching: Sollten Schüler ein nur schwer zu überwindendes Problem haben, nicht weiter wissen oder eine Inspiration benötigen, kann der Lehrer helfen, ansonsten hält er sich – vor allem auch mit wertenden Äußerungen – zurück. Zuspruch und Motivation sind allerdings legitim und notwendig. Damit sollte der Lehrer auch ausdrücken, dass er Kreativität begrüßt und wertschätzt. Er berät, begleitet, informiert und unterstützt wo erforderlich und erwünscht.

## 7.4 Kreativität in Unterrichtsbeispielen der Informatik

Mit Hilfe der aufgestellten Kriterien kreativen Informatikunterrichts soll nun untersucht werden, inwieweit Kreativität im Informatikunterricht bereits in Unterrichtsbeispielen der LOG IN berücksichtigt wird. Da eine explizite Thematisierung von Kreativität in der LOG IN bisher noch nicht erfolgte, wird im Folgenden untersucht, ob sich kreative Aspekte des Informatikunterrichts implizit hierin wiederfinden. Hierzu wurden 144 Unterrichtsvorschläge, die innerhalb der Jahre 1995-2006 vorgestellt wurden, auf Indizien analysiert.<sup>105</sup>

### 7.4.1 Zum Vorgehen

Die Unterrichtsbeispiele beinhalten meist eine inhaltliche und eine methodische Ebene. Für die Untersuchung waren vor allem die methodischen Hinweise, die Unterrichtsempfehlungen sowie – wenn dargestellt – die (Teil-)Lernziele interessant. Schüleraktivität wird in der Regel durch Arbeitsaufträge oder Aufgaben gesteuert. Solche durch Unterrichtsbeispiele suggerierte Schüleraufgaben wurden genauer betrachtet. Ein großer Teil der Unterrichtsskizzen bezog sich allerdings auf Inhalte und deren didaktische Aufbereitung. Bzgl. des Aufbaus der Vorschläge stellten

<sup>105</sup> Die Untersuchung wurde mit Unterstützung einer Gruppe von Lehramtsstudenten des Hauptstudiums durchgeführt, nachdem die Kriterien ausgiebig besprochen wurden.

sich folgende Fragen: Lässt die vorgeschlagene Vorgehensweise die Erfüllung der Kreativitätskriterien zu? Bleibt im Kontext des dargestellten Unterrichts Raum für kreative Phasen? Die Analyse brachte folgende Ergebnisse.<sup>106</sup>

#### 7.4.2 Kriterien der Fach-Dimension

Nur gut die Hälfte der Unterrichtsvorschläge lässt den Schülern die Möglichkeit, selbst *problemlösend oder gestaltend* tätig zu werden. Hier scheinen viele Lehrer v. a. dem Verstehen von Informationssystemen und dem Aneignen von Wissen einen Vorzug zu geben. Indikatoren für solche Unterrichtsvorschläge sind formulierte rezeptive Lernziele der Art: „*die SuS<sup>107</sup> wissen, dass...*“, „*die SuS erkennen, dass...*“ oder „*die SuS vollziehen nach, wie...*“. Erfüllt wurde dieses Kriterium z. B. von Unterrichtsbeispielen zur Kryptologie oder zur Gestaltung von Internetpräsentationen.

*Subjektiv neue* Aufgaben werden in 57% Prozent der Unterrichtsskizzen gestellt. Dieser Wert korrespondiert mit der Art der Aufgabenstellungen: Sollen die Schüler nicht problemlösend oder gestaltend tätig werden, besteht ihre Aufgabe wohl darin, den Lernstoff an Aufgaben zu festigen, die zu den behandelten analog sind, bzw. an ähnlichen Aufgaben nachzuvollziehen. Mitunter würden einfache Modifikationen reichen, um einer Aufgabenstellung Neuigkeitswert zu geben, aber dennoch Gelerntes anwendbar zu machen: „*Definiere ausgehend von der hier beschriebenen Ameisenwelt eine Hamsterwelt gemäß folgender Vorstellung [...]*“ (Prätorius 2004). Immerhin zwei Drittel der Unterrichtsskizzen zielen auf die Vermittlung und Anwendung von *Konzeptwissen*. Dies bedeutet bspw. die Vermittlung der TextverarbeitungsDarstellungen „kursiv“ oder „fett“ als Attribute des Objekts „Zeichen“. Die Vermittlung einer Handlungsfolge zum „Kursiv-machen“ eines markierten Wortes bewirkt dagegen prozedurales Wissen. Gut die Hälfte der Unterrichtsskizzen lässt eine variable *Bearbeitungstiefe* zu und immerhin die Hälfte regt die Schüler zum Einbringen eigener Ideen an. So bietet eine Unterrichtssequenz zu interaktiven Animationen eine Bandbreite an Möglichkeiten, eigene Ideen einzubringen, während „Beobachtungen an Bildschirmen“ wenig Einfluss zulassen.

Der Grad der *Offenheit* einer Aufgabe ist ein entscheidender Anhaltspunkt dafür, wie viel Kreativität verlangt wird und eingebracht werden kann. Aufgaben mit offener Zielstellung und damit mit Möglichkeiten des Problemfindens regen nur ein Fünftel der vorgestellten Unterrichtssequenzen an (i. d. R. Aufgaben zum Modellieren oder Programmieren in einem bestimmten Themenbereich). Verschiedene Ergebnisse sind nur bei 44% der Aufgaben möglich, verschiedene Bearbeitungswege bei knapp der Hälfte. Vor dem Hintergrund, dass Modellieren und Problemlösen einen zentralen Stellenwert in der Informatik besitzt und wohl jeder dieser Prozesse in seiner Durchführung variabel ist, scheint dieses Ergebnis unbefriedigend. Konvergente Aufgaben mit strikten Lösungsmustern scheinen den Informatikunterricht zu dominieren.

#### 7.4.3 Kriterien der Schüler-Dimension

Fast drei Viertel der Unterrichtsskizzen wählten als Unterrichtsgegenstand Themen, die für die Schüler *relevant* sein dürften<sup>108</sup> und motivierten diese ansprechend. Publizierende Lehrkräfte sind sich offenbar dem wichtigen Einflussfaktor von Motivation auch auf den allgemeinen Unterricht bewusst. Gezielt werden in vielen Beispielen schülerrelevante Themen ausgewählt, die von Gefahren im Internet bis hin zu gut motivierten Graphenproblemen reichen. Dieses Kriterium wurde z. B. nicht erfüllt von Unterrichtsvorschlägen zur Turingmaschine, zu Teleheimarbeit oder zur Datenmodellierung<sup>109</sup>.

Eine Auswertung des Kriteriums, ob sich ein Schüler mit einem Thema *identifizieren* kann, ist problematisch. Während der unterrichtende Lehrer ein Gefühl für die Interessen und Bedürfnisse

<sup>106</sup> Da es sich um eine indizienbezogene Auswertung handelt, werden die Anteilsangaben nur grob dargestellt.

<sup>107</sup> Schülerinnen und Schüler.

<sup>108</sup> Die tatsächliche Relevanz ist natürlich vom Individuum abhängig. Bei der Beurteilung wurden die Fragestellungen zu Grunde gelegt: Stammt das Thema aus der Lebenswelt eines Schülers? Ist das Thema schülerorientiert?

<sup>109</sup> Was nicht bedeutet, dass es unmöglich ist, diese Themen entsprechend zu motivieren.

seiner Schüler hat und diese bei der Berücksichtigung des Kriteriums in der Planung des Unterrichts einbeziehen kann, ist dies bei der Auswertung von Unterrichtsvorschlägen schülerunabhängig nur schwer durchzuführen. Die zugrunde liegende Frage bei der Auswertung war entsprechend: „Ist es grundsätzlich möglich, sich mit dem Unterrichtsgegenstand bzw. der eigenen Tätigkeit im Rahmen dieses Unterrichts zu identifizieren?“ Offensichtlich ist das der Fall bei Modellierungen, Problemlösungen oder Themen, welche die Schüler direkt betreffen. Bei der Behandlung von z. B. Protokollen ist das nicht ohne weiteres möglich. 41% der ausgewerteten Themen ließen in diesem Sinn eine Identifizierung zu. Das Einbringen einer „*eigenen Note*“ ermöglichten 39% der Unterrichtsvorschläge, welches häufig durch das Gestalten einer GUI, einer Präsentation o. ä. geschah.

#### **7.4.4 Kriterien der Unterrichtsumgebung**

Gut die Hälfte der Unterrichtsskizzen gibt den Schülern die Möglichkeit zu *experimentieren* – hier spiegelt sich die Durchführung des Informatikunterrichts in Computerkabinetten wider, welche softwarebasierte Experimentier- und Simulationsumgebungen bereitstellen. Entsprechend beziehen sich auch einige Unterrichtsskizzen auf die Anwendung solcher Software. Ein gewisser *zeitlicher Freiraum* wird den Schülern in immerhin 41% der Unterrichtsskizzen zugestanden. Dieses geschieht meist durch das Einbinden von (Mini-) Projekten oder längeren Schülerarbeitsphasen. Eine Mehrzahl der Lehrer hält an der traditionellen *Lehrerrolle* fest – in nur 38% gibt der Lehrer phasenweise diese Rolle ab und tritt in die Rolle des Coachs. Ein *Unterrichtsklima der Vielfalt* ist in nur einem Drittel der Unterrichtssequenzen vorstellbar. Möglicherweise offenbart sich hier ein Bedürfnis der Lehrkräfte nach konvergenten, leicht zu überschauenden und zu bewertenden Lösungen sowie einer möglichst homogenen Schülerschaft, welche den Unterricht einfacher machen.

#### **7.4.5 Fazit**

Insgesamt ist festzustellen, dass eine Mehrheit der untersuchten Unterrichtsvorschläge einem kreativen Informatikunterricht noch wenig Beachtung schenkt. Es spiegelt sich hierin allerdings auch wider, dass Informatikunterricht einen Teil der Anforderungen für kreatives Arbeiten per se schon anwendet, sei es durch das Arbeiten an Computern, welche kreatives Arbeiten begünstigen, oder durch den zentralen Stellenwert von Problemlösen und Modellieren. In nicht wenigen Unterrichtsskizzen wäre allerdings durch eine Änderung der Aufgabenstellung und durch die Berücksichtigung der Kriterien ein stärkeres Einbinden kreativer Unterrichtsphasen möglich.

## 8 Anwendung in einem Unterrichtsbeispiel

In den vorausgegangenen Kapiteln wurde ein auf theoretischen Erkenntnissen basierendes Modell entwickelt und durch verschiedene Studien plausibilisiert sowie durch Kriterien konkretisiert. Zur Verdeutlichung des Faktorenmodells und der Anwendung der Kriterien kreativen Informatikunterrichts wird in diesem Kapitel ein Unterrichtsbeispiel beschrieben, welches das Faktorenmodell und die Kreativitätskriterien anwendet und zum Ziel hat, damit die Kreativität der Schüler in den Unterrichtskontext einzubinden und zu nutzen. Das Unterrichtsbeispiel dient im Weiteren einer empirischen Untersuchung der in Kapitel 5 aufgestellten Hypothesen in der Praxis.

### 8.1 Überlegungen zur Einführung in die Programmierung<sup>110</sup>

Programmierung ist ein problematisches Thema im Informatikunterricht. Die Vielzahl der Veröffentlichungen und unterschiedlichen Ansätze zur Vermittlung der Grundlagen der Programmierung verdeutlichen die Brisanz und Unsicherheit des Themas, bei welchem Lernende häufig Misserfolge einstecken müssen (vgl. auch Kapitel 4). Da Programmieren ein großes Potential zur kreativen Entfaltung bietet, soll Kreativität in diesem Beispiel zur Einführung in die Programmierung von Anfang an mit einbezogen werden. Die inhaltlichen Ziele der Unterrichtssequenz sind in Abbildung 31 zusammengefasst.

Die Unterrichtssequenz verfolgt in Anlehnung an die Bildungsstandards Informatik die Umsetzung der folgenden Lernziele und das Herausbilden folgender Kompetenzen der Schülerinnen und Schüler:

1. Sie entwickeln ein grundlegendes Programmierverständnis
2. Sie beschreiben die Eigenschaften von Algorithmen, nennen Beispiele für erfüllte und verletzte Eigenschaften und entscheiden, ob ein Problem algorithmisch lösbar ist.
3. Sie verwenden algorithmische Grundbausteine zur Programmierung
  - Sequenz, Schleife, Entscheidung, Variablen (lokale/globale)
  - Eingabe und Ausgabe von Information
  - Rechenoperationen und Vergleichsoperatoren
  - Objekt, Botschaft, Attribut, Methode
4. Sie stellen Algorithmen dar
  - mit Scratch-Bausteinen
5. Sie verstehen von Algorithmen und Programmtexte:
  - Lesen und analysieren Programme
  - Implementieren, modifizieren und erweitern Programme
6. Sie entwerfen, implementieren und testen Programme
7. Sie lösen Probleme selbstständig
8. Sie verwenden Hilfesysteme selbstständig

Darüber hinaus werden folgende Prozesskompetenzen angesprochen:

- Modellieren und implementieren
- Begründen und Bewerten
- Strukturieren und Vernetzen
- Kommunizieren und Kooperieren
- Darstellen und Interpretieren

Abbildung 31: Lernziele der Unterrichtssequenz.

<sup>110</sup> Zielgruppe dieses Beispiels ist ein Grundkurs Informatik der 11. Klasse eines Gymnasiums.

### Zur Auswahl des Unterrichtswerkzeugs

Als Programmiersprache und Tool zur Unterstützung von Kreativität wird die visuelle Programmiersprache Scratch (vgl. Maloney 2004) verwendet. Wie bereits in Kapitel 3 verdeutlicht wurde, erfüllt Scratch Shneidermans Kriterien für kreative Softwaretools und ist damit für den genannten Zweck bestens geeignet.

## 8.2 Ein kreativer Einstieg in die Programmierung

Die Umsetzung der Kriterien kreativen Informatikunterrichts und der Anforderungen des Challenge-Cycles wird durch Berücksichtigung folgender Aspekte erreicht.

### Motivation

Das Motivieren der Schüler ist ein wesentlicher Teil des Unterrichtens und Grundvoraussetzung für kreatives Lernen. Die Aufmerksamkeit der Schüler und die Förderung der Schülermotivation sollen angesprochen werden, indem die Relevanz und Anwendbarkeit der zugrunde liegenden Konzepte verdeutlicht werden und durch die Auswahl von Themen, zu denen die Schüler einen Bezug aufbauen können, z. B. durch das Animieren einer Geschichte aus ihrem Leben oder durch das Entwickeln auf eigenen Ideen basierender Spiele.

### Betonung von Konzepten

Neue Konzepte und Inhalte werden eingeführt und veranschaulicht mit Hilfe der Baustein-Metapher. Eigenschaften, Anwendungen und Vernetzungen der grundlegenden Programmierkonzepte werden im Vorhinein durch den Lehrer erklärt, von den Schülern selbst erarbeitet oder von den Schülern in der praktischen Auseinandersetzung mit der Programmierung selbst entdeckt. Hilfreich ist hierbei die visuelle Darstellung von Programmierkonzepten in Scratch als Bausteine, die nur in syntaktisch richtiger Verwendung kombiniert werden können. Methodisch werden die Konzepte entweder im Unterrichtsgespräch, durch Arbeitsblätter oder innerhalb der Schülerpräsentationen eingeführt.

### Inspiration

Grundlegend für kreativen Unterricht ist es, Schüler zu inspirieren. Dies geschieht vor allem durch Beispiele und Brainstorming über die vorhandenen Möglichkeiten. Hierdurch wird die Kreativität der Schüler angeregt. Zusätzlich lernen sie abzuwägen zwischen dem, was sie erreichen wollen und dem, was sie mit den gelernten Konzepten und den Möglichkeiten der Programmiersprache erreichen können.

### Challenge

Das Finden einer eigenen Challenge wird den Schülern ermöglicht, indem offene Aufgaben gestellt werden, die sich variabel komplex bearbeiten lassen. Aufträge geben grundsätzlich eine Richtung und einen generellen Rahmen von dem vor, was geschaffen werden soll. Damit müssen die Schüler Probleme lösen, die sie sich zuvor präzisieren und definieren („Was will ich tun?“). Für Aufträge gibt es keine „richtige“ Lösung (Offenheit) und Lösungen können im Rahmen der Unterrichtszeit – oder darüber hinaus – von den Schülern entsprechend dem eigenen Können und Interesse optimiert und erweitert werden, bspw.:

*Animiere deinen Namen, indem die Buchstaben auf Mausbewegungen und/oder Mausklicks reagieren.*

Hierdurch können sich die Schüler mit den neu gelernten Konzepten vertraut machen, die Programmierumgebung weiter erkunden, Lösungen für ihre Ideen finden, diese umsetzen und ausprobieren/testen.

### **Präsentation und Reflexion**

Die (Mini-)Projekte werden mit dem Upload der Programme auf einen Internetserver (Scratch-Webpage) und der Vorstellung einiger Ergebnisse im Kurs zu Beginn der jeweils nächsten Unterrichtsphase abgeschlossen. Bei den Präsentationen werden jeweils Ideen, Probleme und Strategien vorgestellt und diskutiert. Wenn Schüler in ihren Lösungen neue Konzepte entdecken und anwenden, werden diese dem restlichen Kurs am Beispiel verdeutlicht.

Aus Platzgründen wird auf die weitere Darstellung des Unterrichtsbeispiels an dieser Stelle verzichtet. Die gesamte Unterrichtssequenz und die Erfahrungen einer Erprobung können im Anhang nachgelesen werden.

## 9 Evaluation des Unterrichtsbeispiels

Zur Überprüfung der Durchführbarkeit eines kreativen Informatikunterrichts wurde das beschriebene Unterrichtsbeispiel in einem Grundkurs Informatik der 11. Klasse eines Gymnasiums erprobt. Die Evaluation der Erprobung folgte zwei Zielen: Der Evaluation der Unterrichtssequenz und der Überprüfung der in Kapitel 5 aufgestellten Hypothesen in der Praxis.

Darüber hinaus soll das Bild, das die Schüler von der Informatik hinsichtlich der Rolle von Kreativität in der Informatik entwickeln, nach Abschluss der Unterrichtssequenz genauer untersucht werden.

### 9.1 Auswertung der Unterrichtssequenz<sup>111</sup>

Die Auswertung der Unterrichtssequenz konkretisiert sich an folgenden zu beantwortenden Fragen:

1. Wie wirkt sich der kreativitätsorientierte Unterricht auf die Motivation und das Interesse der Schüler aus?
2. Wie wirkt sich der kreativitätsorientierte Unterricht auf das Verständnis der thematisierten Inhalte (und damit auf die Leistungen) aus?

#### 9.1.1 Zum Vorgehen

Um vergleichende Ergebnisse zu erhalten, wurde die Einführung in die Programmierung parallel in zwei Grundkursen Informatik (11. Klasse) der gleichen Schule durchgeführt. Parallel zur Durchführung der in Kapitel 8 beschriebenen kreativitätsorientierten Unterrichtssequenz, die durch mich durchgeführt wurde, unterrichtete eine erfahrene Kollegin den Parallelkurs nach einem „traditionellen“<sup>112</sup> Ansatz, der das Lösen von gegebenen Problemen in den Vordergrund stellt. In diesem Kurs wurde das Werkzeug „Roboter-Karol“ verwendet sowie dazugehörige empfohlene Lernaufgaben eines Schulbuchs für den Informatikunterricht (Engelmann 2004).

Der untersuchte Kurs bestand aus 21 Schülern zwischen 15 und 17 Jahren, davon 38% Mädchen. Die Kontrollgruppe bestand aus 23 Schülern gleichen Alters mit 61% Mädchen. Die Vorerfahrungen beider Kurse in Informatik sind vergleichbar. So waren die Unterrichtsinhalte beider Kurse bis hierher themengleich. Keiner der Schüler hatte Vorerfahrungen im Programmieren. In beiden Kursen wurde in der vorhergehenden Unterrichtssequenz das Thema „Datenbanken“ gemäß dem Brandenburger Rahmenlehrplan (vgl. MBS Brandenburg 2006) behandelt.

#### 9.1.2 Datenerhebung

Als Forschungsinstrument wurde zur Beantwortung der ersten Frage ein Fragebogen verwendet. Zur Beantwortung der zweiten Frage wurden die Durchschnittsnoten der Schüler vor und nach der Unterrichtssequenz sowie die Ergebnisse eines Tests zum Ende der Unterrichtssequenz herangezogen.

Der Fragebogen<sup>113</sup> (siehe Anhang) war wie folgt strukturiert:

1. Skalenbasierte Fragen zu Aussagen über Informatikunterricht allgemein, z. B. „Informatikunterricht macht mir Spaß.“

<sup>111</sup> Teile dieses Abschnitts wurden bereits zum Teil sinngemäß und zum Teil wortwörtlich in (Romeike 2008a) veröffentlicht.

<sup>112</sup> Als „traditionellen“ Ansatz des Problemlösens bezeichne ich die Methodik, die von vielen Lehrern bei der Einführung in die Programmierung gewählt wird: durch Thematisierung der Konzepte und einer Sequenz konvergenter Problemlöseaufgaben mit steigender Komplexität und Schwierigkeit.

<sup>113</sup> Der Fragebogen wurde basierend auf Fragebögen der Fachbereiche Didaktik der Informatik der Universitäten Siegen und Paderborn entwickelt.

2. Fragen zur letzten Unterrichtssequenz: Empfundene Schwierigkeit, Stoffumfang und Angemessenheit.
3. Bewertung der Lehrtechniken, -methoden und Werkzeuge.
4. Skalenbasierte Fragen zu den Unterrichtsthemen, z. B. „Ich konnte im Unterricht etwas Neues entdecken.“
5. Fragen zum empfundenen eigenen Lernerfolg und zu den empfundenen Lernerfolgen des Kurses.

Der Fragebogen wurde von den Schülern vor und nach der Unterrichtssequenz zur Einführung in die Programmierung beantwortet. Die Unterrichtssequenz umfasste insgesamt 11 Unterrichtsstunden à 45 Minuten, welche sich auf vier Wochen verteilten.

Die abschließende Lernerfolgskontrolle (siehe Anhang) beinhaltete zwei Teile:

1. Theorie
  - Angabe der Definition und Eigenschaften von Algorithmen
  - Beschreibung der Konzepte der Programmierung und Angabe von Beispielen
2. Praxis
  - Erklären der Funktion und Optimierung zweier Programme auf Papier
  - Implementieren eines Programms zu einem gegebenen Problem
  - Implementieren eines Programms, welches eine selbst gestellte Aufgabe löst und dabei alle im Unterricht thematisierten Konzepte verwendet/demonstriert.

Aufgrund der unterschiedlichen Werkzeuge wurden die Aufgaben der Kontrollgruppe entsprechend angepasst.

### 9.1.3 Ergebnisse

Im Folgenden werden zuerst die Ergebnisse des Kurses im Pre-Post-Vergleich dargestellt. Anschließend werden die Ergebnisse mit den Ergebnissen der Kontrollgruppe verglichen.

#### Motivation, Interesse, Bild der Informatik

Es ist anzunehmen, dass sich das Bild, welches Schüler vom Unterrichtsfach Informatik haben, auch auf ihr Verständnis von der (Wissenschaft) Informatik auswirkt. Humbert (2003) untersuchte in seiner Dissertation u. a. die Vorstellungen, die Schüler von der Informatik haben und resümierte, dass Informatik als eine Art „Wissenschaft vom Computer und der Anwendung von Computern“ gesehen wurde. Die Möglichkeiten des Gestaltens und Anpassens von Softwaresystemen wurde kaum reflektiert. Ebenso wirkte sich ein Jahr Informatikunterricht nur unwesentlich auf eine Änderung dieser Sichtweise aus.

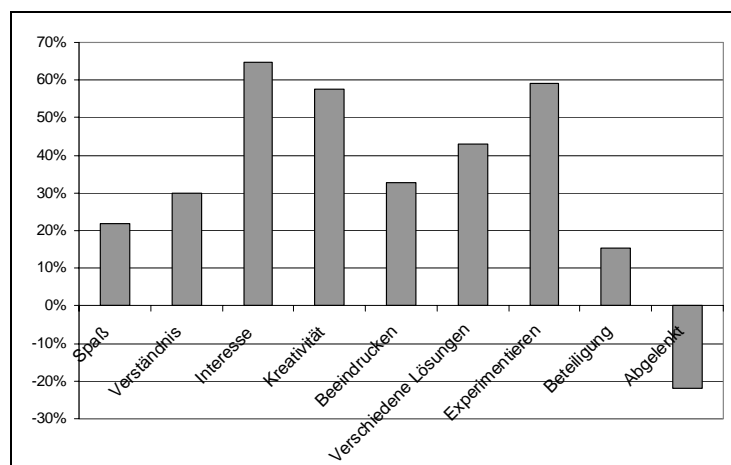


Abbildung 32: Zustimmungsänderung im Vergleich zur vorherigen Unterrichtssequenz.

Die kreativitätsorientierte Unterrichtssequenz änderte das Bild von der Informatik<sup>114</sup> in positiver Weise (vgl. Abbildung 32). Spaß und Interesse am Fach steigerten sich erheblich:

„Informatikunterricht macht mir Spaß“	Zustimmung vorher:	71%
	Zustimmung nachher:	93%
„Ich finde die Inhalte im Informatikunterricht interessant“	Zustimmung vorher:	29%
	Zustimmung nachher:	93%

Diese Faktoren haben einen großen Einfluss auf die Motivation der Schüler und können dazu dienen, das Interesse an der Informatik aufrecht zu erhalten: Programmieren war für die Schüler sehr motivierend – im Gegensatz zu vielen Erfahrungen im Informatikunterricht und in Einführungskursen an Universitäten, wo Programmierung oft ein Grund für Misserfolg ist.

Übereinstimmend mit dem Unterrichtskonzept gab es eine deutliche Änderung in der Bewertung von Kreativität. 93% der Schüler geben nun an, dass sie im Informatikunterricht kreativ sein können, im Gegensatz zu 36% vorher.

Informatik ist grundsätzlich ein Fach, in welchem für eine Aufgabe verschiedene Lösungen möglich sind und wo Verständnis für schwierige Beziehungen auch durch Experimentieren erreicht wird. Experimentieren spielt in der Praxis eine immer wichtigere Rolle, z. B. bei der Analyse des Verhaltens von Software (vgl. Reed 2002). In der Schulinformatik werden diese Aspekte allerdings oft nicht deutlich, insbesondere wenn Lehrer „effektive“ Wege des Unterrichts wählen, wie z. B. Frontalunterricht mit konvergenten Problemlöseaufgaben, um den Unterrichtsstoff innerhalb kürzester Zeit „zu schaffen“. Vor der kreativen Unterrichtssequenz ist sich die Mehrheit der Schüler dieser Aspekte nicht bewusst. Nach der kreativitätsorientierten Unterrichtssequenz stimmen die meisten Schüler den folgenden Aussagen zu

„Im Informatikunterricht sind für eine Aufgabe verschiedene Lösungswege und Lösungen möglich“	Zustimmung vorher:	43%
	Zustimmung nachher:	86%
„Im Informatikunterricht wird viel experimentiert“	Zustimmung vorher:	14%
	Zustimmung nachher:	73%

Diese Steigerung ist auch vor dem Hintergrund interessant, dass die Schüler in der vorherigen Unterrichtssequenz Datenbanken untersuchten, entwarfen und damit experimentierten. Offensichtlich trifft das eigenständige Entwerfen und Ändern von Software durch kreatives Programmieren besser das Verständnis der Schüler von „Experimentieren“ als das Untersuchen der Eigenschaften eines gegebenen „festen“ Systems, wie der Datenbanken, mit konvergenten Problemlöseaufgaben.

Das Erstellen von präsentierbaren Produkten (Programmen) kann sich auch auf das Informatikbild von Freunden und Familie auswirken. Nach anfangs fast keinem Schüler (7%) stimmen später immerhin 40% der Schüler der Aussage zu, dass sie mit Ergebnissen aus dem Informatikunterricht zu Hause oder bei Freunden beeindrucken können.

### Verständnis und Leistung

Der eigene Lernerfolg wird von den Schülern als hoch eingeschätzt. Diese Frage wird vom gesamten Kurs nahezu übereinstimmend beantwortet, welches im Gegensatz zum wahrgenommenen Lernerfolg der vorherigen Unterrichtssequenz steht. Hier streuen sich die Antworten merklich und verdeutlichen Probleme in etwa der Hälfte der Antworten. Diese Sichtweise wird auch in der Einschätzung des Lernerfolges des gesamten Kurses durch die einzelnen Schüler deutlich. 87% geben für die Unterrichtssequenz zur Programmierung an, dass alle oder die meisten Schüler des Kurses die Inhalte gut verstanden haben. Für die Unterrichtssequenz vorher gibt die Mehrheit der Schüler dagegen an, dass ihre Mitschüler beim Lernen Probleme hatten. Interessant hierbei ist, dass die Einschätzung des eigenen Lernerfolgs beim Thema Datenbanken im Allgemeinen besser war als der eingeschätzte Lernerfolg des restlichen Kurses. Auch wenn mehr als die Hälfte der

<sup>114</sup> „Bild von der Informatik“ bezieht sich hier auf Charakteristika des Unterrichtsfachs wie *Experimentieren*, *verschiedene Lösungen möglich* und *kreativ* sowie subjektive Zuschreibungen wie *Spaß*, *interessant* oder *verständlich*.

Schüler die Lerninhalte verstanden, lernte der Kurs in einem Klima von Problemen und Nichtverstehen. Bei der Programmierung dagegen wurde durch die vielen Möglichkeiten, Ergebnisse und Schülerleistungen vorzustellen, ein positiveres Unterrichtsklima erreicht. Dieses wirkte sich motivierend auf die Ausdauer der Schüler und ihr Streben aus, schwierige Sachverhalte zu verstehen, wenn Probleme auftraten.

Die tatsächlichen Lernerfolge wurden durch eine schriftliche Lernerfolgskontrolle zum Abschluss der Unterrichtssequenz festgestellt. Diese wurde von 94% der teilnehmenden Schüler erfolgreich absolviert. Der Kursdurchschnitt des Tests ist 0,2 Zensureinheiten besser als die Kursdurchschnittsnote in der ersten Hälfte des Schulhalbjahres und im Durchschnitt eine ganze Zensur besser gegenüber dem Abschlusstest der vorherigen Unterrichtssequenz. Die Leistungen der Schüler lassen sich in zwei Gruppen aufteilen: 69% erreichten die Benotung „gut“ und „sehr gut“, 25% erreichten „befriedigend“ und „ausreichend“. Eine Betrachtung der Ergebnisse getrennt nach Geschlechtern zeigt, dass alle Mädchen „gut“ und besser abschnitten, während die Ergebnisse der Jungs über die ganze Notenskala verteilt liegt. Vor dem Hintergrund, dass viele Informatik- und Programmierkurse Probleme mit den Leistungen von Mädchen haben, überrascht dieses Ergebnis positiv.

### Unterrichtsmethodik

Schüler sind es häufig nicht gewohnt, eigenständig und mit eigenen Zielen zu arbeiten. Auch wenn seit langem andere Unterrichtsmethoden aus Sicht der Pädagogik angestrebt werden, überwiegt in der Praxis immer noch Frontalunterricht als am häufigsten angewandte Unterrichtsmethodik. Das Anwenden einer neuen Unterrichtstechnik kann sich vor diesem Hintergrund als problematisch für alle Beteiligten herausstellen; die Schüler wissen nicht genau, was von ihnen erwartet wird und können keinem vertrauten Schema folgen. Entsprechend kann es aufschlussreich sein, die Wahrnehmungen und Einstellungen der Schüler für die neue Unterrichtsmethodik zu untersuchen. Die Antworten der Schüler zur Unterrichtsmethodik, den Aufgaben und den Unterrichtsstunden werden in Abbildung 33 dargestellt.

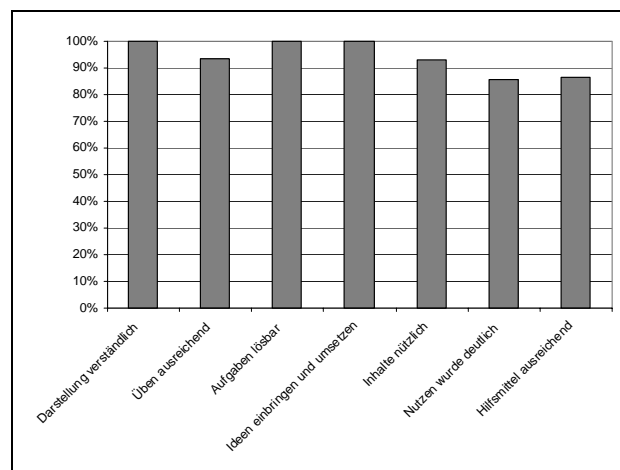


Abbildung 33: Einschätzung der Unterrichtsmethodik durch die Schüler.

Alle Schüler bezeichnen die Darstellung der Unterrichtsinhalte als verständlich. Dieses Ergebnis wurde angestrebt, ist allerdings aus zwei Gründen erstaunlich: Zum einen wurden durch die Lehrperson nur wenig Anstrengungen unternommen, die Programmierkonzepte einzuführen und zu erklären. Mehr oder weniger alle Lernkompetenzen wurden durch aktive Auseinandersetzung mit den Programmierkonzepten erreicht. Inhalte wurden kurz präsentiert oder gemeinsam zusammengetragen und sofort angewandt. Zum anderen ist Programmieren als schwieriges Unterrichtsthema im Informatikunterricht bekannt. Die Schüler bezeichnen hier den Schwierigkeitsgrad als angemessen und bewerten auch ihren Lernerfolg als angemessen oder hoch. Auf die Frage nach

der Schwierigkeit des Themas antwortet die Hälfte der Schüler mit „mal leicht, mal schwer“, die andere Hälfte mit „eher leicht“.

Die Übungszeit wurde als angemessen empfunden, auch wenn die Umstände der Unterrichtssequenz für einen gewissen Zeitdruck sorgten. Die Schüler geben an, dass sie eigene Ideen einbringen konnten, und bezeichnen die gestellten Aufgaben als lösbar.

Die Rolle des Lehrers wird in Antworten auf die Frage, wodurch die Schüler am meisten lernten, reflektiert: Durch Projektarbeit (60%) und Beschäftigung mit den Aufgaben (60%) im Gegensatz zu Erklärungen des Lehrers (13%). Es ist überraschend, dass Programmieren, zumindest auf dieser einführenden Stufe, so intuitiv gelernt werden kann.

### Fragen zum Unterrichtsthema

Die Antworten zur Relevanz des Unterrichtsthemas sind etwas ambivalent vor dem Hintergrund, dass die Schüler in einer technologiereichen Umgebung aufwachsen. Jeder Schüler des Kurses hat einen Computer zu Hause. Dennoch meinen weniger als ein Drittel der Schüler, dass das Unterrichtsthema etwas mit dem täglichen Leben zu tun hat und nur 43% der Schüler glauben, dass sie das Gelernte in Zukunft gebrauchen können. Diese Zustimmungen sind sogar weniger als in der Unterrichtssequenz davor. Im Unterricht wurden die Relevanz von Programmierkonzepten und die Bedeutung im täglichen Leben nicht explizit durch die Lehrperson betont. Bezüglich der Relevanz im täglichen Leben als wichtige Motivationsquelle ließe sich vermuten, dass die Schüler kaum Motivation für den Unterricht entwickelten. Erstaunlicherweise geben 80% der Schüler an, dass einige Unterrichtsinhalte für sie besonders interessant waren, 87% geben an, dass sie etwas gelernt haben und 73% finden, dass ihnen das Thema Spaß gemacht hat. Auch wenn die Schüler nicht durch die lebensweltliche Relevanz motiviert wurden, waren die Aufgaben und das kreative Tun offenbar motivierend genug, dass die Schüler lernten und dabei sogar Spaß hatten.

### Vergleich mit der Kontrollgruppe: Betrachtung der Fragebögen

Vor der Einführung in die Programmierung wurden in beiden Kursen die gleichen Informatik-inhalte behandelt. Auch die Schulnoten der beiden Kurse waren vergleichbar.

Der Vergleich der Antworten der beiden Kurse bezüglich der Einschätzung der Informatik vor Einführung in die Programmierung zeigt überwiegend Gemeinsamkeiten. Kurs B<sup>115</sup> stimmte etwas mehr zu bei der Frage nach „Spaß“ und stimmte deutlich mehr zu bei der Frage nach dem Verständnis der Inhalte. Die weiteren Antworten sind vergleichbar.

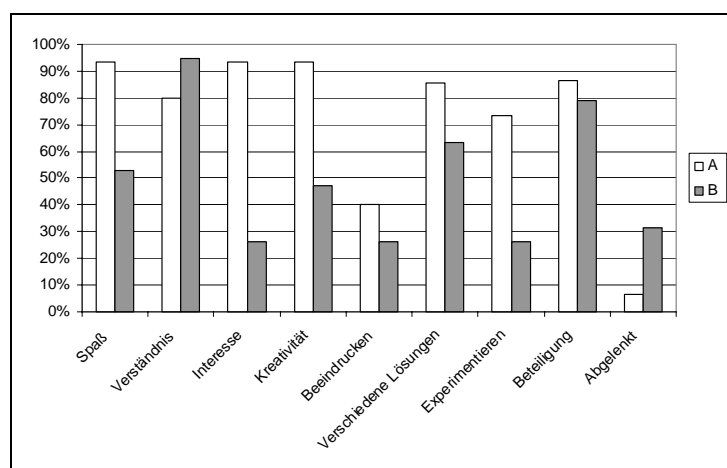


Abbildung 34: Zustimmung im Vergleich zur Kontrollgruppe.

<sup>115</sup> Im Folgenden wird der Kurs, der eine kreative Einführung in die Programmierung erlebte, mit A bezeichnet, der Kurs, in welchem ein Problemlöseansatz verfolgt wurde wird mit B bezeichnet.

Dies ändert sich deutlich, wenn man die Antworten nach der Einführung in die Programmierung vergleicht. Spaß nimmt in A um 22% zu, während er in B um 32% zurückgeht. Nur gut die Hälfte der Schüler der Kontrollgruppe empfindet Informatik nach der Unterrichtssequenz zur Programmierung als ein Fach, das Spaß macht. Während Programmieren einen deutlichen Einfluss auf das Interesse der Schüler im Kurs A ausübt, bleibt das Fach für 75% der Schüler in B uninteressant.

In Kurs B steigt ebenfalls die Zustimmung auf die Frage nach Kreativität (+18%). Dies ist ein interessantes Ergebnis, welches zeigt, dass auch im problemlösenden Unterrichtsansatz die Kreativität der Schüler angesprochen wird und dies auch den Schülern deutlich wird. Nicht für alle Schüler in B wird deutlich, dass für ein Problem verschiedene Lösungen möglich sind. Die Zustimmung zu dieser Aussage steigt in B auf 63%, während in A 86% der Schüler zustimmen. Die Zustimmungen im Vergleich mit der Kontrollgruppe werden in Abbildung 34 dargestellt.

### **Vergleich der Leistungen**

Die Unterrichtssequenz wurde in beiden Kursen mit den gleichen Lernzielen begonnen. Leider konnten in Kurs B nicht alle Lernziele erreicht werden. Variablen sind im dort verwendeten Werkzeug nicht implementiert. Ebenso wurden die Eigenschaften von Algorithmen vom Lehrer dieses Kurses aus Zeitmangel nicht thematisiert.

Beide Kurse machen im Fragebogen Angaben zu ihrem Lernerfolg. Während in A alle Schüler den Lernerfolg als angemessen oder hoch bezeichnen, waren dies in B nur zwei Drittel der Schüler. Die Ergebnisse des abschließenden Tests waren allerdings vergleichbar. Die Schwierigkeit und Komplexität der Unterrichtssequenz bezeichnen beide Kurse gleichermaßen als angemessen oder hoch.

Im Vergleich der Zensuren mit der vorausgehenden Unterrichtssequenz zeigt sich, dass Kurs A sich deutlich verbesserte, während Kurs B im Durchschnitt bei gleichen Leistungen blieb. Unterteilt man die Zensuren allerdings nach Geschlechtern wird deutlich, dass in Kurs B sich die Jungen verbesserten, während sich die Mädchen verschlechterten. Im Gegensatz dazu stehen die Leistungen des Kurses A. Hier verbesserten sich die Mädchen deutlich stärker als die Jungen. Vor der Einführung in die Programmierung war das Leistungsniveau beider Geschlechter vergleichbar.

#### **9.1.4 Kritische Betrachtung**

Die Studie unterlag aus organisatorischen Gründen zwei Schwachpunkten. Zum einen wurden die Kurse von unterschiedlichen Lehrern unterrichtet. Der Unterrichtsstil, die Lehrerpersönlichkeit und die Art und Weise, wie die Schüler mit der Lehrperson zurecht kommen, kann einen deutlichen Einfluss auf den Lernerfolg und die Motivation der Schüler haben. Andererseits, wie aus den Fragebögen deutlich wurde, welche die Schüler vor der untersuchten Unterrichtssequenz ausfüllten, waren Motivation und Leistungen in beiden Kursen vergleichbar. Da die Aufgaben und Übungen, die in der Kontrollgruppe verwendet wurden, einem etablierten Schulbuch entstammen, scheinen dieses Aufgaben für einen Kurs, der mit Problemlöseaufgaben in die Programmierung einführt, typisch zu sein. Zum anderen wurden in den beiden Gruppen nicht nur unterschiedliche Methoden, sondern auch unterschiedliche Softwarewerkzeuge verwendet. Dieser Fakt könnte durchaus einen maßgeblichen Einfluss auf die Motivation und die empfundene Kreativität haben.

Eine grundlegende Aussage dieser Untersuchung bleibt auch vor diesem Hintergrund gleich: Eine kreative Einführung in die Programmierung ist möglich und kann erfolgreich durchgeführt werden.

## **9.2 Untersuchung zum Verständnis von Kreativität in der Informatik<sup>116</sup>**

Die Ergebnisse der Fragebogenuntersuchung zeigen, dass sich das Bild von der Informatik durch die kreative Unterrichtssequenz verbessert hat. Fast alle Schüler geben nun an, dass sie im Infor-

---

<sup>116</sup> Teile dieses Abschnitts wurden bereits zum Teil sinngemäß und zum Teil wortwörtlich in (Romeike 2008c) veröffentlicht.

matikunterricht kreativ sein können. Ungeklärt bleibt bisher die Frage, was sich die Schüler konkret unter Kreativität in der Informatik vorstellen und inwieweit sich ein kreatives Bild von der Informatik als attraktiv darstellt. Der Unterrichtsversuch hat bei den Schülern viel Kreativität ermöglicht und gefordert, bis dahin aber nicht explizit Kreativität thematisiert, was in einer qualitativen Untersuchung im Anschluss an die Unterrichtssequenz mit dem Ziel, Kreativität besser in den Lernprozess einzubinden, getan wurde.

### 9.2.1 Datenerhebung und Analyse

Die Frage nach dem Verständnis von Kreativität in der Informatik wird mittels eines qualitativen Forschungsansatzes adressiert. Die Untersuchung mit Hilfe detaillierter Fragebögen in der Evaluation des Unterrichtsversuchs ermöglicht bereits einen abstrakten Überblick über den Erfolg der kreativen Unterrichtssequenz sowie über die Änderungen der Vorstellungen der Schüler vom Fach Informatik. Demnach denken fast alle Schüler nach der kreativen Programmierung, dass sie im Informatikunterricht kreativ sein können. Ein qualitativer Ansatz ermöglicht es, über vorgegebene Fragen hinaus, die Vorstellungen der Schüler zu erfassen. Zwei verschiedene Arten von Daten wurden erhoben:

1. Bearbeitungen einer Brainstorming-Aufgabe mit den Stichwörtern *Kreativität* und *Informatik* und der Frage: „Was fällt dir ein zum Thema Kreativität und Informatik?“ (vgl. Abbildung 35)

Die Schüler bearbeiteten diese Aufgabe etwa 10 Minuten.

2. Ausführungen zu einer Erläuterungsaufgabe:

„Fasse nun das, was dir zu Kreativität im Zusammenhang mit Informatik einfällt in Worte. Dazu kannst du deine Ergebnisse des Brainstormings verwenden und erläutern oder andere Aussagen beschreiben. Verwende ggf. Beispiele, um deine Erklärungen zu illustrieren. Fange damit an zu beschreiben, was du unter Kreativität verstehst.“

Die Schüler arbeiteten etwa 20 Minuten an dieser Aufgabe.

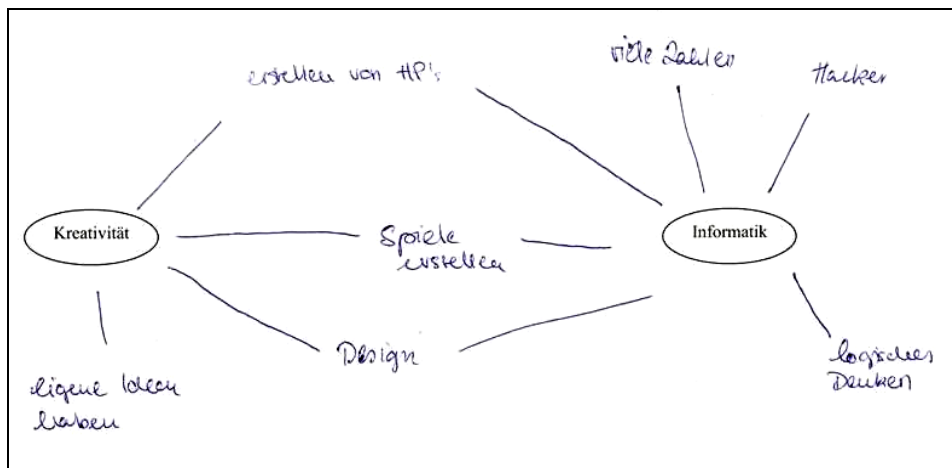


Abbildung 35: Brainstorming zu Kreativität und Informatik (Schülerbeispiel).

In der Auswertung wurden zuerst die Brainstorming-Skizzen analysiert und kategorisiert. Als nächstes wurden die verschiedenen Ideen des Brainstormings den gefundenen Kategorien zugeordnet. Die Erläuterungen der Schüler wurden analysiert und bezüglich der Kernaussagen systematisiert.

## 9.2.2 Ergebnisse

### Analyse der Brainstorming-Skizzen

Die Ergebnisse des Brainstormings können drei Kategorien zugeschrieben werden: Stichwörter, die sich auf Kreativität beziehen, Stichwörter, die sich auf Informatik beziehen und Stichwörter, die sich zwischen diesen beiden Bereichen als verbindende Elemente befinden. Während die meisten Schüler die verbindenden Elemente visuell darstellen, betrachten zwei Schüler die beiden Bereiche separat. In den Erklärungen der zweiten Aufgabe werden allerdings auch von diesen Schülern verbindende Elemente beschrieben. Die häufigsten genannten Antworten sind in Abbildung 36 dargestellt.

Kreativität		Verbindende Elemente		Informatik	
Kunst	9	Programmieren	9	Computer	10
Eigene Ideen haben	5	Scratch	6	Daten	3
Ideen umsetzen	5	Homepageerstellung	4	Logik	3
Erschaffen, Schöpfen	4	Photoshop/Fotobearbeitung	3	Programmieren	3
Basteln	3	Musik bearbeiten/erstellen	3	Schulunterricht	2
Malen/Zeichnen	3	kreatives Schreiben (Word)	3	Internet/-anwendungen	2
Freie Möglichkeiten	3	Computerspiele	3	Informationen	1
einfallsreich	2	Ideen	3	Hacker	1
menschl. Fähigkeit	2	Internet	2	Algorithmus	1

Abbildung 36: Häufige Schülerantworten des Brainstormings zu Kreativität und Informatik.

Die Analyse der Brainstormingskizzen erlaubt einen generellen Einblick in die Erfahrungen und Vorstellungen der Schüler. Die am häufigsten genannten Stichwörter für Kreativität und auch für Informatik entsprechen gängigen Stereotypen: Die Hälfte der Schüler nennt „Kunst“ als repräsentativ für Kreativität und ebenso nennt etwa die Hälfte der Schüler „Computer“ als typisch für Informatik. Als verbindendes Element zwischen Kreativität und Informatik wird von der Hälfte der Schüler „Programmieren“ genannt. Die zweitpopulärste Antwort in dieser Kategorie ist die Programmiersprache Scratch. Des Weiteren wurden verschiedene Möglichkeiten des praktischen Arbeitens mit dem Computer als verbindendes Element genannt. Damit spiegeln die genannten verbindenden Elemente zwei Dimensionen des Faktorenmodells wider: Kreativität aus der Informatik-Fachsicht sowie IKT als kreativitätsfördernde Werkzeuge der Informatik.

### Analyse der Erläuterungen

Alle Schüler beschreiben Elemente und Aktivitäten, die Kreativität mit Informatik verbinden. Während die Analyse der Brainstorming-Stichwörter allerdings ein recht homogenes Bild mit verschiedenen Antworten innerhalb der drei Kategorien zeigt, offenbart die Analyse der einzelnen Begründungen und Meinungen verschiedene Sichtweisen, wie Kreativität im Bezug auf Informatik von Schülern aufgefasst werden kann. Die Aussagen der Schüler bzgl. der Fragestellung, welche Zusammenhänge es zwischen den beiden Disziplinen gibt, können fünf verschiedenen Kategorien zugeordnet werden:

#### *Informatik ist von Natur aus kreativ*

Viele Schüler erklären, dass ihrer Meinung nach Kreativität eine Grundvoraussetzung in der Informatik ist, z. B. weil Programmieren Kreativität erfordert:

*Mir sind im Zusammenhang der beiden Themen sofort Programmiersoftware wie Scratch oder Delphi eingefallen. Man kann dabei seine eigenen Ideen in die Projektarbeit einbeziehen. [FB3T6]<sup>117</sup>*

<sup>117</sup> Der Code identifiziert die Fragebögen und Schüler (FB3 steht für den Fragebogen, TX für den Teilnehmer).

In einer ähnlichen Sichtweise beschreiben die Schüler Informatik als eine Art angewandter Kreativität, welche sich scheinbar von der Produktorientierung der Informatik ableitet. Hierbei wird mitunter auch der persönliche Standpunkt hervorgehoben, indem Kreativität als Möglichkeit der Umsetzung eigener Ideen empfunden wird. Die Schüler beziehen sich in ihren Erklärungen auf die Sichtweise von P-Kreativität.

### ***Informatik unterstützt Kreativität***

*Informatik ist für mich insofern kreativitätsfördernd, als dass sie einem nicht nur eine, sondern mehrere Türen öffnet, in verschiedenste Bereiche. Jeder kann etwas damit anfangen. [FB3T12]*

Die Schüler sind sich der kreativitätsunterstützenden Eigenschaften von Informations- und Kommunikationstechnologien (IKT) bewusst. Diese Sichtweise dehnen sie auf die Strategien und Konzepte der Informatik aus, die sie im Informatikunterricht kennen gelernt haben und welche es ihnen erleichtern, IKT effizient und kreativ einzusetzen. Diese Sichtweise beinhaltet auch die Anwendung verschiedener Werkzeuge, durch die sie sich kreativ ausdrücken können:

*Außerdem ist auch das E-Mail schreiben eine Sache bei der oftmals das kreative Vermögen eine wichtige Rolle spielt, wenn man einer Person beispielsweise eine Liebeserklärung in einer E-Mail verschicken will und diese etwas Besonderes darstellen soll, ist Kreativität sehr gefragt. [FB3T5]*

### ***Kunst-Sichtweise von Informatik***

In dieser Sichtweise beschreiben die Schüler Tätigkeiten innerhalb der Informatik, die mit Kunst vergleichbar sind, z. B. bei der visuellen Informationsdarstellung:

*Um Probleme und Beziehungen besser darzustellen (z. B. Bilder, Diagramme) braucht man in der Informatik Kreativität. [FB3T18]*

Diese Sichtweise findet auch in verschiedenen verbindenden Themenbereichen ihre Anwendung, wie z. B. Media-Design oder Computergraphik, wo beides, Informatikwissen und künstlerischer Ausdruck, kreativ kombiniert werden.

### ***Holistische Sichtweise***

In dieser Sichtweise wird der Einfluss der Informatik und von Produkten der Informatik auf die Gesellschaft betrachtet. Dabei spielt die eigene Beschäftigung mit Informatik eine untergeordnete Rolle. Stattdessen wird die gesellschaftliche Gesamtentwicklung analysiert:

*In der heutigen Industrie ist die Informatik nicht mehr wegzunehmen, alles wird von Nullen und Einsen gelenkt, ob Autos, Flugzeuge, Lebensmittelherstellung, Buchhaltung allgemein, Ingenieurwesen, Architektur oder sonstiges, überall findet sie sich wieder und erleichtert vieles. [FB3T12]*

*Informatik ist heutzutage überall. Sie ist so einflussreich, das bezeichne ich als kreativ. [FB3T7]*

*Kreativität in der Informatik ist verantwortlich für den technischen Fortschritt. [FB3T20]*

Auch in Hinsicht auf kreativen Missbrauch von Technik (z. B. durch Hacking) und die kritische Betrachtung des technischen Fortschritts sind dies wichtige Aspekte, die von Informatiklehrern im Unterricht berücksichtigt werden können, um die Schüler zum kritischen Denken zu ermutigen.

### ***Herausstellung des Unterschieds***

Für einige Schüler war es wichtig, den Unterschied zwischen Kreativität und Informatik herauszustellen. Diese Schüler sehen Kreativität als überragende menschliche Eigenschaft an, die sie nicht zu eng an die Informatik gebunden sehen möchten:

*Programmieren fördert die Kreativität des Programmierers und Kreativität ist notwendig beim Programmieren. Informatik braucht Kreativität, aber Kreativität braucht nicht Informatik! [FB3T13]*

Besonders interessant sind jene Schülerantworten, in welchen die Auseinandersetzung mit Stereotypen deutlich wird:

*Grundlegend gehorchen Informatik und Kreativität unterschiedlichen Gesetzen. Die Informatik braucht Logik und mathematische Strukturen wohingegen die Kreativität meist aus solchen ausbricht. Die Kreativität sprengt Gesetze und Algorithmen während die Informatik durch diese beherrscht wird. Dennoch funktioniert die Informatik nicht ohne Kreativität. Um neue Programme und Quelltexte zu schreiben braucht man einen kreativen Geist. [FB3T19]*

### 9.2.3 Diskussion

Die Schülerantworten zeigen verschiedene Sichtweisen von Kreativität und Informatik. Ihre eigene Beschäftigung mit Informatik wird dabei in der Regel als kreativ bezeichnet – innerhalb und außerhalb des Unterrichts. Vor dem Hintergrund, dass die kreative Beschäftigung mit Informatik als motivierend empfunden wird, kann diese Eigenschaft als Möglichkeit gesehen werden, Motivation und Interesse für die Informatik zu steigern.

Zu berücksichtigen bleibt, dass Kreativität im Informatikunterricht für die Schüler auch durchaus unerwartet und verwirrend sein kann, wenn sie nicht daran gewöhnt sind. So klagte eine Schülerin während der Unterrichtssequenz darüber, dass dies ja „so kreativ“ sei und damit doch nicht in den Informatikunterricht gehöre. In der Umfrage nach der Unterrichtssequenz erläuterte sie ihren Standpunkt. Ihr gefiel nicht, dass

1. der Lehrer Anweisungen so ungenau gibt, dass man Ideen haben muss (z. B. welches Spiel entwickle ich?),
2. man darüber nachdenken muss, wie man diese Ideen umsetzt,
3. jeder andere Ergebnisse hat.

Mit diesen Punkten beschrieb die Schülerin einige Kernpunkte, auf denen die Unterrichtssequenz aufbaut. Ihre Unsicherheit basierte augenscheinlich auf der für sie neuen Situation, in welcher ihr nicht genau gesagt wurde, was sie zu tun hat sowie auf mangelnden Ideen. Während die meisten anderen Schüler kein Problem hatten, Ideen zu entwickeln, konnte sich diese Schülerin nur schwer in das selbständige Arbeiten einfinden. Vor der nächsten Unterrichtssequenz zum Thema Computergrafik wurden verschiedene Kreativitätstechniken besprochen und geübt mit dem Resultat, dass die Bearbeitung offener Aufgaben bei allen Schülern gut gelang.

## 9.3 Fazit

Ziele der Unterrichtssequenz waren zum einen, die Lernziele zu erreichen, und zum anderen, die Schüler mittels kreativer Aufgaben für die Programmierung zu motivieren. Wie verdeutlicht, konnten beide Ziele erreicht werden. Weiterhin ist festzustellen, dass sich das Bild vom Informatikunterricht in eine positive Richtung verändert hat. Informatikunterricht wird häufig als ein Unterrichtsfach aufgefasst, in welchem gelehrt wird, Computer, das Internet und Software wie Word und Excel zu bedienen. Die Schüler in diesem Unterrichtsversuch kamen der kreativen Seite der Informatik näher: Informatik hat viel zu tun mit dem Entwerfen und Ändern von Informatiksystemen, Experimentieren und dem Finden einer guten Lösung bei vielen möglichen Lösungen. Diese Sichtweise wurde nun auch in den Schülerantworten nach der Unterrichtssequenz deutlich. Außerdem konnte festgestellt werden, dass sich Kreativität im Informatikunterricht für die Schüler als attraktiv darstellt. Dabei bleibt allerdings zu beachten, dass die Schüler hiermit nicht überfordert und allein gelassen werden dürfen.

Die aufgestellten Hypothesen können vor dem Erfahrungshintergrund dieses Unterrichtsbeispiels bestätigt werden: Motivation und Interesse der Schüler wurden durch die kreative Programmierung deutlich gestärkt. Programmieren wurde den Schülern als kreativer Prozess deutlich und ermög-

lichte ihnen die Gestaltung vieler kreativer Projekte. Der Einsatz der kreativitätsunterstützenden Programmiersprache Scratch ermöglichte den Schülern eine kreative Auseinandersetzung mit den Aufgaben.

Neben der Berücksichtigung der im Faktorenmodell kreativen Informatikunterrichts dargestellten Aspekte wirkte sich die Anwendung des *Challenge-Cycles* in der Gestaltung der einzelnen Unterrichtsphasen positiv auf die Entwicklung von Kreativität aus. So wurde der Challenge-Cycle mit jeder Einführung neuer Konzepte neu betreten und jeweils durch die Vorstellung verschiedener Ergebnisse der (Mini-)Projekte beendet. Durch die passende Gestaltung der Lernsituation und Aufbereitung der jeweiligen Grundlagen im Zusammenhang mit der Darstellung der Programmierkonzepte in der eingesetzten Programmiersprache wurde ein überwiegend selbständiges Lernen und praktisch kreatives Erarbeiten der Unterrichtsinhalte erreicht. Die Möglichkeit der Schüler, sich eigene Challenges zu bestimmen, führte darüber hinaus zu einer großen Bandbreite an verschiedenen Projekten.

## 10 Zusammenfassung, Diskussion und Ausblick

### 10.1 Zusammenfassung

Den Ausgangspunkt dieser Arbeit bildeten die Feststellungen

1. Kreativität soll in der Schulinformatik nach einhelliger Meinung vieler Fachdidaktiker gefördert werden.
2. Die Förderung von Kreativität in der Informatik, insbesondere in der Modellierung, wird für möglich und angebracht gehalten.
3. Bisher nimmt Kreativität augenscheinlich nur einen unzureichenden Stellenwert im Informatikunterricht ein.

Darüber hinaus wurde – eigene Erfahrungen zugrunde legend – vermutet, dass Informatikunterricht nicht nur Kreativität fördern, sondern das Potential, welches sich aus Kreativität für den Lernprozess ergibt, hinsichtlich eines besseren Lernerfolgs nutzen kann.

Als Ziel dieser Arbeit wurde die Entwicklung einer fundierten theoretischen Grundlage für die Anwendung und Förderung von Kreativität im Informatikunterricht angegeben.

Dazu gehörten

- die Herausarbeitung eines für den Informatikunterricht relevanten Kreativitätsverständnisses
- die Feststellung der Charakteristika von Kreativität im Informatikunterricht
- die Bestimmung der Chancen, die sich aus Kreativität im Informatikunterricht ergeben
- die Entwicklung eines theoretischen Konzepts für die Gestaltung kreativen Informatikunterrichts
- die Untersuchung der Auswirkungen kreativen Informatikunterrichts

Zur Realisierung dieser Ziele wurde, ausgehend von Betrachtungen zur Kreativität und Informatik, ein Faktorenmodell kreativen Lernens im Informatikunterricht entwickelt, überprüft und erprobt.

Ergebnisse der Kreativitätsforschung implizieren verschiedene Faktizitäten für die Gestaltung eines kreativen Informatikunterrichts. Aufgrund des zentralen Stellenwerts des systematischen Entwickelns von Lösungen in der Informatik wird eine strukturorientierte Sichtweise sowie die auf eine Person bezogene Ausprägung von Kreativität als adäquat betrachtet.

Motivation und Kreativität stehen in einem wechselseitigen Verhältnis. Intrinsische Motivation ist eine Grundvoraussetzung für kreatives Tun, gleichwohl kann kreatives Tun auch intrinsische Motivation hervorrufen. In verschiedenen Lernzieltaxonomien wird Kreativität als höchstes kognitives Lernziel beschrieben. Als Chancen, die sich durch Kreativität im Unterricht ergeben, werden gesteigerte Schülermotivation, höhere Aufmerksamkeit, Konzentration, Neugier und ein daraus resultierender verbesserter Lernerfolg angegeben.

Zur Betrachtung von Kreativität in der Informatik ist die Differenzierung heterogener Sichtweisen erforderlich. Die Betrachtung aus der *Fach-Perspektive* verdeutlicht die subjektive kreative Wahrnehmung sowie kreative Fachcharakteristika der Disziplin. Die *Mensch-Perspektive* verdeutlicht einen positiven Einfluss kreativer informatischer Prozesse auf die Motivation. Aus der *Kunst-Perspektive* ergeben sich zahlreiche thematische Anknüpfungspunkte, Informatikinhalte zu transportieren. Die Entwicklung und Anwendung von Informatiksystemen als Werkzeuge für die Förderung kreativen Arbeitens in und außerhalb der Informatik wird in der *Technologie-Perspektive* beschrieben. Die Perspektive der artifiziellen Kreativität greift Bestrebungen in der Informatik auf, Systeme zu konstruieren, die Kreativität nachzubilden versuchen. Die Sichtweisen unterstreichen den zentralen Stellenwert von Kreativität in der Informatik. Ziele kreativen Informatikunterrichts sind die Anwendung und ein Verständnis von Kreativität bezogen auf die Fach-

Perspektive: sich aus informatischen Prozessen ergebende Leistungen, die zu persönlich neuen und verwendbaren Ideen, Produkten, Lösungen oder Erkenntnissen führen.

Informatik wird von Schülern bisher nur partiell als kreatives Unterrichtsfach wahrgenommen. Es konnte auf Projekte verwiesen werden, in denen Kreativität erfolgreich zur Steigerung von Motivation und Interesse instrumentalisiert wurde. Darüber hinaus stellt sich kreatives Lernen als adäquat für das Lernen informatischer Konzepte mit IKT unter Berücksichtigung der Bedürfnisse und Interessen der Lernenden dar. Vor dem Hintergrund eines sich ändernden Verständnisses von Computern sollte in einem kreativen Informatikunterricht auf eine eng ausgelegte Problemorientierung zugunsten von *Herausforderungen* verzichtet werden. Hierzu wurde der Challenge-Cycle als Framework für kreatives Lernen im Informatikunterricht vorgeschlagen.

Basierend auf den Erkenntnissen der zugrunde liegenden Wissenschaftsbereiche wurde ein Faktorenmodell kreativen Lernens für den Informatikunterricht entwickelt. Im Faktorenmodell stellen kreative informatische Prozesse das zentrale Element in einem kreativen Informatikunterricht dar; *Schüler, Fach* und *Informations- und Kommunikationstechnologien* als Umgebungsmerkmale stellen vernetzte Faktoren dar, die Kreativität im Informatikunterricht fördern.

In der Überprüfung des Faktorenmodells konnten in Computernutzungsbiographien von Studienanfängern der Informatik die drei Dimensionen des Faktorenmodells nachgewiesen werden. In einem Interview mit einem herausragend kreativen Schülers wurden Faktoren und Zusammenhänge des Faktorenmodells aus subjektiver Schülersicht verdeutlicht. In einer Studie mit zukünftigen Informatiklehrern wurde festgestellt, dass diese Kreativität wertschätzen, allerdings unkonkrete Vorstellungen von der Umsetzung im Informatikunterricht haben.

Das Modell wurde durch Kriterien kreativen Informatikunterrichts konkretisiert. Die Anwendung dieser bei der Evaluation publizierter Unterrichtsvorschläge bestätigt den bisher geringen Stellenwert von Kreativität im Informatikunterricht.

In einem Unterrichtsbeispiel zur Einführung in die Programmierung wurde eine Umsetzung kreativen Informatikunterrichts exemplarisch verdeutlicht. Die Anwendung von Kreativität im Unterrichtsbeispiel erfüllte die Erwartungen: Die Schüler hatten Spaß am Unterricht, die Zielkompetenzen wurden erreicht und die Schüler verbesserten ihr Bild von der Informatik.

Die aus dem Faktorenmodell abgeleiteten Hypothesen

1. Im Informatikunterricht stimuliert kreatives Arbeiten Motivation und Interesse der Schüler.
2. Im Informatikunterricht wird Kreativität aufgrund des kreativen Charakters des Fachs Informatik gefördert.
3. IKT fördern Kreativität im Informatikunterricht.

konnten in der Überprüfung des Faktorenmodells und der Erprobung kreativen Unterrichts erhärtet werden.

## 10.2 Diskussion

Die in der Einleitung der Arbeit aufgeworfenen Fragen können nun hinsichtlich der Forschungsergebnisse wie folgt zusammenfassend beantwortet werden.

### **Welches Verständnis von Kreativität sollte aus Sicht der Fachdidaktik Informatik verwendet werden?**

Es wurde eine allgemeine Definition von Kreativität herausgearbeitet, die vor dem Hintergrund der verschiedenen Betrachtungsmöglichkeiten von Kreativität vor allem dazu dient, eine Grundvorstellung von dem zu legen, was in dieser Arbeit unter Kreativität verstanden wird. Ein vergleichbares Verständnis von Kreativität wird intuitiv von praktizierenden Informatikern verwendet, die dabei die Bedeutung von Kreativität in ihrem Leben und für die Informatik unterstreichen. Auch in den Studien zum Faktorenmodell wurde ein personenbezogenes pragmatisches Verständnis von Kreativität deutlich.

Eine Definition verleitet dazu, Prozesse, Leistungen und Produkte hinsichtlich der Definition einzuordnen bzw. zu bewerten. In den verschiedenen Studien hat sich allerdings gezeigt, dass auch das Gefühl, etwas Kreatives zu tun, ausschlaggebend für positive Effekte beim Lernen sein kann. Dieses steht im engen Zusammenhang mit weiteren, für die Kreativität relevanten Faktoren, z. B. Selbstbestimmung. Zur thematischen Einbeziehung von Kreativität bieten sich alle im Kapitel 3 vorgestellten Perspektiven an. Diese können auch dazu dienen, Schüler an die Fachsicht von Kreativität heranzuführen. So sollte es das Ziel eines kreativen Informatikunterrichts vor allem sein, Kreativität aus der Fach-Perspektive zu verdeutlichen. Das bedeutet, die Schüler zu Leistungen zu befähigen, die sich aus informatischen Prozessen (wie Programmieren, Modellieren) ergeben und zu persönlich neuen und verwendbaren Ideen, Produkten, Lösungen oder Erkenntnissen führen.

### **Wie manifestiert sich Kreativität in der Informatik bzw. im Informatikunterricht?**

Das Faktorenmodell verdeutlicht die wesentlichen Variablen, die Kreativität im Informatikunterricht aus der Fach-Perspektive charakterisieren und unterstützen. Dazu zählt der zentrale Stellenwert kreativer informatischer Prozesse, der zentrale Stellenwert von Kreativität im Fach selbst und eine durch IKT geprägte Unterrichtsumgebung, welche Kreativität potentiell begünstigt.

### **Wie muss Informatikunterricht gestaltet werden, damit er das Potential, das Kreativität bietet, nutzt und fördert?**

Die Kriterien kreativen Informatikunterrichts gemeinsam mit dem Challenge-Cycle geben konkrete Gestaltungshinweise für die Planung und Durchführung kreativer Unterrichtsphasen.

### **Welche Auswirkungen hat die explizite Berücksichtigung von Kreativität im Informatikunterricht auf das Lernerlebnis und den Lernerfolg?**

Die Evaluation der Unterrichtssequenz zeigt erhöhte Motivation, Interesse, Leistungen und ein verbessertes Informatikbild der Schüler nach der kreativen Unterrichtssequenz im Pre-Post-Vergleich sowie im Vergleich mit dem Kontrollkurs. Diese Ergebnisse stehen im Einklang mit Studien in Hochschulen, in welchen sich die Berücksichtigung von Kreativität positiv auf die Motivation und den Lernerfolg auswirkte. In dieser Unterrichtssequenz arbeiteten die Schüler konzentriert und intrinsisch motiviert und ließen sich häufig selbst durch das Stundenklingeln nicht unterbrechen. Die Präsentationen und Veröffentlichung der Arbeitsergebnisse motivierten die Schüler für Folgeprojekte in den nächsten Unterrichtsstunden. Vor dem theoretischen Hintergrund konnte dies erwartet werden; die Anlage der Untersuchung lässt dennoch keine Verallgemeinerung zu. Hier sind weitere Untersuchungen notwendig.

Die Schüler beschreiben Informatik nun als Fach, in dem Kreativität benötigt wird und das ihnen Kreativität ermöglicht. Offenbar ist es gelungen, durch das Einbinden der Kreativität der Schüler, das Interesse der Schüler zu gewinnen; insbesondere auch der Mädchen des Kurses, deren Ergebnisse sich sogar von denen der Jungs abheben.

Das aus der kreativen Auseinandersetzung mit Informatik resultierende Bild von der Informatik stellt sich für das Fach als sehr attraktiv dar. Es ist eine Wahrnehmung von Informatik, welche die Schüler motiviert, sich mit den Möglichkeiten auseinanderzusetzen, die ihnen Informatik für die kreative Nutzung von IKT bietet. Durch die kreative Auseinandersetzung mit Informatik erwerben die Schüler „digital literacy“ und Fertigkeiten, die für die Beteiligung an der „Kreativen Gesellschaft“ essentiell sind. Sollte ein Transfer eines kreativen, und damit für viele sicherlich attraktiven Informatikbildes auf Eltern und Freunde gelingen, könnte sich allmählich das gesellschaftliche Bild von der Informatik ändern und Informatik wieder zu einem attraktivem Studienfach werden.

Informatiker und Informatiklehrer sollten für sich selbst überprüfen, ob das hier gezeichnete kreative Bild von der Informatik zur ihrer Vorstellung der Disziplin passt. Basierend auf bisheri-

gen Veröffentlichungen<sup>118</sup> zur Kreativität in der Informatik und der Anschaulichkeit stützt sich die Entwicklung des Faktorenmodells vor allem auf die Möglichkeiten, welche die Softwareentwicklung bietet. Gleichwohl kann das Faktorenmodell auch auf andere Aspekte des informatischen Modellierens angewandt werden, wobei der Begriff Softwareentwicklung allerdings bewusst die Entwicklung eines Produktes in die Begrifflichkeit einschließt. Der Kreativitätsbegriff ist soweit gefasst, dass hier auch Ideen und Lösungen, welche nicht implementiert werden, inbegriffen sind, sei es das Erfinden eines eigenen kryptographischen Chiffrierverfahrens, der Entwurf eines Automatenmodells für ein Verhalten aus der Lebenswelt der Schüler oder das Erstellen einer Prognose, wie Computer in 20 Jahren aussehen könnten. Dennoch ist zu vermuten, dass den Schülern der kreative Charakter der Informatik vor allem anhand der Entwicklung von Software deutlich wird.

Das Faktorenmodell stellt die grundlegenden Aspekte für die Planung kreativer Unterrichtsphasen dar, ist aber nicht als Unterrichtsmethode zu verstehen, welche jeglichem Unterricht zugrunde zu legen ist. Hier halte ich einen gesunden Mix verschiedener adäquater, der Lerngruppe und den Lerninhalten entsprechender Unterrichtsmethoden unter Beibehaltung regelmäßiger kreativer Unterrichtsphasen für sinnvoll. Wie die Lernforschung verdeutlicht, ist die Förderung von Kreativität auch eine langfristige Aufgabe.

Nicht erst seit Veröffentlichung der Bildungsstandards Informatik nimmt die Wichtigkeit von Vergleichbarkeit und Prüfbarkeit erreichter Kompetenzen der Schüler zu. Der Ruf nach mehr Kreativität und Freiraum für die Schüler mag in diesem Zusammenhang als deplaziert aufgefasst werden. Wie im Unterrichtsbeispiel demonstriert wurde, muss ein kreativer Ansatz nicht zu schlechteren Leistungen führen, sondern kann sogar ein besseres Verständnis und tiefgründigeres Lernen bei den Schülern erreichen.

Die kreative Nutzung der Informatik und der Informatiksysteme sind als neue Kompetenzen in der Schulinformatik meiner Meinung nach zwingend notwendig. Aktuelle Bemühungen, Schülern ein Informatiksystemverständnis zu vermitteln, können demnach immer nur erste Schritte sein, welche erst in der kreativen Anwendung ihre Bedeutung erhalten. Betrachtet man Kreativität als Beitrag der Informatik zur Allgemeinbildung, eröffnet und vereinfacht Informatikunterricht zahlreiche Betätigungsfelder, die nicht nur dabei helfen, in einer kreativen Wissensgesellschaft zurecht zu kommen, sondern auch noch Spaß machen.

### 10.3 Ausblick

Die Auswertung der Unterrichtssequenz verdeutlicht, dass in besonderem Maße die Interessen der Mädchen angesprochen wurden. Dagegen ist Informatik häufig ein Fach, das sich für Mädchen als uninteressant darstellt. Vor dem Hintergrund der Bemühungen, mehr Mädchen für die Informatik zu begeistern, ist das Einbinden der Schülerkreativität in den Unterricht und die Verdeutlichung der Kreativität der Informatik ein Ansatz, der sich lohnt, auch in dieser Hinsicht genauer untersucht zu werden. Nicht zuletzt wurde durch Studien bereits verdeutlicht, dass Mädchen deshalb seltener Informatik in Schulen und Universitäten belegen, weil sie in dem Fach mangelnde Möglichkeiten zur kreativen Entfaltung vermuten (vgl. Rich, Perry und Guzdial 2004; Guzdial und Soloway 2002). Verschiedene Studien in Hochschulen deuten darauf hin, dass mit Hilfe von Kreativität das Interesse von Frauen für die Informatik gewonnen werden kann (vgl. Margolis, Fisher und Miller 2000; AAUW 2000; Roberts, Kassianidou und Irani 2002).

Nach dieser ersten Betrachtung von Kreativität im Informatikunterricht empfiehlt es sich, einzelne Faktoren der Kreativität isoliert in ihrer Wirkung im Unterricht zu untersuchen. Des Weiteren bleibt die Frage ungeklärt, ob die positiven Ergebnisse der Erprobung nicht auch dem Werkzeug oder anderen Umständen zuzuschreiben sind. Hier empfiehlt sich ein empirischer Vergleich zweier Klassen, die mit dem gleichen Werkzeug, aber unterschiedlicher Methodik lernen.

---

<sup>118</sup> Fast die gesamte Literatur zur Kreativität in der informatischen Bildung scheint sich auf diesen Bereich zu beschränken.

Zur Unterstützung bei der Auswahl geeigneter Werkzeuge ist eine Evaluation und Kategorisierung von im Informatikunterricht verwendeten Werkzeugen opportun. Hierzu können Shneidermans Kriterien für kreativitätsunterstützende Softwarewerkzeuge (vgl. Kapitel 3, Abschnitt 3.4) herangezogen werden, die ggf. informatikunterrichtsspezifisch ergänzt werden sollten.

Um eine Etablierung kreativen Informatikunterrichts in der Unterrichtswirklichkeit tatsächlich zu erreichen, ist meiner Ansicht nach eine Vielzahl erprobter und gut beschriebener Beispiele notwendig.<sup>119</sup> Eine Ausarbeitung weiterer Beispiele ist geplant.

Weitere systematische Untersuchungen sind notwendig, um festzustellen, inwieweit sich in weiteren Inhaltsbereichen (z. B. theoretischer und technischer Informatik) ein kreativer Informatikunterricht umsetzen lässt. Ebenso fehlen noch publizierte Beispiele zur expliziten Einbeziehung von Kreativität in fortgeschrittenen Stufen mit komplexeren Konzepten (z. B. objektorientierte Modellierung oder deklarative Programmierung mit Prolog).

Mit dem Schuljahr 2008/09 wird die kontinuierliche Gestaltung kreativen Informatikunterrichts in einem Informatikkurs der Sekundarstufe I durchgeführt und untersucht. Hierzu wurde ein Unterrichtskonzept „Mediengestaltung Informatik“ mit dem Ziel entwickelt, durch kreatives Lernen die Mindestanforderungen der Bildungsstandards Informatik zu erfüllen, dabei aber weit darüber hinaus bei den Schülern Interesse und Begeisterung für die Informatik zu wecken. Eine Beschreibung des Vorhabens kann im Anhang nachgelesen werden.

## 10.4 Fazit

Der Informatikunterricht steht seit jeher vor der Herausforderung, dass sich die dem Unterricht zugrunde liegenden Paradigmen, Verständnisse und Werkzeuge häufig ändern (z. B. Programmierparadigmen). Zudem sind viele Grundlagen, die wir gegenwärtig noch für essentiell erachten, in wenigen Jahren vielleicht schon obsolet.<sup>120</sup>

Meiner Ansicht nach kann sich der Informatikunterricht dem nur entziehen, wenn die kreativen Möglichkeiten, die sich aus den grundlegenden Konzepten der Informatik ergeben, in den Vordergrund gestellt werden und damit vor allem Neugier und Spaß, aber auch kritisches Denken im Umgang mit Informations- und Kommunikationstechnologien, für die Schüler erlebbar werden.

---

<sup>119</sup> Offenbar nehmen Lehrer solche Beispiele gern an: Die Internetpublikation zum kreativen Einstieg in die Programmierung wurde bisher über 5000 mal aufgerufen. Ebenso erreicht mich regelmäßig positives Feedback von Lehrern, die diese Unterrichtssequenz einsetzen oder planen einzusetzen.

<sup>120</sup> Wie man bspw. an verschiedenen Bemühungen, den Schülern ein Systemverständnis für Informatiksysteme zu vermitteln, erkennen kann: Wurde vor Kurzem noch ein Verständnis für die E-Mail-Protokolle POP3 und SMTP als wichtig erachtet, spielen diese inzwischen aufgrund der Verwendung von Webmailern für immer weniger Nutzer eine Rolle.

## A Unterrichtsbeispiel: Ein kreativer Einstieg in die Programmierung<sup>121</sup>

### Einleitung

Wie sollte Programmieren eingeführt werden? Diese Fragestellung wird im Zusammenhang mit der Softwareentwicklung als zentralem Inhalt der Informatik immer wieder aufgeworfen. Nicht nur Schulen, auch Universitäten tun sich teilweise schwer daran, Schülerinnen und Schülern und Studentinnen und Studenten diesen Einstieg in die Informatik zu ebenen. Oftmals sind es gerade diese Kurse und Unterrichtsinhalte, die Informatik als schwer, techniklastig oder gar uninteressant und demotivierend erscheinen lassen. Dabei muss dies nicht so sein: Die Möglichkeit, in einer Programmiersprache eigene Software entwickeln zu können, kann sehr schön verdeutlichen, dass Informatiksysteme gestaltbar sind.

Zur Realisierung eines Unterrichtsbeispiels, welches die Schüler motiviert und gleichzeitig informatiknah lernen lässt, wurden Kriterien kreativen Informatikunterrichts (vgl. auch Romeike 2007b) herangezogen und berücksichtigt.



### Didaktische Richtlinien

#### Kriterien kreativen Informatikunterrichts

Vor dem Hintergrund, dass Informatikunterricht ein großes Potential hat, Kreativität zu fördern und dieses sich positiv auf die Schülermotivation und den Lernerfolg auswirken kann, wurden Kriterien formuliert, die für einen kreativen Informatikunterricht zu beachten sind. Diese wurden der Planung dieser Unterrichtssequenz zugrunde gelegt.

Die Kriterien umfassen:

#### Anforderungen an den Unterrichtsgegenstand:

- Relevanz
- Problemlösung oder Erstellen eines Produkts

<sup>121</sup> Dieser Teil wurde veröffentlicht in (Romeike 2007a).

Anforderungen an die Aufgabenstellungen:

- Subjektive Neuheit
- Offenheit in Ergebniserwartung und Lösungs-/Bearbeitungsweg
- Variable Bearbeitungstiefe
- Anwenden von Konzeptwissen
- Ideenanstregung

Schülerorientierte Anforderungen:

Ermöglichen von

- Identifikation mit dem Unterrichtsgegenstand
- Originalität

Anforderungen an die Unterrichtsumgebung

- Experimentieren
- Zeitlicher Raum
- Unterrichtsklima der Vielfalt
- Lehrerrolle – Lehrer als Coach

Weitere didaktische Grundlagen:

Entdeckendes Lernen, Experimentelles Lernen, Learning by Design, Konstruktivismus (Papert)

Unterrichtsbegleitende Hinweise:

- Immer neue Anweisungsblöcke im Heft mit Funktion notieren.
- nach jedem Projekt Vorstellen der verschiedenen Ergebnisse
- Online-Upload aller Programme auf die Scratch-Homepage

Objektorientierung ja/nein?

„Objects first“ oder „Objects last“ ist eine Diskussion, die schon eine Weile auch für die Schulformatik geführt wird. In mehreren Dissertationen wurde ein objektorientierter Ansatz für den Schulunterricht empfohlen. Dieser findet sich auch in vielen neuen Rahmenlehrplänen wieder. Dennoch ist das Verständnis für die Konzepte der Objektorientierung (OO) bei vielen Informatik-lehrerinnen und Lehrern noch vage (vgl. Kohl und Romeike 2006), oftmals wird bei der Einführung in die Programmierung ein Weg gegangen, der sich an der strukturierten, imperativen Programmierung orientiert. Das hier vorgestellte Vorgehen lässt beide Wege offen – in Scratch finden sich wesentliche Aspekte der imperativen Programmierung wieder. Dennoch baut Scratch auf grundlegenden OO-Konzepten auf: Programme (Skripte/Methoden) werden für Objekte geschrieben, diese basieren im Wesentlichen auf der Manipulation ihrer Attribute, zur Steuerung werden Botschaften verwendet, die an alle Objekte gesendet werden. Die Entwicklungsumgebung basiert auf Squeak (Small talk) und verfolgt damit das objektorientiert Paradigma, welches aber nicht aufdringlich, sondern vielmehr intuitiv deutlich wird. Vielmehr helfen die Konzepte der OO, die Funktion von Objekten in Scratch spielerisch zu verstehen.



Hier soll vielmehr ein Weg aufgezeigt werden, wie die Schülerinnen und Schüler unter Berücksichtigung der Kriterien kreativen Informatikunterrichts entdeckend und mit viel Freude an der Sache in die Programmierung eingeführt werden. Denn eins steht im Mittelpunkt: „Informatik macht Spaß!“. So wurde diese Unterrichtssequenz, die in einer elften Klasse erprobt und evaluiert wurde, auch von den Schülern aufgenommen.

### **Lernziele und Kompetenzen:**

Die Unterrichtssequenz verfolgt in Anlehnung an die Bildungsstandards Informatik die Umsetzung der folgenden Lernziele und das Herausbilden folgender Kompetenzen der Schülerinnen und Schüler:

1. Sie entwickeln ein grundlegendes Programmierverständnis
2. Sie beschreiben die Eigenschaften von Algorithmen, nennen Beispiele für erfüllte und verletzte Eigenschaften und entscheiden, ob ein Problem algorithmisch lösbar ist.
3. Sie verwenden algorithmische Grundbausteine zur Programmierung
  - Sequenz, Schleife, Entscheidung, Variablen (lokale/globale)
  - Eingabe und Ausgabe von Information
  - Rechenoperationen und Vergleichsoperatoren
  - Objekt, Botschaft, Attribut, Methode
4. Sie stellen Algorithmen dar
  - mit Scratch-Bausteinen
5. Sie verstehen von Algorithmen und Programmtexte:
  - Lesen und analysieren Programme
  - Implementieren, modifizieren und erweitern Programme
6. Sie entwerfen, implementieren und testen Programme
7. Sie lösen Probleme selbstständig
8. Sie verwenden Hilfesysteme selbstständig

Darüber hinaus werden Prozesskompetenzen angesprochen, die das Folgende umfassen:

9. Modellieren und implementieren
10. Begründen und Bewerten
11. Strukturieren und Vernetzen
12. Kommunizieren und Kooperieren
13. Darstellen und Interpretieren

### **Zielgruppe:**

Die primäre Zielgruppe der Entwickler waren vor allem Grundschüler. Dessen ungeachtet wird Scratch wegen seiner Unterstützung einer intuitiven Herangehensweise auch in den Sekundarstufen I und II, ja sogar in Einführungskursen an der Universität verwendet.

Nicht nur vor diesem Hintergrund, sondern auch von der Offenheit in Bearbeitungstiefe und Zeitplanung kann diese Unterrichtssequenz sowohl in der Sekundarstufe I als auch einleitend in der Sekundarstufe II verwendet werden. Nicht zuletzt kann ein Unterrichtsvorschlag immer nur Ideen liefern, die an die jeweilige Lerngruppe und –situation angepasst werden sollten.

## Ein kreativer Einstieg in die Programmierung mit Scratch

Im Folgenden werden jeweils zuerst die Planungen der einzelnen Unterrichtsphasen skizziert, anschließend werden die Unterrichtserfahrungen geschildert und Schülerergebnisse gezeigt.

### Einführung in die Oberfläche, grundlegende Funktionen und Hilfefunktion

1. Einführungstutorial gemeinsam mit den Schülern:
2. Verwenden und Bewegen von Objekten.
3. Bausteine: Vorwärts, SpieleSchlagzeug, Sage, Denke und Background
4. Beschränkung auf Kategorien Bewegung, Kontrolle, Aussehen, Klang
5. Zeigen eines kleinen [Beispiels](#) für einen Animationsfilm
6. Aufgabe: „Erstelle einen kurzen Animationsfilm, verwende Bausteine aus o.g. Kategorien.“
7. Vorstellen verschiedener Filme, Schüler erklären verwendete Bausteine und Strategien

*Verbleibende Zeit für den Beginn einer neuen UE und die Einführung in Scratch: Eine Stunde. Die Schüler haben vorher das Thema Datenbanken abgeschlossen, einen Test geschrieben und gemeinsam ausgewertet.*

*Die Stimmung wird merklich gut, als wir „endlich“ zum „Kern“ der Informatik kommen – zum eigenen Entwickeln von Programmen.*

*Mit der Schritt-für-Schritt-Einführung werden die Schülerinnen und Schüler mit den Grundlagen von Scratch vertraut gemacht. Dabei werden wesentliche Funktionen, die Hilfe, Sprites sowie deren Attribute im Klassendiagramm dargestellt.*

*Als Kontrollstruktur wird die Sequenz eingeführt und als Beispiel eine einfache Animation gezeigt.*

*Die Aufgabe ist nun: Erstelle einen Kurzfilm.*

*Nach etwa 20 min hat jede Schülergruppe einen kleinen Kurzfilm erstellt.*

*Anschließend wird demonstriert, wie ein Scratch-Projekt online gestellt werden kann. Einige Schüler stellen ihr Unterrichtsergebnis schon auf die Scratch-Website.*

*Eine knappe Woche später stelle ich erstaunt fest, dass Schüler eines Parallel- (Leistungs-!) Kurses mit dem, von einem meiner Schüler erstellten, Spiel online spielen. Dieser Schüler wurde nun gebeten, den Schülerinnen und Schülern des Leistungskurses sein Programm und Scratch vorzustellen – der LK schwenkt um auf Scratch!*

**Wiederholungen (Schleifen), Erstellen von Animationen/ Filmen**

1. Objekte
2. Sequenzen mit Grundbausteinen, Wiederholungen (Schleifen)
3. Storyboard
4. Erstellen eines eigenen Animationsfilms
5. Upload des Films auf Scratch-Website
6. Algorithmen und deren Eigenschaften

Gemeinsam an der Tafel entwickeln:

Objekte (Sprites) besitzen Attribute und Methoden (Anschreiben des Klassendiagramms für Sprites).

Gemeinsame Attribute sind: x, y, Richtung, aktuellesKostüm, Größe in Prozent (ggf. Rotationstyp (in IDE festzulegen))

Methoden werden als Skripte in der Scripts Area aus Bausteinen zusammengefügt.

Das Ausführen von mehreren Bausteinen (Anweisungen) nacheinander nennt man Sequenz.

Mitunter möchte man mehrfach einen Baustein oder eine Sequenz ausführen: Hierfür verwendet man Wiederholungen (Schleifen) (Demonstrieren).

Aufgabe des Tages: Entwerfen und Erstellen eines längeren Animationsfilms.

Als Beispiel wird der Film „Polarbear School“ gezeigt.

Gemeinsam mit den Schülern werden anhand des Beispielfilms folgende Feststellungen gemacht:

- Es werden verschiedene Sprites verwendet.
- Handlungsabläufe geschehen mittels Pausenplanung.

Zum Entwurf eines Films eignet es sich die Handlungen der einzelnen Akteure (Objekte/Sprites) in einem Storyboard festzuhalten

*Ein Storyboard bildet die Inhalte und Handlungsverläufe eines Drehbuchs visuell ab und dient so als Vorlage für die Erstellung eines Films. Die Handlungen aller Akteure werden hierbei auf einem Zeitstrahl dargestellt.*

Gemeinsam wird nun der Beginn des Storyboards für den Beispielfilm an der Tafel erarbeitet.

Anschließend erhalten die Schüler folgenden Arbeitsauftrag:

*Entwerfe und realisiere einen eigenen Animationsfilm zu einem Thema deiner Wahl. Überlege dir zuerst, welche Objekte (Sprites) in dem Film vorkommen sollen und welche Haupthandlungen dargestellt werden sollen.*

*Schreibe ein Storyboard für den Film und realisiere diesen anschließend mit Scratch.*

*Wenn du fertig bist, verwende den „Share“ Button um das Programm online zu stellen.*

*Mögliche Ideen für das Thema des Films können eine Szene aus deinem Lieblingsfilm, dein Schultag oder ein Besuch im Zoo sein. Du kannst aber auch andere Themen darstellen.*

In den letzten 20 Minuten wird der Begriff des Algorithmus mit seinen Eigenschaften eingeführt. Die Schüler sollen feststellen, ob es sich bei ihrem Animationsfilm um einen Algorithmus handelt.

**SPRITE**

x-Position

y-Position

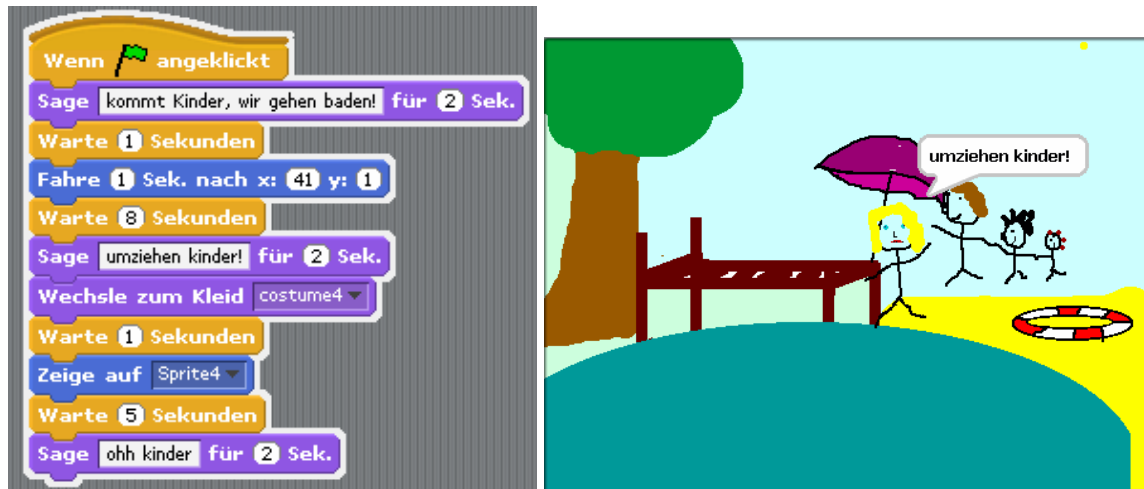
Richtung

Größe

aktuellesKostüm

Skript 1

Skript 2



Nach der Vorstellung einiger Schülerergebnisse wollen alle Schülerinnen und Schüler mit Scratch loslegen bzw. weiterarbeiten. Vorher wiederholen wir allerdings wesentliche Konzepte (einige Schüler fehlten in der letzten Stunde): Sprites als Objekte und die Kontrollstruktur Sequenz. Als neue Kontrollstruktur wird die (inzwischen schon von einigen Schülern verwendete) Wiederholung eingeführt.

Nun soll es um die Erstellung eines längeren Animationsfilms gehen. Als Grundlage für die Erstellung des Animationsfilms wird die Methode des Storyboards eingeführt – eine Skizze des Films, die die Handlungen der Akteure auf einem Zeitstrahl darstellt. Zur Motivation wird ein Beispielprojekt („Polarbear School“) gezeigt, welches verschiedene Möglichkeiten mit Objekten und Bühnen umzugehen, demonstriert.

Zu guter Letzt wird wiederholt, wie die Schülerinnen und Schüler sich registrieren und ihre Projekte auf die Scratch-Website laden können. Nun wollen sie aber auch loslegen.

Da die Schülerinnen und Schüler einige Zeit für das Ausprobieren und das Kennenlernen der Entwicklungsumgebung benötigten, musste bis zum Ende der 3 Stunden Zeit gelassen werden.

Schülerinnen und Schüler, die es nicht schafften, ihren Film in dieser Zeit zu Ende zu bringen, wollten das unbedingt zu Hause weitermachen – in der nächsten Woche werden die Ergebnisse gesammelt und der Klasse vorgestellt. Mal sehen, wie die Projekte geworden sind.

In der letzten halben Stunde sollte das Thema „Algorithmen“ besprochen werden. Dazu kam es aus o. g. Grund nicht mehr. Die Schülerinnen und Schüler erhielten stattdessen die Hausaufgabe, sich die Eigenschaften von Algorithmen an einem Arbeitsblatt zu erarbeiten, ihr Projekt anhand der Eigenschaften zu prüfen und zu entscheiden, ob es sich um einen Algorithmus handelt.

### **Algorithmen und das Reagieren auf Ereignisse: Entscheidungen**

1. Vorstellen der Filme der Schülerinnen und Schüler unter Nutzung der Uploads auf die Scratch-Website
2. Thematisierung der Eigenschaften von Algorithmen Überprüfung dieser an einem (von einem Schüler erstellten) Animationsfilm.

Erkenntnis: Ein Film ist kein Algorithmus: denn ein Algorithmus löst eine Klasse von Problemen. Andere Eigenschaften von Algorithmen sind erfüllt: Eindeutigkeit: Aus jedem Schritt ist klar welcher Schritt danach gemacht wird, Endlichkeit (begrenzte Anzahl an Anweisungen), Ausführbarkeit (Skript wird ausgeführt)

1. Einführen des Bausteins „Wenn“
2. Zeichnen eines Sprites „Ball“, Endlose Vorwärtsschritte, wenn „edge-wird berührt“ dann Bausteine „Abdrehen wenn Kante berührt“
3. Hinweis: Hier handelt es sich zwar um eine doppelte Abfrage – der Baustein kann auch ohne „Wenn“ verwendet werden. Das Vorgehen erweist sich aber in diesem Beispiel für das Verständnis der Entscheidung als pragmatisch..
4. Zu Beginn wird die Richtung per Zufall bestimmt
5. Arbeitsauftrag: Füge weitere Bälle oder andere Objekte hinzu, die sich bewegen und aneinander abprallen!

### **Interaktivität: Reagieren auf Benutzereingaben**

Mini-Projekt: Animieren von Namen.

Frage: Wer hat schon Möglichkeiten entdeckt, wie man Einfluss auf die Programme nehmen kann? Welche Möglichkeiten für Ein- und Ausgaben gibt es?

Eine Schülerin oder ein Schüler stellt das folgende vor (es ist zu erwarten, dass einige Schülerinnen und Schüler schon mehr Bausteine ausprobiert haben)

#### Eingaben:

- Wenn Taste x – gedrückt tue – Starten verschiedener Skripte, Reagieren auf eingaben
- Wenn Maustaste gedrückt – Starten verschiedener Skripte, Reagieren auf Eingaben
- x und y- Werte der Maus

#### Ausgaben:

- Darstellungen auf dem Bildschirm mit Sprites
- Sagen
- Denken
- Sound

Nun werden verschiedene Beispiele zu Namensanimationen vorgestellt und anschließend folgender Arbeitsauftrag gegeben:

Animiere deinen Namen, indem die Buchstaben auf Mausbewegungen und/oder Mausclicks reagieren.

HA: Überlege dir je fünf Beispiele für Abläufe aus dem täglichen Leben, die durch einen Algorithmus gelöst werden können (Beispiel: Addieren zweier Zahlen) bzw. die nicht durch einen Algorithmus beschrieben werden können (Beispiel: Schreiben eines Bewerbungsbriefts).

*Einige Schülerinnen und Schüler haben sich Scratch inzwischen zu Hause installiert.*

*Während die Schülerinnen und Schüler anfangs noch zögern, ihre Animationsfilme vorzustellen, sollen später sogar noch mehr Filme gezeigt werden.*

*Das Erstellen der Springbälle klappt gut – alle Schülerinnen und Schüler bewältigen diese Aufgabe gut, wer schneller ist, implementiert drei und mehr Bälle. Parallel werden die Projekte auf die Scratch-Website hochgeladen.*



*Die Möglichkeiten der Internaktion wurden durch die Schülerinnen und Schüler gemeinsam zusammengetragen und die Bausteine exemplarisch vorgestellt..*

*Das Vorstellen der Namensanimationen brachte ebenfalls interessante Ideen und Anregungen für alle. Auch wenn viele Schülerinnen und Schüler die Namensanimationen gut umsetzten, fehlte bei einigen allerdings die Interaktivität – es handelte sich wieder nur um Filme. Hier musste nochmals auf das Reagieren auf Ereignisse eingegangen werden.*

*Zeitlich waren 20 bis 30 Minuten für das Animieren der Bälle ausreichend. Für das Animieren des Namens (45 min) hätte etwas mehr Zeit sein können. Allerdings hatte der straffe Zeitplan bis dahin einen Vorteil: Die Schülerinnen und Schüler waren an den neuen Ideen immer noch sehr interessiert und arbeiteten ausnahmslos an den Mini-Projekten gut mit und ließen sich gute Ideen einfallen. Dabei wurden gestalterische Freiheiten als auch algorithmische Kreationen ausgelebt.*



Die Inhalte der folgenden beiden Unterrichtsphasen werden den Schülerinnen und Schülern als Arbeitsblätter ausgehändigt. Sie haben nun fast drei Unterrichtsstunden Zeit, sich die Konzepte an den Beispielen und eigenen Programmen zu verdeutlichen und umzusetzen.

## Methoden und Botschaften

### Methode: Verhaltensweise eines Objektes (Sprites).

In Scratch wird eine Methode durch ein Script beschrieben.

Das Ausführen einer Methode kann veranlasst werden durch:

- Start des Programms (Grüne Fahne)
- Tastendruck
- Anklicken des Sprites
- Erhalten einer bestimmten ↑Botschaft

Dies hängt jeweils vom entsprechenden Startbaustein ab.

Dabei können auch mehrere Methoden auch gleichzeitig durchgeführt werden.



### Botschaften: Kommunikation von Objekten (Sprites) miteinander.

„Durch Botschaften können Objekte miteinander kommunizieren („sprechen“). So ist es z.B. möglich, mit einer Botschaft von einem Objekt andere Objekte anzustoßen, eine bestimmte ↑Methode auszuführen.

Das Senden von Botschaften erfolgt mittels des Bausteins



### Beispiel:

Eine Tanzgruppe soll auf Anweisung ihres (Klick auf) Trainers bewegen.

Skript des Sprite „Trainer“:



Skript der drei Sprites „Dancer“:



### Aufgabe:

1. Erstelle eine eigene Animation einer Tanzgruppe. Die Tänzer sollen auf verschiedene Botschaften reagieren und unterschiedliche Tanzbewegungen vollziehen.

**Variablen: Platzhalter für Werte.**

**Variablen und Rechnen**

Wenn ein richtiges Spiel entworfen werden soll, will man auch wissen, wie gut man abgeschnitten hat. Hierzu wird ein Zähler benötigt. Da ein Spielstand sich während des Spiels ändert, also variabel ist, benutzt man **Variablen**, um den Spielstand zu speichern.

Erstellt man eine Variable, muss man sich überlegen, wer alles auf die Variable Zugriff benötigt – wenn nur ein Sprite die Variable benutzen soll, beschränkt man sich auf den Zugriff für nur dieses Sprite (**lokale Variable**). Sollen alle Sprites die Variable verändern und benutzen können, verwendet man **globale Variablen**.



**Initialisierung**

Damit eine Zählvariable bei jedem Programmstart anfängt von vorne zu zählen muss sie initialisiert (auf einen Ursprungszustand gebracht) werden.

**Aufgabe:**

Füge einen Zähler zu der Tanzgruppe hinzu der zählt, wie fleißig die einzelnen Tänzer waren und für jede Bewegung einen Punkt nach oben zählt.

**Rechnen:**

Variablen eignen sich sehr gut um mit ihnen zu rechnen. Scratch unterstützt die Grundrechenoperationen und einige mehr.

Möchte man nach je zehn Tanzbewegungen einen Klang spielen möchte, kann bspw. folgendes Skript verwenden.



**Aufgabe:**

Füge zu einem deiner bisher erstellten Projekte verschiedene Zähler hinzu. Wenn ein bestimmter Zählerstand erreicht ist, Spiele einen Klang o.ä.

*Der Unterricht wird diesmal mit einer mündlichen Leistungskontrolle zu Algorithmen, den kennengelernten Kontrollstrukturen und den Hausaufgaben begonnen. Die Klasse ist relativ ungeduldig und möchte mit Scratch weiterarbeiten.*

*Vornweg werden weitere Ergebnisse aus der letzten Woche gezeigt. Nach jedem Programm gibt es sogar einen kleinen Applaus für die Programmierer. Anschließend wird das Vorhaben für den Tag und die nächste Woche mitgeteilt: Zwei Arbeitsblätter mit neuen Konzepten, Erklärungen dazu und Aufgaben sowie ein eigenes Spiele-Projekt. Mit Austeilen der Arbeitsblätter legen die Schülerinnen und Schüler auch schon los.*

*Das Verwenden der Botschaften funktioniert ohne größere Probleme. Wo Schwierigkeiten auftauchen, helfen sich die Schülerinnen und Schüler meist gegenseitig oder schauen selbst im Hilfesystem von Scratch nach. Einige Schülerinnen und Schüler haben bereits vorher Botschaften und Variablen eingesetzt. Bei der Aufgabenstellung, eine Tanzgruppe zu animieren, entstehen wieder vielfältige Umsetzungen. Teilweise gestalten die Schülerinnen und Schüler die Tanzgruppe interaktiv, so dass bspw. auf Mausclick Botschaften versandt und damit Aktionen gestartet werden. Teilweise geschehen die Methodenaufrufe vom „Trainersprite“ aus, so dass keine Interaktionen möglich sind. In diesen Fällen bespreche ich mit diesen Schülerinnen und Schülern kurz das Programm und wie sie Interaktionen einbauen könnten (was sie aber nicht müssen) und stelle so sicher, dass das Prinzip von Algorithmen (nicht immer der gleiche Ablauf) verstanden wurde. Auch das Verwenden von Variablen macht keine Probleme, so dass die meisten Schülerinnen und Schüler in der dritten Stunde bereits mit dem eigenen Projekt anfangen.*

*Dabei fiel mir im Übrigen ein weiterer Effekt der eigenständigen kreativen Projektarbeit im Unterricht auf: Als ich die letzte der drei Stunden beendete und es zur großen Pause klingelte, stürmten die Schülerinnen und Schüler nicht – wie häufig – schnell nach draußen. Als wäre nichts gewesen, blieben die meisten sitzen, brachten noch das Umsetzen ihrer Ideen zu Ende und verließen dann erst nach und nach den Raum. Das Interesse scheint geweckt und beständig zu sein.*

**Projekt:**

Bis hierher wurden elementare Konzepte der Programmierung vermittelt. Von nun an sollen die Schülerinnen und Schüler ihr erlerntes Wissen in einem eigenen kleinen Projekt anwenden:

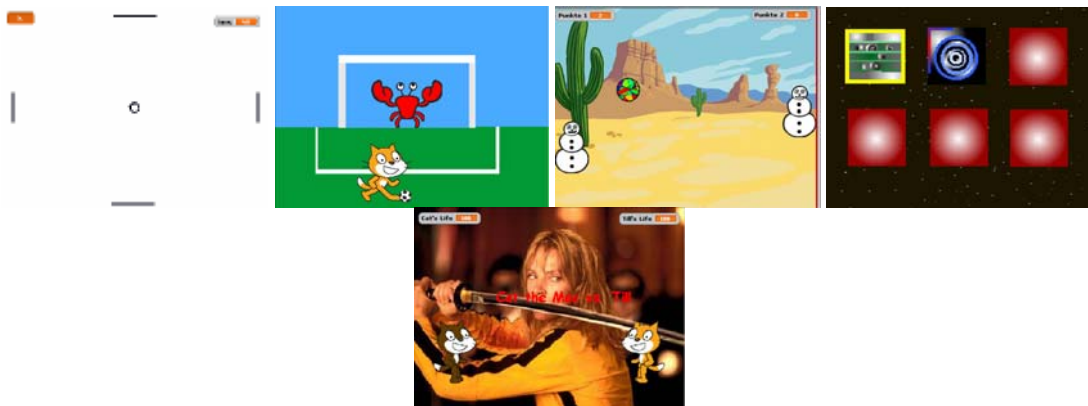
**Aufgabe:**

Entwickle ein eigenes Spiel in Scratch, in welchem du alle bisher kennengelernten Konzepte verwendest (verschiedene Sprites, Botschaften, Schleifen, Entscheidungen und Variablen, ...).

Ideen können bspw. solchen Spielen wie Mario Brothers, Pong, Moorhuhn, Autorennen, Memory usw. entstammen.

*Den Auftakt der letzten drei Unterrichtsstunden bildete wieder das Präsentieren von Ergebnissen aus der letzten Woche durch die Schülerinnen und Schüler. Nachdem sie sich über einige Ideen und Vorgehensweisen ausgetauscht hatten, wollten sie nun an ihrem Projekt weiterarbeiten. Leider konnte es nur ein kleines Projekt werden, da nur 1,5 Stunden Zeit übrig blieb – in der letzten Stunde sollte eine Leistungskontrolle zu Algorithmen und Programmierung geschrieben werden. Jeder Schüler bzw. jede Schülergruppe schaffte es aber, vorher ein kleines Spiel fertig zu stellen.*

Die 45-minütige Leistungskontrolle umfasste einen theoretischen und einen praktischen Teil – wobei der praktische Teil aus einer Problemlöse- und einer freien kreativen Aufgabe bestand. Diese wurde von der Mehrheit der Schülerinnen und Schüler gut bis sehr gut bewältigt. Bisher habe ich bei der Einführung in die Programmierung noch mit keinem Kurs so schnell so gute Ergebnisse erzielt.



### Möglicher weiterer Unterrichtsverlauf:

Mit Sicherheit kann die weitere Arbeit mit den Schülerinnen und Schülern auf das bisher Erreichte aufbauen. Insoweit könnten weitere Projekt-Ideen und Unterrichtsinhalte angepackt werden, beispielsweise:

- Interaktive Musik
- Algorithmische Computergrafik
- Simulationen

Ebenso können weitere Arbeitstechniken bei den Schülerinnen und Schülern entwickelt werden, zum Beispiel

*Problemfindung und Problemlösung mit Scratch:*

Stelle einem Tauschpartner eine Aufgabe, die er mit Scratch lösen soll.

Formuliere diese Aufgabe schriftlich.

Erstelle selbst eine Lösung für diese Aufgabe mit Scratch, aber zeige sie deinem Tauschpartner nicht. Tauscht nun eure Aufgaben und versucht jeweils die Aufgabe des anderen zu lösen.

Wenn dein Tauschpartner deine Aufgabe gelöst hat, stellt euch gegenseitig die Lösungen vor und vergleicht sie hinsichtlich der fehlerfreien Lösung der Aufgabe, der Benutzerfreundlichkeit, der Effizienz und der Eleganz der Scripte.

Scratch bietet einen motivierenden Einstieg in die Programmierung mit geringen Einstiegshürden. Gleichwohl können aber auch viele anspruchsvolle Projekte umgesetzt werden.

Auf Scratch aufbauend könnte es sich anbieten mit Squeak weiterzuarbeiten. Squeak unterstützt dann auch den Textmodus und weitere, in Scratch nicht vorhandene, Konzepte.

## B Fragebogen zur Unterrichtserprobung



### Fragebogen zum Informatikunterricht

#### 1. Fragen zur Person

Geschlecht

Männlich

Weiblich

	Ihr Code
Erster Buchstabe des Vornamens Ihrer Mutter	
Erster Buchstabe des Vornamens Ihres Vaters	
Letzter Buchstabe des eigenen Nachnamens	
Erste beide Ziffern des eigenen Geburtsjahrs, ggf. mit 0 beginnen (nicht Monat)	

Bearbeitungsdatum: \_\_\_\_\_

Anzahl der versäumten Unterrichtstermine: \_\_\_\_\_

#### Befragung zum Informatikunterricht allgemein

Informatikunterricht macht mir Spaß.

Ich finde die Inhalte im Informatikunterricht interessant.

Ich verstehe den Unterrichtsstoff im Fach Informatik.

Im Informatikunterricht kann ich kreativ sein.

Ich kann zu Hause oder bei Freunden mit Ergebnissen (Software, Dateien, Dokumente etc.) aus dem Informatikunterricht beeindruckt werden.

Im Informatikunterricht sind für eine Aufgabe verschiedene Lösungswege und Lösungen möglich.

Im Informatikunterricht wird viel experimentiert.

Ich beteilige mich am Informatikunterricht.

Ich bin im Informatikunterricht abgelenkt.

	trifft völlig zu	trifft eher zu	trifft nicht zu	kann ich nicht beurteilen
Informatikunterricht macht mir Spaß.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich finde die Inhalte im Informatikunterricht interessant.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich verstehe den Unterrichtsstoff im Fach Informatik.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Im Informatikunterricht kann ich kreativ sein.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich kann zu Hause oder bei Freunden mit Ergebnissen (Software, Dateien, Dokumente etc.) aus dem Informatikunterricht beeindruckt werden.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Im Informatikunterricht sind für eine Aufgabe verschiedene Lösungswege und Lösungen möglich.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Im Informatikunterricht wird viel experimentiert.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich beteilige mich am Informatikunterricht.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich bin im Informatikunterricht abgelenkt.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Wie schätzen Sie den Schwierigkeitsgrad des vorgestellten Lernstoffs der letzten Unterrichtsreihe ein?

Der Schwierigkeitsgrad war ...

hoch

angemessen

niedrig

Der Stoffumfang war ...

viel

angemessen

wenig

Ich habe im Unterricht gelernt ...

Geben Sie bitte an, in welchem Maße Sie den vorgegebenen Aussagen zustimmen.

	ich stimme überhaupt nicht zu	ich stimme nicht zu	ich stimme etwas zu	ich stimme voll zu	ich kann ich nicht beurteilen
Darstellung und Veranschaulichung der Lerninhalte waren verständlich.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Das praktische Üben war ausreichend	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die gestellten Aufgaben waren lösbar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich konnte meine Ideen im Unterricht einbringen und umsetzen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Inhalte sind vermutlich für den Umgang mit Computersystemen sehr nützlich.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Verwendbarkeit und der Nutzen des behandelten Stoffes wurden deutlich.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Hilfsmittel zur Unterstützung des Lernens (z.B. Arbeitsblätter, Software) sind ausreichend und in guter Qualität vorhanden.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

#### Befragung zum Unterrichtsthema

Der Unterricht beschäftigte sich mit Aufgaben, die mir im täglichen Leben begegnen.

Ich konnte im Unterricht etwas Neues entdecken.

Einige Themen haben mich besonders interessiert.

Ich habe auch außerhalb des Unterrichts über die Aufgaben nachgedacht.

Ich konnte mich leicht auf die Sache konzentrieren.

Ich habe das Gefühl, für mich etwas dazugelernt zu haben.

Ich kann mir vorstellen, dass ich das erworbene Wissen in Zukunft gebrauchen kann.

Es hat mir Spaß gemacht, mein Verständnis für dieses Thema zu vertiefen.

	trifft völlig zu	trifft eher zu	trifft nicht zu	kann ich nicht beurteilen
Der Unterricht beschäftigte sich mit Aufgaben, die mir im täglichen Leben begegnen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich konnte im Unterricht etwas Neues entdecken.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Einige Themen haben mich besonders interessiert.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich habe auch außerhalb des Unterrichts über die Aufgaben nachgedacht.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich konnte mich leicht auf die Sache konzentrieren.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich habe das Gefühl, für mich etwas dazugelernt zu haben.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich kann mir vorstellen, dass ich das erworbene Wissen in Zukunft gebrauchen kann.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Es hat mir Spaß gemacht, mein Verständnis für dieses Thema zu vertiefen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

<b>Wie schätzen Sie den Lernerfolg im Kurs für die letzte Unterrichtsreihe ein?</b>	
a) Ausgewogen – alle haben die Inhalte gut bis sehr gut verstanden	<input type="checkbox"/>
b) Ausgewogen - alle haben die Inhalte nur oberflächlich oder gar nicht verstanden	<input type="checkbox"/>
c) Viele Mitschüler kamen gut bis sehr gut mit und wenige hatten größere Probleme mit dem Stoff	<input type="checkbox"/>
d) Viele Mitschüler kamen mittelmäßig bis schlecht mit und wenige hatten keine Probleme	<input type="checkbox"/>
e) weiß ich nicht	<input type="checkbox"/>
<b>Wie schätzen Sie Ihren eigenen Lernerfolg ein?</b>	
a) hoch - ich habe alle Inhalte gut bis sehr gut verstanden	<input type="checkbox"/>
b) eher hoch - die wichtigsten Inhalte habe ich verstanden	<input type="checkbox"/>
c) eher gering - vieles habe ich nur oberflächlich verstanden	<input type="checkbox"/>
d) gering - viele Inhalte bereiten mir noch große Schwierigkeiten	<input type="checkbox"/>
e) weiß ich nicht	<input type="checkbox"/>
<b>Wodurch haben Sie am meisten gelernt? (max. 2 ankreuzen)</b>	
a) Erklärungen des Lehrers im Unterricht	<input type="checkbox"/>
b) Austausch mit Mitschülern	<input type="checkbox"/>
c) Arbeit am Projekt	<input type="checkbox"/>
d) Vorstellung der Projektergebnisse	<input type="checkbox"/>
e) eigene Bearbeitung der Aufgaben	<input type="checkbox"/>
f) Vorstellung der Aufgaben	<input type="checkbox"/>
g) Sonstiges: _____	<input type="checkbox"/>
<b>Fiel Ihnen der behandelte Unterrichtsstoff dieser Unterrichtsreihe allgemein leicht oder schwer?</b>	
a) Ziemlich leicht	<input type="checkbox"/>
b) Eher leicht	<input type="checkbox"/>
c) Mal leicht, mal schwer	<input type="checkbox"/>
d) Eher schwer	<input type="checkbox"/>
e) Ziemlich schwer	<input type="checkbox"/>

Anmerkungen:

## C Lernerfolgskontrolle: Algorithmen und Programmierung

1. Was ist ein Algorithmus?
  - a) Beschreiben Sie, was Sie unter dem Begriff verstehen.
  - b) Geben Sie die 4 Eigenschaften von Algorithmen an und erläutern Sie jede kurz (1-2 Sätze).
  - c) Entscheiden Sie, welche der Prozesse durch Algorithmen beschrieben werden können!

Prozess	Ja	Nein	Begründung
Lösen einer Quadratischen Gleichung der Form $ax^2 + bx + c = 0$			
Schreiben eines Liebesbriefes			
Ordnen von 1000 gebrochenen Zahlen nach ihrer Größe			
Ermitteln der Häufigkeit eines Wortes in einem Text			
Leiten einer Diskussion			
Konstruieren eines Kreises durch drei nicht auf einer Gerade liegenden Punkte			
Schießen eines Tores beim Handball			

2. Nennen und erläutern Sie kurz 3 Kontrollstrukturen, die bei der Beschreibung von Algorithmen verwendet werden können. Geben Sie jeweils ein kleines Beispiel an, welches genau illustriert, warum die jeweilige Kontrollstruktur verwendet werden muss.

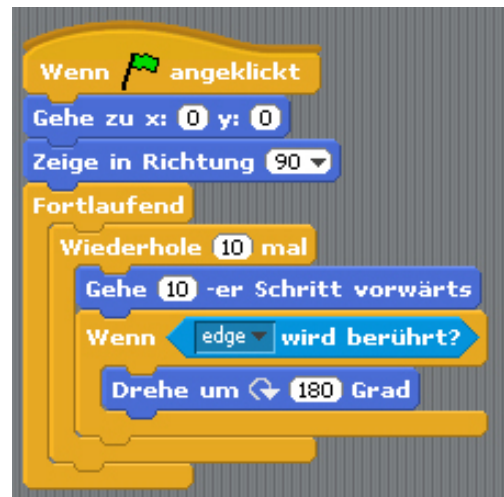
Kontrollstruktur	Funktion	Beispiel
1.		
2.		
3.		

3. Geben seien die folgenden Programme.
- Beschreiben Sie die Funktion des Programms kurz mit eigenen Worten. (Was macht das Sprite?)
  - Gibt es eine Möglichkeit, das Programm zu verbessern (So dass es kürzer/ effizienter/ besser ist?). Wenn ja, geben Sie die Möglichkeit an, wenn nein, begründen Sie warum nicht!

Beispiel 1:



Beispiel 2:



Bei beiden Programmen verwendetes Sprite:



4. Erstellen Sie ein Programm in Scratch, in welchem die Katze bis zum Baum läuft, dort umdreht und dann bis zum Ende des linken Bildschirms läuft (siehe Bilder).

Verwenden Sie die Vorlage: Aufgabe\_4.sb.

Speichern Sie Ihr Ergebnis im Testate-Ordner unter dem Namen *A4IhrName.sb*.

Anfangssituation:



Ende



5. Erstellen Sie ein kleines Programm in Scratch, in welchem Sie die 3 kennen gelernten Kontrollstrukturen sowie Botschaften und Variablen verwenden.

Beschreiben Sie in einem Satz, was Ihr Programm leistet (Funktion).

Speichern Sie Ihr Ergebnis im Testate-Ordner unter dem Namen *A5IhrName.sb*.

## D Arbeitsblatt zur Untersuchung des Verständnisses von Kreativität

Was fällt Dir ein zum Thema Kreativität und Informatik?  
Brainstorming:

Informatik

Kreativität



## E Bildungsstandards kreativ erreichen – ein Unterrichtskonzept für die Sekundarstufe I<sup>122</sup>

Mit dem neuen Schuljahr besteht die Möglichkeit, an einer neu gegründeten Schule bei der Gestaltung eines Konzeptes der informatischen Bildung von Anfang an mitzuwirken. Ziel ist es, einen Unterrichtsvorschlag für die Sek. I zu erhalten, der basierend auf neuesten informatikdidaktischen wissenschaftlichen Erkenntnissen die Mindestanforderungen der Bildungsstandards erfüllt, dabei aber weit darüber hinaus bei den Schülern Interesse und Begeisterung für die Informatik weckt. Mit dem Ziel, dass die Schüler lernen, dass die sie umgebenden Medien gestaltbar sind und die Informatik hierfür wichtige Grundlagen liefert, nennt sich der Kurs Mediengestaltung Informatik. Hier sollen die Schüler mit Hilfe neuer Medien ihre eigenen Ideen erforschen und umsetzen, grundlegende Fähig- und Fertigkeiten in der Informatik erwerben sowie Selbstvertrauen und Verantwortungsbewusstsein im Umgang mit Informations- und Kommunikationstechnologien (IKT) entwickeln.

Die Konstruktion von Softwareartefakten in einer technologieunterstützten Unterrichtsumgebung steht im Zentrum des Informatikunterrichts. Da IKT eine kreativitätsfördernde Rolle zugesprochen wird, ist die Schulinformatik in einer glücklichen Position: Computertechnologie, Software und ausgebildete Lehrer bilden einen Schwerpunkt der Unterrichtsumgebung. Zudem werden viele der Anforderungen an kreativitätsunterstützende Softwaretools von im Informatikunterricht verwendeten Softwarewerkzeugen erfüllt. Außerdem sind Grundlagen, die notwendig sind, um IKT zu verstehen und kreativ einzusetzen, Inhalte des Informatikunterrichts. Die Möglichkeiten, welche die Informatik damit bietet, um kreativ zu arbeiten, können einen starken Einfluss auf Motivation und Interesse der Schüler ausüben. Damit stützt sich kreatives Lernen im Informatikunterricht auf drei Säulen (Romeike 2007c): Die kreative Person, Informatik als Fach, welches Kreativität erfordert und fördert, sowie eine kreativitätsunterstützende Umgebung, die im Informatikunterricht durch die verfügbare Computertechnologie unterstützt wird. Eine Möglichkeit, Kreativität in der Schulinformatik zu erreichen, ist durch die Berücksichtigung der Kriterien kreativen Informatikunterrichts (Romeike 2007b). Hierdurch können sowohl hohe Lernmotivation als auch ein tiefes Verständnis erreicht werden (Romeike 2008a).

Als Ideengeber für das Konzept des Kurses Mediengestaltung Informatik dienen die „Computer Clubhouses“ (Resnick und Rusk 1996), welche das Ziel verfolgen, Schülern ein kreatives Lernen mit Computern zu ermöglichen. Quasi „nebenbei“ lernen die Schüler dort grundlegende Verfahren, Konzepte und Anwendungen der Informatik. Dem Lernansatz liegen vier Leitprinzipien zugrunde: Lernen durch Design, folgen eigener Interessen, bilden einer Gemeinschaft und eine Umgebung des Respekts und Vertrauens. Lässt sich das Konzept des freiwilligen Lernens in der Freizeit mit dem formalen Rahmen der Schule verbinden; lässt sich intrinsisch motivierte Kreativität in einem (Wahl-)Pflichtkurs realisieren? In dem geplanten Kurs soll der Spagat gewagt werden, indem versucht wird, aus dem starren lernzielgeleiteten Unterrichtsgefüge auszubrechen. Unterrichtsinhalte sollen auf den Ideen der Schüler basierend induktiv erarbeitet werden. Hierzu ist es notwendig, eine Unterrichtsatmosphäre zu schaffen, die Ideen begrüßt und anregt, insbesondere durch strikte Befolgung der „Kriterien kreativen Informatikunterrichts“ (Romeike 2007b). Darüber hinaus werden verschiedene Softwarewerkzeuge eingesetzt, um Schülerideen auszuprobieren und umzusetzen. Durch regelmäßige Präsentationen von Schülerprojekten wird eine gegenseitige Ideenankündigung zwischen den Schülern initiiert. Damit folgt jeder Unterrichtszyklus dem Challenge-Cycle (Romeike 2008f), nach welchem selbständiges Finden von persönlichen Herausforderungen, Problemmanagement, Produktverbessern und die Präsentation des Ergebnisses folgen.

<sup>122</sup> Dieser Artikel wurde veröffentlicht in (Romeike 2008b).

Im Groben sind mindestens folgende Themen in den ersten zwei Jahren geplant:

- Informatik und ich – Grundlagen und Inhalte der Informatik,
- Grafik und Bildbearbeitung, Animationen und Spiele gestalten: Ein kreativer Einstieg in die Programmierung,
- Informatik und Kunst,
- Poster, Präsentationen, Schülerzeitung: Mediengestaltung mit Standardsoftware,
- Gestaltung von Onlinemedien,
- Musikverarbeitung/Computermusik.

Parallel zum Kurs soll eine Dokumentation und wissenschaftliche Begleitung des Kurses erfolgen, in welchen die Erfahrungen ausgewertet werden.

## F Student Teachers' Beliefs about Creativity in Computer Science

Ralf Romeike, [romeike@cs.uni-potsdam.de](mailto:romeike@cs.uni-potsdam.de)

University of Potsdam, A.-Bebel-Str. 89, 14482 Potsdam, Germany

### Abstract

The study presented in this paper examines student teachers' beliefs about creativity, creative outcomes, and factors related to creativity in computer science education. Even though several studies have shown high value of creativity in computer science education, not much research has been conducted on the conceptualizations of creativity. For this study an online questionnaire was completed by student teachers of computer science. Findings indicate that student teachers highly value creativity and that they refer to creativity from a personal standpoint. The results and implications are discussed.

### Keywords

Computer science education, creativity, human factors

### INTRODUCTION

In the context of life long learning creativity seems to appear as a key factor in the discussion. In the position paper on life long learning of the International Federation on Information Processing (IFIP) the role of ICT is seen as "one of empowerment, enhancement of creativity and support" (Kendall, Samways, and Wibe, 2002). In the IFIP/Unesco ICT Curriculum for secondary schools creativity is considered in the "transforming approach" at the highest stage of school development (van Weert and Tinsley (2000)). Resnick (2007) even extends the picture of the knowledge society. He claims that knowledge alone is not enough. Rather, he suggests speaking about a Creative Society because "[i]n today's rapidly changing world, people must continually come up with creative solutions to unexpected problems. Success is based not only on what you know or how much you know, but on your ability to think and act creatively" (Resnick (2007, p.1)).

In computer science (CS) exists a subject in schools that does not only make use of ICT but which has the creative use and design of technology in the centre of its focus. Furthermore CS lessons offer a wealth of possibilities for creative engagement: CS is a highly creative subject involving creative processes; motivation and interest for the field of CS are likely to arise through creative practice; and ICT, as being central to CS, offers a fertile ground for creativity. Thus, CS is a subject that contributes to the creative society (cp. (Romeike, 2008b)).

Even though it is widely acknowledged that the facilitation of creativity is necessary, not much research has been conducted on teachers' perceptions of creativity. However, it is important that teachers recognize creative work and have a tolerant attitude for creative pupils. In the CS lesson context creativity is sometimes even seen as disturbing: "Anyhow, I am afraid that pupils let their creativity play to much so that the results of this project would be of limited usefulness." (Janneck, 2006). If such an attitude is promoted by the teacher, pupils are likely to choose familiar ways that seem safe and risk free but do not leave much space for creativity<sup>1</sup>.

The aim of this study is to examine student teachers' beliefs of creativity, of connected factors and to examine their experiences with creativity in their role as

---

<sup>1</sup> Taking risks is difficult for creative students because creativity is not always rewarded with good grades (Sternberg and Lubart, 1991). Perhaps this is due to the negative attitudes teachers hold towards creative students, which is supported by the findings of Westby and Dawson (1995).

teachers and their own development.

Student teachers are between leaving university and beginning teaching careers. They are provided with a full repertoire of teaching methods, experiences gathered during their own learning process and are full with ideas. Additionally they have already collected teaching experience during their internships and probationary teaching periods and therefore can relate theory to practice. Revealing their perceptions and understandings of creativity in the context of CS helps determine what is possible from a creative teaching perspective in CS and indicates where support needs to be provided to the teachers.

The research framework used in this study is based on a study conducted by Diakidoy and Kanari (1999). In the first part of this paper we will briefly describe a summary of relevant literature in creativity research and research in creativity in CS education. In the following sections we will outline the research method, the results and will provide a discussion of results.

This study is part of a broader research project at the University of Potsdam, where we investigate the impact of creativity in CS Education. Intermediate results show a positive correlation between the use of creative tasks and student motivation and student outcomes in secondary education CS classes.

### **Creativity**

The term creativity is used with different meanings and is discussed controversially in psychology. Correspondingly, research about creativity focuses on a variety of aspects such as identifying, accessing, assessing, and fostering creativity and the various dimensions of creativity such as creative person, process, environment, and products (refer to e.g. (Runco, 2007) for further consideration). In an educational context the question arises, what characterizes a creative student? Boden (1990) describes two aspects of creative achievements. Historical creativity (h-creativity) describes ideas that are novel and original in the sense that nobody has had them before. Something that is fundamentally novel to the individual Boden describes as psychologically creative (p-creativity). In an educational context the latter is more interesting and can be aimed for in the classroom.

### **Research in Computer Science Education and Creativity**

In (Romeike, 2007b) we pointed out three drivers of creativity in the context of CS education. In summary it was found that CS can be described as a highly creative subject involving creative processes. Furthermore, motivation and interest for the field of CS are likely to arise through creative practice. ICT, as being central to CS, offers a fertile ground for creativity. Additional support for the relevance of creative activities in the context of CS education comes from studies where researchers found that:

- Students of CS perceived software development as creative; creative phases were preferred (Gu and Tong, 2004; Romeike, 2006)
- In programming courses, allowing and encouraging students to be creative increased motivation and interest (Lewandowski et al., 2005; Long, 2007; Peppler and Kafai, 2005; Resnick, 2002; Romeike, 2008)
- Computers and programming software form fertile tools for supporting creativity (Clements, 1995; Shneiderman, 2000; Thomas et al., 2002)

In (Romeike, 2007a) we pointed out requirements that need to be considered in order to conduct creative CS lessons. These requirements include: problem management or creation of a product, relevance, subjective novelty, open tasks, inspiration, experimenting, climate of diversity, teaching more as coaching.

In the same study an analysis was performed of how much creativity was already involved in German CS lesson examples. It was found that many lesson outlines would allow for creativity but in the descriptions of the lesson outlines themselves

creativity was rarely regarded. We assume this oversight is due to a lack of awareness of the potential creativity offers or a problematic perception of creativity itself. Even attempts to suppress creative efforts by students were found.

In a study about pathways leading to students' enrollment in university level CS studies, Knobelsdorf and Romeike (2008) found creativity to be a pathway and reason why students pursue CS at university level. In the same context it was found in the students' biographies that creativity was occurring in general in their leisure time only. CS in high school often did not allow for creativity and thus was perceived negatively.

In order to manifest creativity in the classroom, not only do new tools and guidelines need to be developed, but also the attitudes and beliefs of the teachers need to be taken into consideration.

### **Research about teachers' perceptions of creativity**

Most of the research in creativity has been conducted in regard to creative outcomes, assessment of creativity (e.g. creativity tests) and creative processes. However, there have been a few studies focusing on the subjective views of teachers in regard to creativity (see (Fryer and Collings, 1991) and (Diakidoy and Kanari, 1999) for a summary of the research). The results of these studies are ambiguous, reflecting a variety of different backgrounds. These studies have in common that creativity was seen as desirable and in connection with personal characteristics which can be fostered in the classroom. Also, teachers described an understanding of creativity that more fits the definition of p-creativity.

In a more recent study Fryer (2006) investigated the views of 94 teachers in higher education. Creativity was regarded as important and as a key to success. Also, creativity was regarded as an ability that could be developed rather than seen as a rare gift. Statements about how creativity can be developed and fostered were in general in line with the existing literature in creativity research.

## **METHOD**

### **Instrument**

The perceptions and views of the student teachers' together with their perspectives on creativity in relation to CS were assessed with an online questionnaire that consisted of 5 parts. The first two parts were based on a questionnaire developed by Diakidoy and Kanari (1999). Part one consisted of 7 questions, assessing the general understanding of creativity with open-ended questions and the perceived impact of related factors such as intelligence. Part two consisted of 20 questions assessing circumstances about how creativity can be fostered and assessed. In part 3 to 5 of the questionnaire the student teachers were asked to reflect on creativity in the context of CS. Part 3 consisted of 14 questions concerning the role of creativity in the field, part 4 consisted of 3 questions about the impact of creativity in ones own motivation, and part 5 consisted of 13 questions concerning ideas and experiences with creativity in CS lessons.

The student teachers had to answer three types of questions: open-ended questions that required pointing out and explaining their own perspectives of creativity and creativity with respect to CS, statements allowing for agreement or disagreement, multiple choice questions, and questions that required the indication of agreement on a 6-point scale from full-agreement to full disagreement. For the analysis of the responses the interrogation tool Graftstadt was used. The content of open-ended questions was analyzed for distinct categories described in the texts.

### **Sample**

The questionnaire was completed by 13 computer science education students in total (10 male, 3 female). Five students were enrolled at the University of Potsdam, had completed all relevant courses in CS education and had finished their school

practicum. Eight student teachers were involved in the second phase of German teachers' education<sup>2</sup> conducting their probationary period at schools in Berlin.

## RESULTS

### Definitions of creativity and creative Outcomes

The majority of student teachers see creativity closely related to the capability of coming up with, developing, and/or realizing individual ideas (53.8%). In the explanatory part of this question the student teachers often refer to creative outcomes and/or products in a broad sense (46.2%). A creative process by the majority is described as taking personally novel paths in order to achieve a personal goal. This view is comparable with the description of p-creativity by Boden (1990), where creativity is seen in a personal context. This view is affirmed in a question about requirements for creative outcomes: 46.2% of the student teachers state that creative outcomes are novel for the person, 46.2% even state that creative outcomes are not necessarily novel. Just one student teacher answers with the h-creativity perspective. He states that it is important for a creative outcome to be novel to the person and the society. However, this would be hard or impossible to achieve in school. Only one of the student teachers mentioned in his definitions usefulness and appropriateness as mandatory characteristics for creative outcomes. Furthermore 81.8% consider it as creative if a child discovers a new way to add three-digit numbers but the strategy does not lead to the correct solution. In addition, solving of problems in special ways, experimenting, designing and independence were mentioned when describing creativity.

Creativity is:

- to have lots of ideas, exceptional ideas. It includes communicating and realizing these ideas.
- having freedom in approaching a problem
- to develop own ideas and problems in the context of an existing rule system and/or material
- seeing content in new contexts

Table 1: Examples of definitions of creativity

Imagination	84,6%
Ability to set own goals	84,6%
Autonomy	61,5%
Independence	61,5%
Self-confidence	61,5%
Critical thinking ability	61,5%
Analogical reasoning ability	61,5%
Problem finding ability	61,5%
Many interests	53,8%
Ability to set own rules	46,2%
Intelligence	38,5%
Divergent thinking ability	38,5%
Need for praise and reinforcement	30,8%
Need for recognition and acceptance	23,1%
Willingness to accept guidance	15,4%
Obedience to rules and expectations	15,4%
Innate talent	15,4%
Artistic tendencies	15,4%
Convergent thinking ability	15,4%
Need to avoid mistakes	15,4%
Fear of failure	7,7%

Table 2: Personality characteristics and abilities identified by the sample as necessary for creativity

### Individual Differences in creativity and frequency of its appearance

61.6% of the student teachers agree that creativity is a characteristic of all people. However, 91.7% believe that some children are more creative than others and pupils "manifest their creativity in a variety of domains and in a variety of ways"

<sup>2</sup> In the German system teachers' education consists of two parts: the first phase happens at a university, includes several practical trainings and is finished with a first state examination. It is succeeded by a second phase, referred to as practice time, where students teach part time in schools but in addition attend a study seminar.

(91.7%). These results suggest that creativity is seen as a general character trait. However, creativity is seen as appearing in different forms and extends.

54.5% of the student teachers describe intelligence as directly related to creativity including that intelligence is a necessary requirement or beneficial for creativity. In contrast, the remaining 45.5% do not see any relation. 53.8% think it is possible for a very intelligent person not to be creative and 84.6% disagree with the statement that good pupils are more likely to be creative than average pupils. This statement stands in contrast to the indicated correlation of intelligence and creativity. However, achievement in school is often determined by adopting, memorizing and applying familiar approaches. These are characteristics that are atypical of creative persons. In contrast to intelligence, imagination (84.6%) and the ability to set individual goals (84.6%) are mentioned most as a prerequisite for creativity (refer to table 2 for more characteristics).

92.3% believe that they will encounter creative children often or very often. This includes a broad perception of creativity, as 75% of student teachers consider it creative when a child discovers on his/her own how to use a paper clip.

### Factors influencing the manifestation of creativity

The factors that can influence creativity in learning settings were assessed in questions concerning the influence of personality, domain and environment.

**Personality.** It can be seen in table 2 that the majority of student teachers believe that creativity depends on personality characteristics such as imagination, goal setting abilities and autonomy. Asked directly, student teachers consider creativity being dependent on ones personality (100%), the environment (92.3%), and a task (76.9%). Interestingly, artistic tendencies, innate talent and the need for recognition and acceptance were rarely seen as prerequisites for creativity.

**Domain.** Less than half believe that the manifestation of creativity depends on the domain, and all consider it as possible that someone could even manifest creativity in the domain of history<sup>3</sup>. Interestingly, here all given examples do reflect real creative processes in the domain of history, not artistic "extensions" like drawing maps or illustrations.

These examples parallel responses to the question: in which subjects or domains the student teachers can imagine someone to show creativity. A variety of domains were mentioned and 46.2% of the student teachers neglected the question and just stated "in all". It is apparent that creativity is seen to be possible in

Emphasis on discovery learning	76.9%
Opportunities to correct own mistakes	76.9%
Emphasis on intrinsic motivation	61,5%
Opportunities to question theories and assumptions	61,5%
Emphasis on autonomy and independence	53,8%
Frequent and detail feedback	46,2%
Acceptance of all work outcomes	46,2%
Frequent praise	38,5%
Choice in assignments	30,8%
Emphasis on knowledge acquisition	23,1%
Use of external rewards	23,1%
Frequent evaluation of outcomes	15,4%
Emphasis on competition	15,4%
Emphasis on collaborative learning	15,4%
Emphasis on following instructions	0,0%

Table 3: Environmental aspects identified by the sample as likely to facilitate creativity

<sup>3</sup> History was taken as an example Domain by Diakidoy and Kanari illustrating the connection between how someone conceptualizes the nature and process of a domain in relation to how someone conceptualizes creativity in the domain. E.g. for historians creativity is needed in the domain, e.g. in comprehension of historical events. For high achieving pupils history was understood as uncreative because historical texts were understood as a report of facts and interrelated events that could not be changed.

every domain. This could be an interpretation of the teachers who are required to teach two subjects and thus should foster creativity where ever possible.

**Environment.** Environmental factors were seen as crucial to creativity among almost all student teachers. While personality factors are only accessible to a certain extent by the teacher, the in class environment is more under their control. Table 3 shows the environmental aspects that were considered to facilitate creativity.

### **Facilitation of creativity**

All student teachers indicate that creativity can be fostered by the teacher. Also, all indicate that it can be fostered in everybody. In table 4 we present the most frequently mentioned suggestions of how the student teachers think creativity can be fostered.

Many of the answers comply with the criteria for creative CS lessons, where certain requirements for tasks and the environment are proposed in order to achieve creative CS lessons.

However, less than half (45.5%) of the student teachers consider school to be a place where pupils have the possibility to show creativity. This stands in contrast to statements stating that the CS curriculum allows creativity (100% agreement). The student teachers almost unanimously explain this by saying the curriculum gives just a frame for what needs to be taught. How the lessons are conducted is entirely up to the teacher and hence can take creativity into account. Nevertheless, environmental factors in school settings such as time restrictions, a high work load and grading are seen as inhibiting creativity.

61.6% believe that creative thinking is different from the thinking required to solve problems in school. They elaborate that in school the pupils are often encouraged to apply methods or "mechanism" which have been demonstrated by the teacher and come up with the same results as the teacher. To often there is just a "right" or "wrong". This is obviously problematic from a creative perspective.

Creativity can be fostered through:

- leaving space for experimentation
- accepting different approaches and solutions
- Assigning flexible tasks
- demonstrating that for one problem several approaches and solutions are possible
- rewarding creative approaches
- leaving enough time

Table 4: Examples of suggestions of how to facilitate creativity given by the sample

### **Creativity and Computer Science**

All student teachers agree that creativity and CS go well together. 84.6% agree that creativity is needed in CS in several areas, 46.2% even marking the highest agreement. They emphasize that for a task several solutions are possible, finding analogies and exploration are involved in CS related activities and that CS products are generally the results of creative processes. For example, the design of algorithms is considered as creative by 84.6% of participants. Thus 76.9% indicate that in CS one can be at least as creative as in art or music. As for which one of the fields leaves more room for creative achievements, the answers vary throughout the scale.

Activities in CS that require creativity were mentioned in the free text question as follows: problem solving (46.2%), designing user interfaces/GUIs (38.5%), programming (30.8%), algorithm design (30.8%), problem finding (15.4%), testing (15.4%), debugging, presenting and accommodating theoretical concepts (each 7.7%). These statements are illustrated through mentioning experiences from their own lessons. Success in winning the pupils' interest by assigning creative tasks and/or leaving room for creativity was demonstrated. However, some student teachers also state that they seldom witnessed creative lessons when observing

lessons by their mentor. These statements are supported by answers to the question concerning experiences with creative CS lessons: *“Creative CS lessons are possible, but rarely performed by teachers. In my own experiences, pupils often are surprised by creative approaches at first, but are willing to follow and often come up with great results which they would not have achieved with regular methods.”*

### **Creativity and computer science lessons**

All student teachers agree that computer science lessons can be creative. All student teachers also agree, though less strongly, that CS lessons should be creative. In comparison with other school subjects the student teachers' opinions differ: 53.9% see similar creative potential in CS as in other subjects. 15.3% of student teachers see more creative potential in CS than in other subjects. 15.3% see less potential. 15.3% sees slightly more potential. In the student teachers' opinion, creative CS lessons are characterized by open tasks and problems, a variety of different results, encouragement of pupils' ideas and orientation on pupils' motivation and interest. These and other characteristics mentioned parallel the criteria for creative CS lessons as established and successfully applied in (Romeike, 2008). The student teachers report a variety of ways pupils demonstrated creativity in their lessons:

- coming up with unique approaches for programming tasks
- altering programming tasks in order to apply individual ideas
- designing web pages and GUIs
- including knowledge gained from other fields and drawing analogies, e.g. to social consequences
- independently extending software
- asking interesting and challenging questions

As creativity research suggests, ICT can provide a fertile environment for creative activities. Thus, CS lessons that generally take place in computer labs could take advantage of the technology present in a creative way. 92.3% of the student teachers see working in a computer lab as having a beneficial influence on pupils' creativity. However, the support is quite weak. On a scale from 1 to 6, with 1 being the strongest agreement, the answer average was 2.77. Perhaps the student teachers perceived computers in lessons as distracting to pupils.

As for their own teaching activity, 53.9% report they sometimes considered creativity in the preparation of their lessons. 38.4% did so often and 7.7% regularly considered creativity in their lesson planning. One teacher concludes: *“I believe that with creative approaches I can reach pupils that would not be interested in CS or would otherwise give up CS.”*

### **Own experiences with creativity in CS**

The way of how a teachers' individual experience with CS is perceived can have a great influence on how this subject is taught in school. For 61.5% creativity was relevant for their initial interest in CS. In contrast, studying CS at the university level was not perceived as creative by 83.3% of student teachers. 76.9% indicate that creativity was not demanded in university studies. In response to the question of whether or not they showed creativity, 53.8% answered with an agreement. However, it was indicated that creativity was not often rewarded and the primary goal of the studies was to meet the course objectives. 83.3% describe their recent activities in CS as creative. However, in explanation the majority described activities related to teaching only (e.g. lesson preparation, task creation). In their leisure time computing no longer plays a dominant role.

## DISCUSSION

The findings of this study indicate that student teachers of CS highly value creativity. Creativity is primarily conceptualized as a general ability that appears in different contexts and in different extends. Furthermore, creativity is seen as a method to engage pupils in learning and sparking their interest for the field of CS. In many extents these results parallel findings of creativity research. This includes characteristics of how creativity can be identified, how creativity can be fostered in pupils, and environmental variables that promote creativity. However, in some aspects the perceptions differ. Most of the student teachers judge creativity from a strong personal perspective. We suppose this is coming from teachers' requisites to evaluate pupils in regard to their personal development. In order for manifesting creativity this is a crucial and beneficial viewpoint. The way the student teachers identified creativity in school settings reveals a perspective that may be transferred from the field of CS. While a broad body of research is concerning characteristics of creative outcomes, the student teachers basically emphasized creative processes. It is typical for teaching CS to emphasize problem solving processes. Furthermore it is a creative approach that leads to a unique solution. Keeping the encouragement of creative processes in mind in CS lessons this way will also ensure creative outcomes.

The way of how one conceptualizes creativity in a domain will depend on how one conceptualizes the nature of a domain. Interesting in this context are student teachers' perspectives on creativity in different domains. In a common understanding, CS is not generally seen as a creative field. However, the student teachers do perceive CS as a creative field and thus seem to admit the same to other fields. Repeatedly they stated: "In any field it is possible to show creativity."

The understanding of CS as creative is a necessity for applying creativity in CS lessons. The dimensions of the proposed model for creativity in CS education are all reflected in the answers, examples and explanations as provided by the student teachers. Also, related to their personal biographies student teachers report that creativity was a reason why they initially got interested in CS. This may be a reason why they now try to apply creativity as a method to engage pupils. For a teacher who did not experience creative computing such a transfer will be impossible to make. Especially when keeping in mind the creativity obstructive impressions the student teachers have of their university studies, educators at universities may need to correct the image of the subject they get across.

Student teachers' perceptions of how creativity can be fostered in CS are widely in line with research in this context. None of the student teachers perceives it as his task to identify and foster only exceptionally creative pupils. Based on an understanding that creativity is an ability that can be fostered in every pupil, teachers' interventions that apply to all pupils were pointed out. However, the methods named for fostering creativity are not well sophisticated. The student teachers are uncertain about how they strategically use creativity and which form and benefit for the subject it can have. In order to better manifest creativity didactical support is expedient.

Within the answers we could find a variety of examples for creative pupil performance and creative outcomes. All of these examples come along with an attributed learning success. Unfortunately, the student teachers also point out that generally creativity is a rare phenomenon to encounter in lessons and they rarely see creativity in lessons by their mentor. Obviously the conditions in school do not really allow for creativity. We consider it as important that student teachers do not stop seeing creativity as a favorable incident to include in their lessons by accepting the limiting conditions of the school environment. Rather it is indicated to alter the teaching settings in a way that creativity is encouraged. To a certain extent this should be possible for every dedicated teacher.

The understanding of creativity the student teachers express may be arguable and perhaps is marginal to meeting the definition of creativity. Creative achievements, as described by the student teachers, partly would be better called "productive" due to a lack of novelty. However, this pragmatic understanding of creativity provides a low entrance barrier and motivation for the pupils to approach open and creative tasks and for losing fear of entering novel and individual approaches in school and elsewhere.

Since the perspectives on creativity of the student teachers are very broad and include all kinds of subjects, it seems to be advised to explicitly include creativity techniques in the school curriculum. This way a multidisciplinary manifestation of creativity would be made possible. Furthermore, all students would be better prepared for the demands of a Creative Society and life long learning.

## **CONCLUSION**

In contrast to prevalent stereotypes attributing computer science as non-creative the student teachers who participated in this study report about creative potential of CS which they have experienced and what they are willing to address in their CS lessons. Creativity by the participants is defined in a way that encourages the facilitation of creativity. In fact, CS was seen as a subject that can greatly contribute to the preparation of the "Net-Generation" for the creative society.

## Literaturverzeichnis

- AAUW, American Association of University Women (2000): Tech-Savvy: Educating Girls in the New Computer Age. American Association of University Women, Washington, DC.
- Amabile, T. M. (1996): Creativity in context: update to the social psychology of creativity. Westview Press, Boulder, Colorado [u.a.].
- Amabile, T. M. (1998): How to kill creativity. Harvard Business Review, 76 (5), 76-87.
- Anderson, J. und Weert, T. v. (2002): UNESCO/IFIP Curriculum - Information and Communication Technology in Secondary Education  
<http://www.edu.ge.ch/cptic/prospective/projets/unesco/en/curriculum2000.pdf>, (8.8.2008).
- Anderson, L. W. und Krathwohl, D. R. (2001): A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. Longman, New York [u.a.].
- Antonitsch, P., Kramer, L., Lerchster, R. und Ukowitz, M. (2007): IT-Frust statt Lust? Zur Studienwahl von Jugendlichen aus Sicht von Schülerinnen, Eltern, Lehrenden und Praktikern. In: Information und Gesellschaft: Technologien einer sozialen Beziehung, Greif, H., Mitrea, O. und Werner, M. (Hrsg.), VS Verlag für Sozialwissenschaften.
- Aupperle, M. (2002): Die fortgeschrittene Kunst der Programmierung mit C++. Vieweg und Teubner, Wiesbaden.
- Ausubel, D. P. (1974): Psychologie des Unterrichts. Beltz, Weinheim.
- Ausubel, D. P., Novak, J. und Hanesian, H. (1982): Psychologie des Unterrichts Beltz, Weinheim.
- Barker, L. J., Garvin-Doxas, K. und Roberts, E. (2005): What can computer science learn from a fine arts approach to teaching? SIGCSE Bull., 37 (1), 421-425.
- Barrett, F. J. (1998): Coda: Creativity and Improvisation in Jazz and Organizations: Implications for Organizational Learning. Organization Science, 9 (5), 605-622.
- Bauer, W. F., Juncosa, M. L. und Perlis, A. J. (1959): ACM Publication Policies and Plans. Journal of the ACM (JACM), 6 (2), 121-122.
- Begel (1996): LogoBlocks: A Graphical Programming Language for Interacting with the World. Unpublished Advanced Undergraduate Project, MIT Media Lab.
- Bergin, S. und Reilly, R. (2005): The Influence of Motivation and Comfort-Level on Learning to Program. In: Proceedings of the PPIG 17, University of Sussex, Brighton UK, In P. Romero, J. G., E. Acosta Chaparro & S. Bryant (Hrsg.), 293-304.

- Berliner, P. F. (1994): Thinking in jazz: the infinite art of improvisation. University of Chicago Press, Chicago.
- Bertelsmann (2004): Bertelsmann Wörterbuch der deutschen Sprache. Wissen Media Verlag, Gütersloh/München.
- Biggs, J. (1996): Enhancing Teaching through Constructive Alignment. Higher Education, 32 (3), 347-64.
- Blackwell, A. und Collins, N. (2005): The programming language as a musical instrument. In: Proceedings of the PPIG05 (Psychology of Programming Interest Group), University of Sussex, Brighton UK., 120-130.
- Bloom, B. S. (1974): Taxonomie von Lernzielen im kognitiven Bereich. Beltz, Weinheim.
- Blum, B. I. (1991): Torn between Fun and Tedium. In: Software Creativity, Glass, R. L. (Hrsg.), Prentice Hall PTR, Englewood Cliffs, NJ.
- Boden, M. A. (1990): The creative mind: myths & mechanisms. Basic Books, London.
- Boden, M. A. (1995): Die Flügel des Geistes: Kreativität und künstliche Intelligenz. Dt. Taschenbuch-Verl., München.
- Bond, G. W. (2005): Software as art. Communications of the ACM, 48 (8), 118-124.
- Bortz, J. und Döring, N. (2002): Forschungsmethoden und Evaluation für Human-und Sozialwissenschaftler. (3. überarbeitete Auflage). Springer, Berlin u. a.
- Bringsjord, S. und Ferrucci, D. A. (1999): Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus, a Storytelling Machine. Lawrence Erlbaum Associates.
- Brooks, F. P. (1995): The mythical man-month. Addison-Wesley Longman Publishing Co., Inc.
- Bruce, C., Buckingham, L., Hynd, J., McMahon, C., Roggenkamp, M. und Stoodley, I. (2004): Ways of experiencing the act of learning to program: A phenomenographic study of introductory programming students at university. Journal of Information Technology Education, 3, 143-160.
- Bruder, R. (2001): Kreativ sein wollen, dürfen und können. Kreativität, mathematik lehren, 106, 46-50.
- Brunner, R. B. (1991): Creative Computer Science Teaching Techniques. Computer Science Education, 2 (3), 187-195.
- Cambridge (2008): Cambridge Dictionary of American English Online. <http://dictionary.cambridge.org/Default.asp?dict=A>, (14 Dec 2007).

- Chaytor, L. und Leung, S. (2003): How to creatively communicate Microsoft.NET technologies in the IT curriculum. In: Proceedings of the 4th conference on Information technology curriculum, ACM Press, Lafayette, Indiana, USA, 168-173.
- Clements, D. H. (1995a): Playing with computers, playing with ideas. Educational Psychology Review, V7 (2), 203-207.
- Clements, D. H. (1995b): Teaching Creativity with Computers. Educational Psychology Review, 7 (2), 141-161.
- Cohen, H. (1995): The further exploits of Aaron, painter. Stanford Hum. Rev., 4 (2), 141-158.
- Cohen, H. (1999a): Colouring Without Seeing: a Problem in Machine Creativity. In: "Department of Visual Arts". University of California, San Diego.
- Cohen, H. (1999b): A self-defining game for one player. In: Proceedings of the 3rd conference on creativity & cognition, Loughborough, United Kingdom, ACM.
- Collins, M. A. und Amabile, T. M. (1999): Motivation and Creativity. In: Handbook of Creativity, Sternberg, R. J. (Hrsg.), Cambridge University Press, Cambridge, 297-312.
- Computer Science and Telecommunications Board, N. R. C. (1999): Being Fluent with Information Technology. National Academic Press, Washington, D.C.
- Conti, R. und Amabile, T. (1999): Motivation/Drive. In: Encyclopedia of Creativity, Runco, M. A. und Pritzker, S. R. (Hrsg.) 2, Academic Press, San Diego, Calif. [u.a.] 251-260.
- Conti, R., Coon, H. und Amabile, T. M. (1996): Evidence to Support the Componential Model of Creativity: Secondary Analyses of Three Studies. Creativity Research Journal, 9 (4), 385-389.
- Couger, J. D. (1990): Ensuring Creative Approaches in Information System Design. Managerial and Decision Economics, 11 (5), 281-295.
- Craft, A. (2001): An analysis of research and literature on creativity in education. Report prepared for the qualification and curriculum authority.  
[http://www.ncaction.org.uk/creativity/creativity\\_report.pdf](http://www.ncaction.org.uk/creativity/creativity_report.pdf) (10.06.2008).
- Cropley, A. J. (2000): Defining and Measuring Creativity: Are Creativity Tests Worth Using? Roeper Review, 23 (2), 72-79.
- Csikszentmihalyi, M. (1990): Flow : The psychology of optimal experience. Harper [and] Row, New York u.a.
- Csikszentmihalyi, M. (1997): Kreativität: wie Sie das Unmögliche schaffen und Ihre Grenzen überwinden. Klett-Cotta, Stuttgart.

- Curtis, B., Guindon, R., Krasner, H., Walz, D., Elam, J. und Iscoe, N. (1987): Empirical Studies of the Design Process. In: Proceedings of the Second Workshop on Empirical Studies of Programmers.
- Curzon, P. und Rix, J. (1998): Why do Students take Programming Modules? In: Proceedings of the 6th annual conference on the teaching and computing and the 3rd annual conference on integrating technology into CSE: Changing the delivery of Computer Science Education. ITICSE '98, Dublin, Ireland, 59-63.
- Cusumano, M. A. (1991): Japan's software factories: a challenge to U.S. management. Oxford University Press, Inc.
- Davis, G. A. (1991): Teaching creativity thinking. In: Handbook of gifted education, Colangelo & N. und Davis, G. A. (Hrsg.) 236-244, Allyn & Bacon, Boston:.
- Davis, T. A. und Kundert-Gibbs, J. (2006): The role of computer science in digital production arts. In: "Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education". ACM, Bologna, Italy.
- De Bono, E. (1970): Lateral Thinking: Creativity Step by Step. Harper & Row.
- Deci, E. L. und Ryan, R. M. (1985): Intrinsic motivation and self-determination in human behavior. Springer, New York.
- Desel, J. (2001): Das ist Informatik. Springer, Berlin [u.a.].
- Diakidoy, I.-A. N. und Kanari, E. (1999): Student Teachers' Beliefs about Creativity. British Educational Research Journal, 25 (2), 225-43.
- Diethelm, I. (2007): Strictly models and objects first - Unterrichtskonzept und -methodik für objektorientierte Modellierung im Informatikunterricht. Universität Kassel, Kassel.
- Dreifus, C. (2002): Getting it about John Maeda. [http://www.maedastudio.com/catalog\\_lores.pdf](http://www.maedastudio.com/catalog_lores.pdf), (10.06.2008).
- Duraes, J. A. und Madeira, H. S. (2006): Emulation of Software Faults: A Field Data Study and a Practical Approach. IEEE Trans. Softw. Eng., 32 (11), 849-867.
- Eastman, E. G. (2003 ): Fact-based problem identification precedes problem solving J. Comput. Small Coll. , 19 (2) , 18-29
- Edelmann, W. (1996): Lernpsychologie. Beltz, Weinheim.
- Eich-Soellner, E. (2005): Informationen zum Fachbereich Informatik, Mathematik an der Fachhochschule München (FHM). <http://www.cs.fhm.edu/broschuere/> (2.9.2006).
- Engelmann, L. (2004): Informatische Grundbildung. Paetec, Altenburg.

- Epstein, R. G. (2006): An ethics and security course for students in computer science and information technology. In: Proceedings of the 37th SIGCSE technical symposium on Computer science education, ACM Press, Houston, Texas, USA, 535-537.
- ETH-Zürich (2003): Interview with Prof. Gustavo Alonso.  
[http://www.inf.ethz.ch/news/focus/edu\\_focus/alonso](http://www.inf.ethz.ch/news/focus/edu_focus/alonso) (2.9.2006).
- Fasko, D. (2000): Education and creativity. *Creativity Research Journal*, 13 (3-4), 317-327.
- Feldgen, M. und Clua, O. (2003): New motivations are required for freshman introductory programming. In: Proceedings of the 33rd ASSE/IEEE Frontiers in Education Conference, Boulder, Vol. 1, T3C-T24.
- Feldhusen, J. F. und Goh, B. E. (1995): Assessing and Accessing Creativity - an Integrative Review of Theory, Research, and Development. *Creativity Research Journal*, 8 (3), 231-247.
- Feldhusen, J. F. und Treffinger, D. J. (1985): Creative thinking and problem solving in gifted education. Kendall-Hunt, Dubuque, Iowa.
- Feldman, D. H. (1988): Creativity: Dreams, insights, and transformations. In: The nature of creativity, Sternberg, R. J. (Hrsg.), Cambridge University Press, New York.
- Feller, G. (1996): Defizite der Ausbildung - Erfordernisse der Weiterbildung? Ergebnisse und Betrachtungen an der Schnittstelle Lehrabschluß aus der Sicht von Auszubildenden. In: Berufliche Aus- und Weiterbildung., Diepold, P. (Hrsg.), Nürnberg, 251-260.
- Flaherty, A. W. (2005): Frontotemporal and dopaminergic control of idea generation and creative drive. *The Journal of Comparative Neurology*, 493 (1), 147-153.
- fm4.orf.at (2004): Nerd's not dead. Internetdiskussion zu Nerd, Kreativität und Informatik.  
<http://fm4.orf.at/janis/187570/main>, (24.4.2008).
- Fothe, M. (2002): Problemlösen mit Python. ThILLM, Bad Berka.
- Frenkel, K. A. (1987): Profiles in computing: Brian K. Reid: a graphics tale of a hacker tracker. *Commun. ACM*, 30 (10), 820-823.
- Friedrich, S. (1995): Grundpositionen eines Schulfaches. Ergebnisse des 2. Fachdidaktischen Gesprächs zur Informatik an der Technischen Universität Dresden. *LOG IN*, 15 (5/6).
- Fryer, M. (2006): Facilitating Creativity in Higher Education: The Views of National Teaching Fellows. Higher Education Academy.
- Fryer, M. und Collings, J. A. (1991): Teachers' views about creativity. *British Journal of Educational Psychology*, 61 (2), 207-219.

- Galton, F. (1869): Hereditary Genius. An Inquiry into Its Laws and Consequences. Macmillan & Co, London and New York.
- Garner, S. R. (2005): Hacking with the Divine: A Metaphor for Theology-Technology Engagement. *Colloquium*, 37 (2), 181-195.
- Gärtner, H.-J. (1997): Kreativität im Chemieunterricht. *Naturwissenschaften im Unterricht. Chemie*, 8 (42), 12-16.
- Geser, H. (1999): Mängel der Schulausbildung aus Arbeitgebersicht. In: *Wandel der Arbeitswelt*, Online Publications, Zürich, <http://geser.net/work/geser/05.pdf> (2.9.2006).
- GI (1993): GI-Empfehlungen für das Fach Informatik in der Sekundarstufe II allgemeinbildender Schulen. Beilage zu LOG IN 13 (1993), Nr. 3.
- GI (2006): Was ist Informatik? Gesellschaft für Informatik e.V. (GI), Bonn.
- Ginat, D. (2001): Misleading intuition in algorithmic problem solving. In: *Proceedings of the 32nd SIGCSE technical symposium on Computer Science Education*, Charlotte, North Carolina, United States, ACM Press, 21-25.
- Glass, R. L. (1995): *Software creativity*. Prentice Hall PTR, Englewood Cliffs, NJ.
- Glass, R. L. (2006): *Software creativity 2.0. developer .\** Books, Atlanta.
- Goorhuis, H. (1994): *Konstruktivistische Modellbildung in der Informatik*.
- Graham, P. (2004): *Hackers and Painters: Big Ideas from the Computer Age*. O'Reilly Associates Inc.
- Gropius, W. (1918): Manifest. <http://www.bauhaus.de/bauhaus1919/manifest1919.htm>, (26.5.2008).
- Gu, M. und Tong, X. (2004): Towards Hypotheses on Creativity in Software Development. *Lecture Notes in Computer Science*, 3009, 47-61.
- Guilford, J. P. (1950): Creativity. *American Psychologist*, 5, 444-454.
- Guindon, R. und Curtis, B. (1988): Control of cognitive processes during software design: what tools are needed? In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, Washington, D.C., United States, ACM Press, 263-268.
- Guzdial, M. und Soloway, E. (2002): Teaching the Nintendo generation to program. *Commun. ACM*, 45 (4), 17-21.

- Häcker, H. und Stapf, K. H. (1998): Dorsch Psychologisches Wörterbuch (13. überarbeitete und erweiterte Auflage). Huber, Bern.
- Hadamard, J. S. (1949): The psychology of invention in the mathematical field. Princeton, NJ: Princeton University Press.
- Haladyna, T. M. (1997): Writing Test Items To Evaluate Higher Order Thinking. Allyn & Bacon, Boston.
- Hartmann, W. (2004): Computer und Internet im Unterricht: Zu hohe Erwartungen! Gymnasium Helveticum 1/04, 1 (44-45).
- Hascher, T. (2005): Emotions in Everyday School-Life: Effects and forms of regulation (In German). Zeitschrift für Pädagogik, 51 (5), 610-625.
- Hengl, H.-T. (2008): Informatiker müssen Bandbreite ihres Fachs besser darstellen. Computer Zeitung, 39 (32-33), 21.
- Hennessey, B. A. (2007): Creativity and Motivation in the Classroom: A Social Psychological and Multi-Cultural Perspective. In: Creativity: A Handbook for Teachers, Tan, A.-G. (Hrsg.), World Scientific, Singapore.
- Hennessey, B. A. und Amabile, T. M. (1987): Creativity and Learning: What Research Says to the Teacher. National Education Association, Washington D.C.
- Hentig, H. v. (2000): Kreativität : hohe Erwartungen an einen schwachen Begriff. Beltz, Weinheim [u.a.].
- Hill, A. M. (1998): Problem solving in real-life contexts: An alternative for design in technology education. International Journal of Technology and Design Education, 5 (3), 1-18.
- Hofstadter, D. R. (1996): Die FARGonauten. Über Analogie und Kreativität. Stuttgart: Klett-Cotta.
- Hopkins, S. (1995): Listen. The Economist.
- Hubwieser, P. (2000): Didaktik der Informatik : Grundlagen, Konzepte, Beispiele. Springer, Berlin [u.a.].
- Humbert, L. (2003): Zur wissenschaftlichen Fundierung der Schulinformatik. Pad-Verl., Witten.
- Hunt, A. und Thomas, D. (2001): The Art in Computer Programming. The Pragmatic Programmers, LLC. September.
- Isaksen, S. G. und Treffinger, D. J. (1985): Creative problem solving: the basic course. New York: Bearly Limited.

- Jablko, L. (1994): Zur Entwicklung von Kreativität im Physikunterricht. In: Zur Didaktik der Physik und Chemie; Probleme und Perspektiven, Behrendt, H. (Hrsg.) 15, Leuchtturm-Verlag Alsbach, 328-331.
- James, V., Lederman Gerard, R. und Vagt-Traore, B. (2004): Enhancing creativity in the classroom. <http://projects.coe.uga.edu/epltt/>, (02.06.2008).
- Janneck, M. (2006): Partizipative Systementwicklung im Informatikunterricht. LOG IN, (138/139), 60-66.
- Kaasbøll, J. J. (1998): Teaching critical thinking and problem defining skills. *Education and Information Technologies*, 3 (2), 101-117.
- Kay, A. und Goldberg, A. (1977): Personal Dynamic Media. *Computer*, 10 (3), 31-41.
- Kendall, M., Samways, B. und Wibe, J. (2002): Position Paper Lifelong Learning (LLL) Version 1. International Federation for Information Processing (IFIP), Laxenburg.
- Kidder, T. (2000): *The Soul Of A New Machine*. Back Bay Books.
- Kind, P. M. und Kind, V. (2007): Creativity in Science Education: Perspectives and Challenges for Developing School Science. *Studies in Science Education* 43, 1-37.
- Klütsch, C. (2007): *Computergrafik: Ästhetische Experimente zwischen zwei Kulturen. Die Anfänge der Computerkunst in den 1960er Jahren*. Springer, Wien, New York.
- Knobelsdorf, M. und Romeike, R. (2008): Creativity as a Pathway to Computer Science. In: *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education (ITICSE 2008)*, Madrid, ACM Press.
- Knobelsdorf, M. und Schulte, C. (2005): Computer Biographies - A Biographical Research Perspective on Computer Usage and Attitudes Toward Informatics. In: *Proceedings of the 5th Annual Finnish / Baltic Sea Conference on Computer Science Education*, Koli.
- Knobelsdorf, M. und Schulte, C. (2007): Computer Science in Context - Pathways to Computer Science. In: *Proceedings of the 7th Baltic Sea Conference on Computing Education Research ( Koli Calling 2007)*, Koli National Park, Finnland, Lister, R. und Simon (Hrsg.), *Conferences in Research and Practice in Information Technology*, Vol. 88.
- Knuth, D. E. (1974): Computer Programming as an Art. *Communications of the ACM*, 17 (12), 667 - 673.
- Knuth, D. E. (2001): *Things a computer scientist rarely talks about*. CSLI Publications, Stanford, CA.
- Koberg, D. und Bagnall, J. (1981): *The All New Universal Traveler: A Soft-Systems Guide to Creativity, Problem-Solving, and the Process of Reaching Goals*. William Kaufmann Inc.

- Kohl, L. und Romeike, R. (2006): Aktueller Stand der Objektorientierung bei Informatiklehrerinnen und -lehrern. In: Proceedings of the HDI2006 - 2. GI-Fachtagung "Hochschuldidaktik der Informatik", München, Forbrig, P., Siegel, G. und Schneider, M. (Hrsg.), LNI, Vol. P-100, 63-75.
- Krämer, S. (2007): Chemie, die begeistert. [http://www.partner-fuer-schule.nrw.de/wirwollen/download/fl\\_jundykraemer.pdf](http://www.partner-fuer-schule.nrw.de/wirwollen/download/fl_jundykraemer.pdf), (25.3.2008).
- Krampen, G. (1993): Diagnostik der Kreativität. In: Tests und Trends 10: Jahrbuch der pädagogischen Diagnostik, Trost, I., Jäger (Hrsg.), Beltz, Weinheim, Basel, 11-34.
- Küveler, G. und Schwoch, D. (2006): Informatik für Ingenieure und Naturwissenschaftler. Vieweg+Teubner.
- Kynigos, C. (1995): Programming as a means of expressing and exploring ideas: Three case studies situated in a directive educational system. In A. diSessa, C. Hoyles and R. Noss (eds.), Computers and Exploratory Learning, Springer Verlag NATO ASI Series, Berlin, 399-420.
- Lakhani, K. und Wolf, R. (2005): Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Projects. In: Perspectives on Free and Open Source Software, Feller, J., Fitzgerald, B., et al. (Hrsg.), MIT Press, 3-22.
- Lammers, S. (1986): Programmers at work: interviews. Microsoft, Redmond, Wash.
- Lawrence, R. (2004): Teaching data structures using competitive games. IEEE Transactions on Education, 47 (4), 459- 466.
- Leach, R. J. und Ayers, C. A. (2005): The Psychology of Invention in Computer Science. In: Proceedings of the 17th Annual Workshop of the PPIG, University of Sussex, Brighton UK.
- Lefrancois, G. R. und Leppmann, P. K. (1994): Psychologie des Lernens. Springer, Berlin [u.a.].
- Leon, A. (1997): Being 'Creative' in Software Development. <http://www.alexisleon.com/art/1997/08/06/creativity-in-software.html>, (19.04.2008).
- Lepper, M., Greene, D. und Nisbett, R. (1973): Undermining children's intrinsic interest with extrinsic rewards: A test over the "overjustification" hypothesis. Journal of Personality and Social Psychology, 28, 129-137.
- Lepper, M. R., Corpus, J. H. und Iyengar, S. S. (2005): Intrinsic and Extrinsic Motivational Orientations in the Classroom: Age Differences and Academic Correlates. Journal of Educational Psychology, 97 (2), 184-196.
- Lewandowski, G., Johnson, E. und Goldweber, M. (2005): Fostering a Creative Interest in Computer Science. In: Proceedings of the SIGCSE '05, St. Louis, MO.

- Lewis, T., Petrina, S. und Hile, A. M. (1998): Problem Posing-Adding a Creative Increment to Technological Problem Solving. *Journal of Industrial Teacher Education*, 36 (1), 5-35.
- Ligon, B. (1996): *Connecting chords with linear harmony*. Hal Leonard Publishing Corporation.
- Lippman, S. B. (2006): Ist Programmieren eine Kunst? <http://msdn2.microsoft.com/de-de/library/bb979378.aspx>, (28.04.2008).
- Littler, J. (2005): *Art and Computer Programming*.  
<http://www.onlamp.com/pub/a/onlamp/2005/06/30/artofprog.html>, (28.04.2008).
- LOG IN (1995): Themenheft Computer, Kreativität und Ästhetik. In: LOG IN 15 (4), LOG IN Verlag Berlin.
- Long, J. (2007): Just For Fun: Using Programming Games in Software Programming Training and Education - A Field Study of IBM Robocode Community. *Journal of Information Technology Education*, 6, 279-290.
- Lubart, T. I. (2000): Models of the creative process: Past, present and future. *Creativity Research Journal*, 13 (3-4), 295-308.
- Luthiger Stoll, B. (2006): *Spass und Software-Entwicklung: Zur Motivation von Open-Source-Programmierern*. Dissertation. Universität Zürich, Zürich.
- Maaß, S. und Wiesner, H. (2006): Programmieren, Mathe und ein bisschen Hardware... Wen lockt dies Bild der Informatik? *Informatik-Spektrum*, 29 (2), 125-132.
- Maass, S. und Wiesner, H. (2006): Programmieren, Mathe und ein bisschen Hardware... Wen lockt dies Bild der Informatik? *Informatik-Spektrum*, 29 (2), 125-132.
- Mackin, J. (1996): A creative approach to physics teaching. *Physics Education*, 31 (4), 199-202.
- Maeda, J. (2004): *Creative code*. Thames & Hudson London, New York.
- Maloney, B., Kafai, Rusk, Silverman, Resnick (2004): Scratch: A Sneak Preview. *IEEE Computer Society*, 104 - 109.
- Mamone, S. (1992): Empirical Study of Motivation in an Entry Level Programming Course. *ACM SIGPLAN Notices*, 27 (3), 54-60.
- Manetas, M. (2003): Jackson Pollock.org. <http://www.jacksonpollock.org/>, (27.05.2008).
- Margolis, J., Fisher, A. und Miller, F. (2000): The Anatomy of Interest: Women in Undergraduate Computer Science. *Women's Studies Quarterly*, 28, 104-127.

- Markoff, J. (2005): *What the Dormouse Said: How the 60s Counterculture Shaped the Personal Computer*. Viking Adult.
- MAXQDA (2008): MAXQDA. The Art of Text Analysis.  
<http://www.maxqda.de/produkte/maxqda>, (17.06.2008).
- Mayring, P. (2007): *Qualitative Inhaltsanalyse*. Beltz, Weinheim & Basel.
- MBSJ Brandenburg (2006): *Rahmenlehrplan Informatik für den Unterricht in der gymnasialen Oberstufe im Land Brandenburg*. Verlag Wissenschaft und Technik Berlin, Berlin.
- McIlree, R. (2007): Yet Another "Creativity" Rant Emerges.  
[http://enterprisearchitect.typepad.com/ea/2007/01/yet\\_another\\_cre.html](http://enterprisearchitect.typepad.com/ea/2007/01/yet_another_cre.html), (25.04.2008).
- Meisalo, V., Sutinen, E. und Tarhio, J. (1997): CLAP: teaching data structures in a creative way. In: *Proceedings of the Proceedings of the 2nd conference on Integrating technology into computer science education*, Uppsala, Sweden, 117-119.
- Melymuka, K. (1998): Why musicians may make the best tech workers.  
<http://www.cnn.com/TECH/computing/9807/31/musicians.idg/>, (19.04.2008).
- Mitchell, W. J., Inouye, A. S., Blumenthal, M. S., (U.S.), N. R. C. und Creativity, C. o. I. T. a. (2003): *Beyond productivity: information technology, innovation, and creativity*. National Academies Press, Washington, DC.
- Mittermeir, R. (2000): *Informatik-Unterricht: Bastel-Unterricht, eine intellektuelle Herausforderung oder "Preparation for the information-age"*. *Medienimpulse*, 9/33, 4-11.
- Modrow, E. (2002): *Pragmatischer Konstruktivismus und fundamentale Ideen als Leitlinien der Curriculumentwicklung am Beispiel der theoretischen und technischen Informatik*. Martin-Luther-Universität Halle-Wittenberg.
- Moravcsik, M. J. (1974): *Scientists and Artists: Motivations, Aspirations, Approaches and Accomplishments*. *Leonardo*, 7 (3), 255.
- Moravcsik, M. J. (1981): *Creativity in Science Education*. *Science Education*, 65 (2), 221-27.
- Mumford, M. D. (2003): *Where have we been, where are we going? Taking stock in creativity research*. *Creativity Research Journal*, 15 (2-3), 107-120.
- Noble, J. und Biddle, R. (2002): *Notes on Postmodern Programming*. In: *Proceedings of the ACM conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, 49-71.
- Orloff, M. A. (2006): *Grundlagen der klassischen TRIZ : ein praktisches Lehrbuch des erfinderischen Denkens für Ingenieure*. Springer, Berlin [u.a.].

- Osborn, A. F. (1953): Applied imagination. Scribner, New York.
- Oxford (2007): Compact Oxford English Dictionary. <http://www.askoxford.com>, (14.12. 2007).
- Papert, S. (1993): The children's machine: rethinking school in the age of the computer. BasicBooks, New York.
- Papert, S. (2000): What's the big idea? Toward a pedagogy of idea power. IBM Systems Journal, 39 (3), 720-729.
- Papert, S. und Harel, I. (1991): Situating Constructionism. In: Constructionism, Papert, S. und Harel, I. (Hrsg.), Ablex Publishing, Norwood, N.J.
- Paul, C. (2003): Digital art. Thames & Hudson New York.
- Peppler, K. A. und Kafai, Y. B. (2005): Creative Coding: Programming for Personal Expression.
- Polya, G. (1945): How to Solve It. Princeton University Press, Princeton.
- Pomberger, G. und Dobler, H. (2008): Algorithmen und Datenstrukturen. Pearson Studium, München.
- Prätorius, P. (2004): Virtuelle Ameisenwelt - Digitale Ameisen und Termiten als Modelle künstlichen Lebens in JAVA (Teil 2). LOG IN, (131/132), 81-89.
- Prensky, M. (2001): Digital natives, digital immigrants. On the Horizon, 9 (5), 1-6.
- Racter (1984): The Policeman's Beard Is Half Constructed. Warner Software/Warner Books.
- Rechenberg, P. (2000): Was ist Informatik? Eine allgemeinverständliche Einführung. Hanser, München.
- Reed, D. (2002): The use of ill-defined problems for developing problem-solving and empirical skills in CS1. J. Comput. Small Coll., 18 (1), 121-133.
- Resnick, M. (2002): Rethinking Learning in the Digital Age. In: The Global Information Technology Report: Readiness for the Networked World, Kirkman, G. (Hrsg.), Oxford University Press, Oxford, 32-37.
- Resnick, M. (2007a): All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In: Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition, ACM Press, Washington, DC, USA, 1-6.
- Resnick, M. (2007b): Sowing the Seeds for a More Creative Society. In: Proceedings of the Learning & Leading with Technology, International Society for Technology in Education (ISTE).

- Resnick, M. und Rusk, N. (1996): The Computer Clubhouse: Preparing for life in a digital world. IBM Systems Journal, 35, 431 - 439.
- Rhodes, M. (1961): An analysis of creativity. Phi Delta Kappan, 42 (7), 305-310.
- Rich, L., Perry, H. und Guzdial, M. (2004): A CS1 course designed to address interests of women. In: Proceedings of the 35th SIGCSE technical symposium on Computer science education, ACM Press, Norfolk, Virginia, USA, 190-194.
- Roberts, E. S., Kassianidou, M. und Irani, L. (2002): Encouraging Women in Computer Science. SIGCSE Bulletin., 34, 84-88.
- Romeike, R. (2006): Creative Students - What Can We Learn From Them for Teaching Computer Science? In: Proceedings of the 6th Baltic Sea Conference on Computing Education Research, Koli Calling, Berglund, A. und Wiggberg, M. (Hrsg.), Uppsala University, Uppsala, Sweden.
- Romeike, R. (2007a): Animationen und Spiele gestalten. Ein kreativer Einstieg in die Programmierung. LOG IN, 146/147, 36-44.
- Romeike, R. (2007b): Kriterien kreativen Informatikunterrichts. In: Proceedings of the 12. GI-Fachtagung "Informatik und Schule - INFOS 2007", Siegen, Germany, Schubert, S. (Hrsg.), Köllen, Vol. LNI 112.
- Romeike, R. (2007c): Three Drivers for Creativity in Computer Science Education,. In: Proceedings of the IFIP-Conference on "Informatics, Mathematics and ICT: a golden triangle", Boston, USA.
- Romeike, R. (2008a): Applying Creativity in CS High School Education - Criteria, Teaching Example and Evaluation. In: Proceedings of the 7th Baltic Sea Conference on Computing Education Research ( Koli Calling 2007), Koli National Park, Finland, Lister, R. und Simon (Hrsg.), Conferences in Research and Practice in Information Technology, Vol. 88.
- Romeike, R. (2008b): Bildungsstandards kreativ erreichen - ein Unterrichtskonzept für die Sekundarstufe I. In: Didaktik der Informatik - Aktuelle Forschungsergebnisse. Didaktik der Informatik - Aktuelle Forschungsergebnisse. 5. Workshop der GI-Fachgruppe "Didaktik der Informatik", Erlangen, Brinda, T., Fothe, M. und Hubwieser, P. (Hrsg.), Köllen (Lecture Notes in Informatics (LNI) Bd. LNI-P, Nr. 135), Bonn.
- Romeike, R. (2008c): The Contribution of Computer Science Education in a Creative Society. In: Proceedings of the IFIP World Computer Congress WCC2008 - "TC3 Stream Learning to live in the Knowledge Society", Milan, Italien.
- Romeike, R. (2008d): Sichtweisen einer kreativen Informatik. In: Didaktik der Informatik - Aktuelle Forschungsergebnisse. Didaktik der Informatik - Aktuelle Forschungsergebnisse. 5. Workshop der GI-Fachgruppe "Didaktik der Informatik", Erlangen, Brinda, T., Fothe, M. und Hubwieser, P. (Hrsg.), Köllen (Lecture Notes in Informatics (LNI) Bd. LNI-P, Nr. 135), Bonn.

- Romeike, R. (2008e): Student Teachers' Beliefs about Creativity in Computer Science. In: Proceedings of the IFIP WG 3.1, 3.2 & 3.6 Joint Conference "ICT and Learning for Youths", Kuala Lumpur, Malaysia.
- Romeike, R. (2008f): Where's my Challenge? The Forgotten Part of Problem Solving in Computer Science Education. In: Proceedings of the 3rd ISSEP Intern. Conf. on Informatics in Secondary Schools - Evolution and perspectives, Torun, Polen 2008.
- Romeike, R. und Schwill, A. (2006): "Das Studium könnte zu schwierig für mich sein" - Zwischenergebnisse einer Langzeitbefragung zur Studienwahl Informatik. In: Proceedings of the HDI 2006: Hochschuldidaktik der Informatik, München, Forbig, P., Siegel, G. und Schneider, M. (Hrsg.), Lecture Notes in Informatics, Vol. P-100, 37-49.
- Ruete, E. S. (1990): Response to J. Grudin: The Case against user interface Consistency. ACM Forum: Letter to the editor. Communications of the ACM, 33 (8), 15-16.
- Runco, M. A. (2007): Creativity: Theories and Themes; Research, Development, and Practice. Elsevier, Amsterdam.
- Ruscio, J. und Amabile, T. M. (1996): How does creativity happen? In: Talent Development, Colangelo, N. und Assouline, S. G. (Hrsg.) 3, Ohio Psychology Press, Dayton, OH.
- Saunders, D. und Thagard, P. (2005): Creativity in Computer Science. In: Creativity across domains: Faces of the muse, Kaufman, J. C. und Baer, J. (Hrsg.), Lawrence Erlbaum Associates, Mahwah, NJ.
- Schell, B. und Martin, C. (2006): Webster's New World Hacker Dictionary (Webster's New World).
- Scheuer, W. (2003): Komponieren im Musikunterricht.  
<http://www.musiceducation.de/html/komponieren.html>, (22.04.2008).
- Schmidkunz, H., Lindemann, H. und Schmidkunz, L. (1976): Das forschend-entwickelnde Unterrichtsverfahren : Problemlösen im naturwissenschaftlichen Unterricht. List, München.
- Schubert, S. und Schwill, A. (2004): Didaktik der Informatik. Spektrum Akad. Verl, Heidelberg [u.a.].
- Schulte, C. (2001): Vom Modellieren zum Gestalten–Objektorientierung als Impuls für einen neuen Informatikunterricht? *informatica didactica*, 3.
- Schulte, C. (2003): Lehr-Lernprozesse im Informatik-Anfangsunterricht: theoriegeleitete Entwicklung und Evaluation eines Unterrichtskonzepts zur Objektorientierung in der Sekundarstufe II.
- Schulte, C. und Knobelsdorf, M. (2007): Attitudes towards Computer Science - Computing Experiences as a Starting Point and Barrier to Computer Science. In: Proceedings of the 3rd International Computing Education Research Workshop, New York, USA, 27-38.

- Schwill, A. (1993): Fundamentale Ideen der Informatik. Zentralblatt für Didaktik der Mathematik, 25 (1), 20-31.
- Scragg, G., Baldwin, D. und Koomen, H. (1994): Computer science needs an insight-based curriculum. In: Proceedings of the twenty-fifth SIGCSE symposium on Computer science education, ACM Press, Phoenix, Arizona, United States, 150-154.
- Shasha, D. E. und Lazere, C. A. (1998): Out of Their Minds: The Lives and Discoveries of 15 Great Computer Scientists. Springer, Berlin.
- Shneiderman, B. (2000): Creating Creativity: User Interfaces for Supporting Innovation. ACM Transactions on Computer-Human Interaction, 7 (1), 114–138.
- Shneiderman, B. (2002): Creativity support tools. Commun. ACM, 45 (10), 116-120.
- Shneiderman, B. (2007): Creativity support tools: accelerating discovery and innovation. Commun. ACM, 50 (12), 20-32.
- Shneiderman, B., Fischer, G., Czerwinski, M., Myers, B. und Resnick, M. (2005): Creativity Support Tools. Workshop Report on Creativity Support Tools. National Science Foundation, Washington, DC.
- Smith, D. K., Paradise, D. B. und Smith, S. M. (2000): Prepare Your Mind for Creativity. Communications of the ACM, 43 (7), 110-116.
- Spitzer, M. (2008): Unbewusste Kreativität. Video aus der BR-alpha-Reihe Geist und Gehirn. [http://www.br-online.de/cgi-bin/ravi?verzeichnis=alpha/geistundgehirn/v/&file=spitzer\\_83.rm&g2=1](http://www.br-online.de/cgi-bin/ravi?verzeichnis=alpha/geistundgehirn/v/&file=spitzer_83.rm&g2=1), (1.06.2008).
- Starko, A. J. (2001): Creativity in the classroom : schools of curious delight. Erlbaum, Mahwah, NJ [u.a.].
- Sternberg, R. J. (2003): Creative Thinking in the Classroom. Scandinavian Journal of Educational Research, 47 (3), 325-338.
- Sternberg, R. J. (2004): Creativity : from potential to realization. American Psychological Assoc., Washington, DC.
- Sternberg, R. J. und Lubart, T. I. (1991): Creating Creative Minds. Phi Delta Kappan, 72 (8), 608-614.
- Sternberg, R. J., Pretz, J. E. und Kaufman, J. C. (2002): The Creativity Conundrum: A Propulsion Model of Kinds of Creative Contributions. Psychology Press, New York.
- Sternberg, R. J. und Williams, W. M. (1996): How to Develop Student Creativity. ASCD, Alexandria, VA.

- Stribling, J., Krohn, M. und Aguayo, D. (2005): SCIgen - An Automatic CS Paper Generator. <http://pdos.csail.mit.edu/scigen/>, (22.05.2008).
- Sutinen, E. und Tarhio, J. (2001): Teaching to identify problems in a creative way. In: Proceedings of the 31st Frontiers in Education Conference, IEEE Computer Society, Vol. 1.
- Sweeney, R. B. (2003): Creativity in the Information Technology Curriculum Proposal. In: Proceedings of the 4th conference on Information technology curriculum, Lafayette, Indiana, USA, 139-141.
- Tharp, A. L. (1981): Getting more oomph from programming exercises. SIGCSE Bull., 13 (1), 91-95.
- Thomas, M. (2002): Informatische Modellbildung: Modellieren von Modellen als zentrales Element der Informatik für den allgemeinbildenden Schulunterricht. Universität Potsdam, Potsdam.
- Thomas, M. (2005): Geschichten aus der Geschichte der Informatik. LOG IN, 136/137, 41-46.
- Torrance, E. P. (1974): Torrance Test of Creative Thinking. Scholastic Testing service, Bensenville, IL.
- Treffinger, D. J. (1980): Encouraging creative learning for the gifted and talented. Ventura County Schools/LTI, Ventura, CA.
- Trogemann, G. (2001): Computer und Kreativität. In: Proceedings of the „Mensch und Computer“, Stuttgart, Oberquelle, H., Oppermann, R. und Krause, J. (Hrsg.), Teubner.
- Trogemann, G. und Viehoff, J. (2005): Code@ Art, Eine elementare Einführung in die Programmierung als künstlerische Praktik. Springer-Verlag, Wien, New York.
- Turner, G., Weakley, A., Zhang, Y. und Edmonds, E. (2005): Attuning: A Social and Technical Study of Artist-Programmer Collaborations. 106 - 119. Sussex University.
- Ursyn, A. und Scott, T. (2007): Web with art and computer science. In: "ACM SIGGRAPH 2007 educators program". ACM, San Diego, California.
- Ursyn, A. und Sung, R. (2007): Learning science with art. In: "ACM SIGGRAPH 2007 educators program". ACM, San Diego, California.
- Van Dyke, C. (1987): Taking "computer literacy" literally. Communications of the ACM, 30 (5), 366-374.
- VanLangen, P. H. G., Wijngaards, N. J. E. und Brazier, F. M. T. (2004): Towards Designing Creative Artificial Systems.

- VDE (2003): VDE Studie "Young Professionals". VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V., Frankfurt am Main.
- Vollrath, H.-J. (1987): Begriffsbildung als schöpferisches Tun im Mathematikunterricht. Zentralblatt für Didaktik der Mathematik, 123-127.
- Wallas, G. (1926): The Art of Thought. Harcourt Brace, New York.
- Wands, B. (2002): Digital Creativity: Techniques for Digital Media and the Internet. Wiley, New York.
- Westby, E. L. und Dawson, V. L. (1995): Creativity: Asset or Burden in the Classroom? Creativity Research Journal, 8 (1), 1-10.
- Weth, T. (1999): Kreativität im Mathematikunterricht: Begriffsbildung als kreatives Tun. Franzbecker, Hildesheim [u.a.].
- Weth, T. (2001): Kreative Produkte. Kreativität, mathematik lehren, 106, 42-45.
- Wheeler, D. A. (2008): The most important software innovations.  
<http://www.dwheeler.com/innovation/innovation.html>, (15.04.2008).
- Wheeler, S., Waite, S. J. und Bromfield, C. (2002): Promoting creative thinking through the use of ICT. Journal of Computer Assisted Learning, 18 (3), 367-378.
- Wikipedia (2008): Digitale Kunst. [http://de.wikipedia.org/wiki/Digitale\\_Kunst](http://de.wikipedia.org/wiki/Digitale_Kunst), (30.05.2008).
- Willms, J., Wentzlaff, I. und Specker, M. (2000): Kreativität in der Informatik: Anwendungsbeispiele der innovativen Prinzipien aus TRIZ. In: Proceedings of the Informatik 2000, Neue Horizonte im neuen Jahrhundert, 30. Jahrestagung der Gesellschaft für Informatik, Berlin, Mehdorn, K. und G.Snelting (Hrsg.), Springer-Verlag.
- Winter, H. (1999): Gestalt und Zahl - Perspektiven eines kreativen Mathematikunterrichts in der Schule. In: Proceedings of the Kreatives Denken und Innovationen in mathematischen Wissenschaften Jena, Zimmermann, B., David, G., et al. (Hrsg.), Univ. Jena, Vol. 99/29, 213-225.
- Wordsmyth (2007): Wordsmyth Educational Dictionary-Thesaurus. <http://www.wordsmyth.net/>, (14 Dec 2007).
- Zimbardo, P. G. und Gerrig, R. J. (2004): Psychologie. Pearson Studium, München.

## Abbildungsverzeichnis

Abbildung 1: Aufgabenbeispiele aus dem TTCT (Version kreatives Denken mit Worten) .....	8
Abbildung 2: Modelle des kreativen Prozesses im Vergleich .....	9
Abbildung 3: Amabiles Componential Model of Creativity .....	10
Abbildung 4: Flow als optimale Balance zwischen Anforderungen und Fähigkeiten .....	14
Abbildung 5: 25 ways to develop creativity .....	17
Abbildung 6: Beispiel für die Kreativitätsroutine „Modifizieren“ .....	20
Abbildung 7: Beispiel: Erfinden mathematischer Begriffe .....	21
Abbildung 8: Phasen der Problembausteine .....	23
Abbildung 9: Ausschnitte einer Online-Forumsdiskussion zur Kreativität in der Informatik .....	27
Abbildung 10: Image des Ingenieurberufs .....	27
Abbildung 11: (Fehl-)Vorstellung von Kreativität in der Informatik .....	28
Abbildung 12: Aussagen kreativer Persönlichkeiten zu ausschlaggebenden Faktoren ihrer Entwicklung .....	29
Abbildung 13: Blackbox-Modell des musikalischen Designprozesses .....	30
Abbildung 14: Modell des kreativen Softwaredesignprozesses .....	32
Abbildung 15: Informatik als Darstellung anhand ihrer Innovationen .....	34
Abbildung 16: Knuth: The Art of Computer Programming .....	36
Abbildung 17: Kunstwerke aus der DEMO-Szene .....	39
Abbildung 18: „Listen“ (Hopkins) .....	40
Abbildung 19: Interaktive Kunst nach Pollock .....	41
Abbildung 20: Ausprägungen Digitaler Kunst laut Wikipedia .....	41
Abbildung 21: Visualisierung der Klasse „Cow“ .....	42
Abbildung 22: Vier Aktivitäten und Aufgaben im Kreativitäts-Framework .....	43
Abbildung 23: Einschätzung eigener Kreativität von Schülern .....	48
Abbildung 24: Creative thinking spiral .....	53
Abbildung 25: Challenge-Cycle .....	58
Abbildung 26: Modell der Einflussfaktoren von Kreativität .....	61
Abbildung 27: Relevante Kreativitätsperspektiven aus Sicht der Informatik .....	62
Abbildung 28: Faktorenmodell eines kreativen Informatikunterrichts .....	63
Abbildung 29: Implikationen für die Gestaltung kreativen Unterrichts .....	65
Abbildung 30: Hauptkategorien der Strukturdimensionen und häufige identifizierte Codes .....	68
Abbildung 31: Lernziele der Unterrichtssequenz .....	86
Abbildung 32: Zustimmungsänderung im Vergleich zur vorherigen Unterrichtssequenz .....	90
Abbildung 33: Einschätzung der Unterrichtsmethodik durch die Schüler .....	92
Abbildung 34: Zustimmung im Vergleich zur Kontrollgruppe .....	93
Abbildung 35: Brainstorming zu Kreativität und Informatik (Schülerbeispiel) .....	95
Abbildung 36: Häufige Schülerantworten des Brainstormings zu Kreativität und Informatik .....	96