

# Université du Luxembourg

Faculté des Lettres, des Sciences Humaines, Arts et des Sciences de l'Éducation  
Bachelor en Sciences de l'Éducation (Bachelor professionnel)

## **Welche Faktoren fördern oder behindern die erfolgreiche Implementierung von Computational Thinking als Schulfach in Grundschulen?**

Anna DEMUTH

unter der Begleitung von: Prof. Dr. Robert A.P. REUTER

Zweitgutachter: Gilbert BUSANA

Bachelorarbeit

2023/2024

## Eidesstattliche Erklärung

Titel der Arbeit: Welche Faktoren fördern oder behindern die erfolgreiche Implementierung von Computational Thinking als Schulfach in Grundschulen?

Abgabesemester: Sommersemester 2024

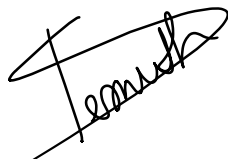
Name, Vorname der Verfasserin: DEMUTH, Anna

Matrikel-Nummer: 0201778447

„Ich erkläre hiermit, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, habe ich als solche gemäß der im Studiengang *Bachelor en Sciences de l'Education* üblichen APA-Norm gekennzeichnet.“

Ort, Datum: Luxemburg, den 24. 05. 2024

Unterschrift:

A handwritten signature in black ink, appearing to read 'Anna Demuth', written over a horizontal line.

## Zusammenfassung

Diese theoretische Bachelorarbeit untersucht die förderlichen und hinderlichen Faktoren bei der Implementierung von Computational Thinking (CT) als Schulfach in Grundschulen. Unter Anwendung einer angepassten Version der "14 Essential Conditions" der International Society for Technology in Education wird eine systematische Literaturanalyse durchgeführt, die die identifizierten Hindernisse und Erfolgsfaktoren für CT detailliert kategorisiert. Die Ergebnisse bieten praxisorientierte Empfehlungen zur effektiven Einführung von CT in luxemburgischen Grundschulen.

**Schlüsselwörter:** Computational Thinking, informatisches und algorithmisches Denken, Grundschulbildung, Erfolgsfaktoren und Hindernisse, Implementierung

## Résumé

Ce mémoire théorique examine les facteurs favorables et défavorables à l'implémentation de la pensée informatique (CT) en tant que matière scolaire dans les écoles primaires. En utilisant une version adaptée des "14 Essential Conditions" de l'International Society for Technology in Education, une analyse systématique de la littérature est réalisée, catégorisant en détail les obstacles et les facteurs de succès identifiés pour la CT. Les résultats offrent des recommandations pratiques pour l'introduction efficace de la CT dans les écoles primaires luxembourgeoises.

**Mots-clefs :** Computational Thinking, pensée computationnelle, éducation primaire, facteurs de succès et obstacles, mise en œuvre

## **Danksagungen**

Ich möchte diese Gelegenheit nutzen, um meinem Tutor, Robert Reuter, meinen tiefsten Dank auszusprechen. Seine umfangreichen Kenntnisse im Fachbereich und fachliche Anleitung haben wesentlich zu meiner akademischen Entwicklung und zum erfolgreichen Abschluss dieser Arbeit beigetragen. Herr Reuter hatte stets ein offenes Ohr für meine Fragen und Zweifel und konnte dank seines tiefgreifenden Wissens immer weiterhelfen. Ich bin sehr dankbar für die Zeit und Energie, die Herr Reuter in die Betreuung meiner Arbeit investiert hat, und für das Vertrauen, das er in meine Fähigkeiten gesetzt hat.

Vielen Dank für alles!

Ich möchte außerdem meiner Mitstudentin Marie-Claire Dione meinen tiefsten Dank aussprechen. Seit dem ersten Tag dieses Studiums hat sie mich stets unterstützt und die letzten vier Jahre zu einer wunderbaren Erfahrung gemacht. Ohne sie wäre dieser Weg weitaus schwieriger und weniger humorvoll gewesen. Marie war nicht nur eine hervorragende Studienkollegin, sondern auch eine inspirierende Freundin, die mir in herausfordernden Zeiten stets zur Seite stand und mich motiviert hat, mein Bestes zu geben.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
<b>2</b>	<b>Theoretischer Rahmen</b>	<b>10</b>
2.1	Einführung in Computational Thinking	10
2.1.1	Entwicklung und Status von Computational Thinking in der Bildung	10
2.1.2	Kompetenzrahmenwerk	12
2.2	Integration von Computational Thinking im luxemburgischen Bildungssystem	15
2.2.1	Überblick von der Primar- bis zur Sekundarstufe	15
2.2.2	Computational Thinking im Luxemburger Lehrplan und Ergänzungen zum Lehrplan	16
2.2.3	Bisher identifizierte Herausforderungen	17
2.3	ISTE Essential Conditions	18
<b>3</b>	<b>Methode</b>	<b>20</b>
3.1	Forschungsansatz	20
3.2	Methoden der Datenanalyse	20
3.3	Analyseschritte	23
<b>4</b>	<b>Forschungsergebnisse</b>	<b>25</b>
4.1	Gemeinsame Vision	25
4.1.1	Erfolgsfaktoren	25
4.1.2	Hindernisse	26
4.2	Befähigte Führungskräfte	27
4.2.1	Erfolgsfaktoren	28
4.2.2	Hindernisse	29
4.3	Umsetzungsplanung	30
4.3.1	Erfolgsfaktoren	30
4.3.2	Hindernisse	36
4.4	Kontinuierliche und ausreichende Finanzierung	38
4.4.1	Erfolgsfaktoren	38
4.4.2	Hindernisse	38
4.5	Gerechter Zugang	39
4.5.1	Erfolgsfaktoren	39
4.5.2	Hindernisse	40
4.6	Fachkundiges Personal	42
4.6.1	Erfolgsfaktoren	43
4.6.2	Hindernisse	44
4.7	Kontinuierliche Weiterbildung	46
4.7.1	Erfolgsfaktoren	47
4.7.2	Hindernisse	48
4.8	Technische Unterstützung	49
4.8.1	Erfolgsfaktoren	50
4.8.2	Hindernisse	50
4.9	Lehrplanrahmen	51
4.9.1	Erfolgsfaktoren	51
4.9.2	Hindernisse	52
4.10	Schülerzentriertes Lernen	54
4.10.1	Erfolgsfaktoren	54
4.10.2	Hindernisse	55
4.11	Bewertung und Evaluation	57
4.11.1	Erfolgsfaktoren	57

4.11.2	Hindernisse.....	59
4.12	Engagierte Gemeinschaften.....	61
4.12.1	Erfolgsfaktoren.....	61
4.12.2	Hindernisse.....	63
4.13	Unterstützende Richtlinien.....	64
4.13.1	Hindernisse.....	64
4.14	Unterstützender externer Kontext.....	65
4.14.1	Erfolgsfaktoren.....	65
<b>5</b>	<b>Diskussion.....</b>	<b>66</b>
5.1	Interpretation der Ergebnisse.....	66
5.2	Grenzen der Arbeit.....	71
5.3	Implikationen.....	73
5.3.1	Implikationen für zukünftige Forschung.....	73
5.3.2	Implikationen für die Praxis in Luxemburg.....	75
<b>6</b>	<b>Fazit.....</b>	<b>78</b>
<b>7</b>	<b>Literaturverzeichnis.....</b>	<b>79</b>

# 1 Einleitung

Bei der Frage, was eigentlich ein Computer ist, denken viele sofort an eine elektronische Blackbox, die erstaunliche Funktionen ausführt, indem sie Daten erhebt, speichert, abrufen und umwandelt. Dabei können die meisten unserer Geräte als Computer betrachtet werden: von Telefonen und Tablets über Webseiten und Navigatoren bis hin zu Uhren, medizinischen Geräten und WLAN-Netzwerken. Heutzutage sind Computer nicht mehr auf Desktops oder Laptops beschränkt, sondern durchdringen fast alle Bereiche unseres Lebens. Die Palette reicht von Bankgeschäften, Transport, Hotelreservierungen, Filmproduktionen, Unterhaltung, Online-Kursen bis hin zu Google-Suchanfragen. Fast alles wird von Computern betrieben, die über ein unsichtbares globales Netzwerk, die ‚Cloud‘, miteinander vernetzt sind. Jeder Aspekt unseres Lebens, der für die Digitalisierung zugänglich ist, wird derzeit digitalisiert. Laut Denning und Tedre (2019) befinden wir uns somit an der Schwelle einer Informatikrevolution.

Moderne Generationen werden in einer Welt leben und arbeiten, die stark von Computern und deren Informatikprinzipien geprägt ist (Barr & Stephenson, 2011; Grover & Pea, 2013). Die Arbeitswelt erfährt derzeit einen drastischen Wandel durch Technologie und Digitalisierung, wobei auch etablierte Berufsfelder betroffen sind (World Economic Forum, 2023). Harteis (2018) betont, dass die Digitalisierung nicht nur die industrielle Arbeit betrifft, sondern sich auch auf alle Wirtschaftssektoren auswirkt. Infolgedessen sehen sich die Arbeitnehmer mit komplexen Herausforderungen konfrontiert, die sowohl Wissens- und Handlungskompetenzen als auch Ausdauer und Engagement erfordern (Sembill et al., 2001).

Um den Veränderungen und Anforderungen der modernen Gesellschaft gerecht zu werden, müssen junge Menschen eine breite Palette von Kenntnissen und Kompetenzen entwickeln (Wenzlik, 2013). Fähigkeiten wie das Finden alternativer Lösungsansätze, die kreative Problemlösung oder die Betrachtung alltäglicher Herausforderungen aus neuen Perspektiven sind

entscheidend (Bocconi et al., 2016). Bildungseinrichtungen müssen sicherstellen, dass Lernende auf eine digitale Zukunft vorbereitet werden, was auch kritisches Denken und Problemlösungsfähigkeiten umfasst (Sembill et al., 2001).

In diesem Zusammenhang gewinnt Computational Thinking, auch abgekürzt CT oder informatisches und algorithmisches Denken genannt, an Bedeutung. CT ist eine grundlegende Fähigkeit, die für den Erfolg in der modernen Gesellschaft unverzichtbar ist und von jedem beherrscht werden sollte (Wing, 2006), da CT-Konzepte in allen Disziplinen von zentraler Bedeutung sind (Bundy, 2007). Das Konzept des CT spielt eine dominierende Schlüsselrolle und entwickelt sich zu einer essenziellen Fähigkeit im 21. Jahrhundert (Wing, 2008; National Research Council [NRC] et al., 2012). Dieses Bewusstsein hat zu einem globalen Interesse an der Integration von CT in den Lehrplan geführt (Liu et al., 2011; Balanskat & Engelhardt, 2015). Um eine praxisnahe und verständliche Definition von CT zu bieten, haben die Computer Science Teachers Association (CSTA) und die International Society for Technology in Education (ISTE) eine Definition entwickelt, die besonders im schulischen Kontext relevant ist. CT wird als ein Problemlösungsprozess beschrieben, der das Formulieren von Problemen zur Lösung durch Computer, das logische Organisieren und Analysieren von Daten, das Darstellen von Daten durch Modelle und Simulationen, algorithmisches Denken und das Übertragen des Problemlösungsprozesses auf verschiedene Probleme umfasst (CSTA & ISTE, 2011).

Die International Computer and Information Literacy Study (ICILS) von 2018 evaluierte die CT-Fähigkeiten in acht Ländern, darunter Luxemburg, wo die Schüler schlechter als der internationale Durchschnitt abschnitten (Milmeister & Baumann, 2021). Folglich wurde CT im September 2020 unter dem Namen ‚Coding‘ ins luxemburgische Bildungssystem eingeführt. Der Prozess begann im Mathematikunterricht des Zyklus 4 und wurde im folgenden Jahr auf die Zyklen 1, 2 und 3 ausgeweitet (SCRIPT, 2020). Diese Integration markiert einen neuen Bereich im luxemburgischen



Schulsystem. Hinsichtlich der Umsetzung dieses Programms äußerte sich der luxemburgische Bildungsminister Claude Meisch in einem Interview im Jahr 2020, dass die kohärente Implementierung von CT, beginnend in den unteren Grundschulzyklen bis hin zu den höheren Klassenstufen der Sekundarschulen, eine erhebliche Herausforderung darstellt. Insbesondere die Anpassung der Lehrkräfte an die neuen Anforderungen sei anspruchsvoll, da sie in der Entwicklung voranschreiten und das volle Potenzial der verfügbaren Unterrichtswerkzeuge ausschöpfen müssen (Meisch, 2020, zitiert nach Rings, 2020, S.5)

Diesbezüglich ist es interessant, zu untersuchen, wie unterschiedliche Länder CT in der K-12 Bildung integrieren. Das Hauptziel dieser Forschungsarbeit besteht darin, eine systematische Auswertung publizierter Forschung durchzuführen, die bisherige Umsetzungen von CT im K-12 Bildungsbereich untersucht und als erfolgreich, beziehungsweise hinderlich validiert haben. Die zentrale Fragestellung, die dieser Forschung zugrunde liegt, lautet: „Welche Faktoren fördern oder behindern die erfolgreiche Implementierung von Computational Thinking als Schulfach in Grundschulen?“. Die Klärung dieser Frage ist entscheidend, da sie eine effiziente und zielgerichtete Integration von CT in den Lehrplan ermöglicht. Die Analyse bietet wertvolle Empfehlungen zur effektiven CT-Lehre, zur gezielten Bewältigung von Herausforderungen und zur optimalen Ressourcennutzung. Dadurch unterstützt die Untersuchung Bildungseinrichtungen in Luxemburg auf dem Weg zu einem modernen und zukunftsorientierten Bildungssystem.

## **2 Theoretischer Rahmen**

### **2.1. Einführung in Computational Thinking**

#### **2.1.1 Entwicklung und Status von Computational Thinking in der Bildung**

Obwohl CT oft als Produkt des Computerzeitalters angesehen wird, das in den 1940er Jahren begann, ist diese Annahme nicht ganz zutreffend. Schon vor der Ära der modernen Computer gab es eine Gruppe mathematisch versierter Experten, die komplexe Berechnungen durchführten und als „Rechner“ (auf Englisch „Computer“) bekannt waren. Diese „Rechner“ waren keineswegs die Ersten ihrer Art, da der Begriff selbst, was sinngemäß „jemand, der rechnet“ bedeutet, bereits im frühen 17. Jahrhundert verwendet wurde. Tatsächlich lassen sich die Ursprünge des CT bis zu den Babyloniern, die heute als Vorläufer des CT gelten, und sogar den ägyptischen Ingenieuren beim Pyramidenbau zurückverfolgen. Diese historischen Beispiele zeigen, dass viele Prinzipien des informatischen und algorithmischen Denkens lange vor der Entwicklung elektronischer Computer existierten (Denning & Tedre, 2019).

Über die Jahrhunderte hinweg entwickelte sich das informatische und algorithmische Denken stetig weiter und nahm zunehmend anspruchsvollere Formen an. Die Transformation in eine professionelle Disziplin erfolgte jedoch erst mit dem Aufkommen des elektronischen Computers in den 1950er Jahren. Zu dieser Zeit begannen Universitäten wie Cambridge und Purdue die ersten Informatik-Studienprogramme anzubieten, obwohl viele akademische Institutionen zunächst skeptisch waren und die Gründung von Informatikabteilungen oft aus politischen Auseinandersetzungen resultierte (Denning & Tedre, 2019). In der Grund- und Sekundarschulbildung erfolgte die Einführung von CT langsamer. In den 1960er Jahren wurden jedoch signifikante Fortschritte erzielt, darunter die Entwicklung der

Programmiersprachen BASIC<sup>1</sup> und Logo. Ein weiterer Meilenstein in der Informatikbildung war Seymour Paperts Buch „Mindstorms“ aus dem Jahr 1980, in dem er den Begriff ‚Computational Thinking‘ prägte und die Bedeutung des Lernens durch Programmieren hervorhob (Denning & Tedre, 2019; Kong & Abelson, 2019).

In den darauffolgenden Jahren manifestierte sich die Vorstellung, dass Informatik ein kraftvolles intellektuelles Werkzeug für alle Kinder sein könnte, welches sie selbstständig erlernen, manipulieren, erweitern und in Projekten anwenden, um eine umfassende und differenzierte Beherrschung der Welt zu erreichen (Papert, 1971). Papert betonte, dass „true computer literacy is not just knowing how to make use of computers and computational ideas. It is knowing when it is appropriate to do so“ (S. 155). Schon in dieser frühen Phase bestand das Anliegen darin, den Unterschied zwischen CT und dem Wissen darüber, wie man programmiert oder computergestütztes Werkzeug nutzt, klar herauszustellen. Diese Herausforderung der Klarheit bleibt auch während der weiteren Entwicklung der CT-Bewegung bestehen (Kong & Abelson, 2019).

Trotz Fortschritten in den 1980er und 1990er Jahren blieb die Standardisierung der Computerbildung in Schulen eine Herausforderung. Kurse konzentrierten sich oft auf die Anwendung von Tools wie Textverarbeitungsprogrammen und Tabellenkalkulationen, statt auf Programmieren (Denning & Tedre, 2019). Es wurde wenig auf die Vorstellung geachtet, dass die Informatik ein kraftvoller Lernrahmen für alle sein könnte, nicht nur für Schülerinnen und Schüler, die eine Karriere im Bereich der Computerprogrammierung anstreben (Kong & Abelson, 2019). Ein Wendepunkt war 2006, als Jeannette Wing eine wegweisende Kolumne veröffentlichte, die CT als grundlegende Fähigkeit für alle definierte. Sie initiierte die „CT für alle“-Bewegung, die bedeutende Ressourcen mobilisierte, um CT in den

---

<sup>1</sup> Beginner’s All-purpose Symbolic Instruction Code

Bildungskontext zu integrieren (Denning & Tedre, 2019). Dies löste ein wachsendes Interesse bei Forschern, Pädagogen und politischen Entscheidungsträgern aus, sodass das Konzept des informatischen und algorithmischen Denkens zunehmende Bedeutung gewann und als ein immer relevanteres Forschungsfeld etabliert wurde (Grover & Pea, 2013). Große Organisationen entwickelten umfassende Lehrplankonzepte für K-12 Computational Thinking, förderten Coding-Initiativen wie Clubs und Bootcamps und trugen dazu bei, dass CT zu einem zentralen Begriff in zahlreichen Nachrichten, Blogs und Forschungsprojekten zur Informatikbildung wurde (Denning & Tedre, 2019).

Trotz des steigenden Interesses besteht immer noch eine allgemeine Unklarheit darüber, was genau unter CT zu verstehen ist (Kong & Abelson, 2019). Die zügige Einbeziehung zahlreicher begeisterter Neulinge, die mit der langen Geschichte des CT nicht vertraut waren, führte zu erheblicher Verwirrung bezüglich der Definitionen und Lernziele von CT. Einige entwickelten gänzlich neue CT-Frameworks für K-12 Schulen, die essenzielle Konzepte wegließen und Missverständnisse über Informatik und Algorithmen enthielten. Dies führte zu Spannungen und Diskussionen, insbesondere bezüglich der Unterschiede zwischen CT für Einsteiger und CT für erfahrene Fachleute. Es ist wichtig anzumerken, dass grundlegendes CT in K-12 erheblich von fortgeschrittenem CT in der Hochschulbildung und in verschiedenen Anwendungskontexten abweicht (Denning & Tedre, 2019).

### **2.1.2 Kompetenzrahmenwerk**

Parmentier et al. (2020) haben im Rahmen des PIAF<sup>2</sup> -Projektes ein CT-Kompetenzrahmenwerk entwickelt, welches sich an internationalen Standards sowie den IT-Lehrplänen angelsächsischer Länder orientiert. Besonders hervorzuheben ist seine Ausrichtung auf die Entwicklung von

---

<sup>2</sup> PIAF steht abgekürzt für „développer la Pensée Informatique et Algorithmique dans l’enseignement Fondamental“

Kompetenzen insbesondere in der Grundausbildung (Kindergarten und Grundschule), welche im Fokus dieser Arbeit liegt. Das Rahmenwerk zeichnet sich durch eine klare Struktur und eine verständliche Sprache aus, was es auch für Nicht-Spezialisten wie Lehrkräfte und Schülerinnen und Schüler zugänglich macht. Die praxisorientierten Beispiele und leicht verständlichen Anleitungen ermöglichen es den Lehrkräften, CT-Konzepte ohne umfangreiche Vorkenntnisse zu vermitteln. Das Rahmenwerk bietet klare Orientierungspunkte und gibt den Lehrpersonen konkrete Hinweise darauf, was bei der Umsetzung von ihnen erwartet wird. Gleichzeitig zeigt es auf, welche Anforderungen sie an die Schülerinnen und Schüler stellen sollen. Diese Zugänglichkeit ist entscheidend für eine wirksame Implementierung in Bildungsumgebungen, insbesondere an Grundschulen, wo eine klare und verständliche Konzeption von CT von großer Bedeutung ist.

Die sechs Hauptkompetenzen des Rahmenwerks sind jeweils in grundlegende (Erwerb zwischen 5 und 8 Jahren) und fortgeschrittene (Erwerb zwischen 9 und 12 Jahren) Teilkompetenzen untergliedert. Diese Strukturierung ermöglicht eine zielgerichtete Entwicklung von Fähigkeiten im Bereich CT entsprechend dem Alter und Entwicklungsstand der Schülerinnen und Schüler. Die 6 Hauptkompetenzen werden im Folgenden grob definiert (Parmentier et al., 2021):

- Kompetenz C1 - Abstraktionen definieren / generalisieren: Unter der Kompetenz 1 wird verstanden, wie man Abstraktionen und Verallgemeinerungen verwendet, um komplexe Informationen zu strukturieren und verständlicher zu machen. Dies umfasst beispielsweise, wie man Objekte (oder Gruppen von Objekten) und Aktionen identifiziert, benennt und kategorisiert, um sie besser zu organisieren. Bei einem Objekt wird dabei zwischen Konstanten (Objekte, die nicht veränderbar sind, z.B. Vornamen) und Variablen (Objekte, die sich verändern können, z.B. Außentemperatur) unterschieden.

- Kompetenz C2 - Eine Sequenz von Aktionen komponieren/dekomponieren: Dies bedeutet, zu lernen, wie man geschickt eine Abfolge von Aktionen gestalten oder analysieren kann, um ein bestimmtes Ziel besser zu erreichen. Diese Aktionen können (Gruppen von) Objekte(n) involvieren, wie sie in C1 beschrieben werden. Dies umfasst zum Beispiel das Anordnen von Aktionen in einer optimalen Reihenfolge, das Hinzufügen von fehlenden Aktionen zu einer unvollständigen Aktionsserie oder die Zerlegung komplexer Ziele in kleinere, leichter handhabbare Teilziele. Auch können verschiedene Aktionsserien zusammengeführt werden, um neue Ziele zu verwirklichen.
- Kompetenz C3 - Kontrolle über eine Sequenz von Aktionen: Die dritte Kompetenz impliziert, eine Sequenz von Aktionen gezielt auszuwählen und in geeigneter Weise eine festgelegte Anzahl von Malen zu wiederholen, um spezifische Probleme kontrollierter und effektiver zu lösen. Diese Fähigkeit umfasst einerseits die Anwendung einfacher Bedingungen, die auf einem bestimmten Kriterium basieren (Wenn...dann), sowie auch komplexe Bedingungen, die auf mehreren Kriterien beruhen und durch logische Operatoren wie ‚und‘, ‚oder‘ und ‚nicht‘ kombiniert werden (z.B. Wenn...und/oder...dann).
- Kompetenz C4 - Objekte oder Sequenzen von Aktionen bewerten: Unter der vierten Kompetenz wird verstanden, Objekte oder Sequenzen von Aktionen anhand spezifischer Kriterien zu bewerten und zu verbessern. Dies schließt das Vergleichen von mindestens zwei Objekten oder Aktionen unter Verwendung von einem oder mehreren Kriterien ein (z.B. Anzahl der Zeilen, Lesbarkeit, Ausführungszeit, usw.). Die Kompetenz impliziert, gezielte Bewertungen vorzunehmen und Ablaufsequenzen so zu verbessern, dass effizientere Prozesse und Ergebnisse erzielt werden.
- Kompetenz C5 - Informationen in verschiedene Darstellungen übersetzen: Die Kompetenz 5 umfasst, wie Informationen oder Handlungen mithilfe von eindeutigen und präzisen Sprachen oder Codes dargestellt werden

können. In anderen Worten bedeutet dies, Symbole oder Codes zwischen verschiedenen äquivalenten Darstellungen zu übersetzen (z.B.  $2 \times 2$  und  $8/2$ ).

- Kompetenz C6 - Iterativ eine Sequenz von Aktionen konstruieren: Die sechste Kompetenz beinhaltet die schrittweise Konstruktion, Überprüfung und Anpassung von Abläufen, bis das angestrebte Ergebnis erreicht ist. Dies schließt die Identifikation von Fehlern in einer Handlungssequenz und deren Korrektur ein, um das definierte Ziel zu erreichen. Des Weiteren besteht die Möglichkeit, eine gegebene Sequenz für ein neues Ziel zu erweitern oder zu modifizieren. Der Schwerpunkt liegt dabei auf dem iterativen Prozess der kontinuierlichen Verbesserung.

## **2.2. Integration von Computational Thinking im luxemburgischen Bildungssystem**

### **2.2.1. Überblick von der Primar- bis zur Sekundarstufe**

Im September 2020 begann erstmals die Integration von CT und Kodierung in das luxemburgische Bildungssystem im Zyklus 4 der Primarstufe, der Schülerinnen und Schüler im Alter von 10 bis 12 Jahren umfasst, und zwar im Fach Mathematik. Dort wurde die Initiative Digital(4)Education gestartet, die das CT sowohl als pädagogisches Instrument als auch als Unterrichtsfach betrachtet, und das Programmieren in den Klassen des Zyklus 4 einführen soll. Im September 2021 wurde dieser Ansatz auf die Zyklen 1, 2 und 3 (4-9 Jahre) ausgeweitet, wo Programmierkenntnisse zusammen mit mathematischen Fähigkeiten und in verschiedenen Fächern wie Sprachen, Naturwissenschaften oder Kreativität fächerübergreifend unterrichtet werden (SCRIPT, 2020). Im Sekundarschulbereich werden Medienbildungsinhalte in verschiedenen Fachbereichen integriert. Seit 2022 bietet das Fach „Digital Sciences“ in den unteren Sekundarklassen eine Einführung in das Leben in einer digitalen Gesellschaft.

Die Lehrinhalte werden von den zuständigen Programmkommissionen, den „Commissions nationales de l'enseignement secondaire“, festgelegt. In zahlreichen Fächern und Modulen erhalten Lehrkräfte die Gelegenheit, medienpädagogische Themen in ihren Unterricht zu integrieren (SCRIPT, 2022). Das Ziel der CT-Bildung in Luxemburg ist es, Schülerinnen und Schüler auf die digitale Welt und die Berufe von morgen vorzubereiten (Luxembourg Centre for Educational Testing [LUCET] et al., 2021).

Im Frühjahr 2021 wurden 15 spezialisierte Lehrerstellen geschaffen, um die Integration des informatischen und algorithmischen Denkens und Programmierens in den Grundschullehrplan zu unterstützen (Ministère de l'Éducation nationale, de l'Enfance et de la Jeunesse [MENJE], 2021). Fortbildungen und Konferenzen zur Sensibilisierung sowie CT-Einführungssets wurden bereitgestellt (z.B. Ozobots, Lux-Robo Modikit). Bildungsressourcen und -projekte (z.B. Coding4Kids, edumedia.lu, educoding.lu) sowie Bildungseinrichtungen (z.B. Code Club Luxembourg, The Luxembourg Media and Digital Design Centre) unterstützen die Lehrplanintegration und sollen das Interesse der Schüler an neuen Technologien und konzeptionellem Denken wecken (Women In Digital Empowerment [WIDE], o.D.).

### **2.2.2. Computational Thinking im Luxemburger Lehrplan und Ergänzungen zum Lehrplan**

Im luxemburgischen Lehrplan, insbesondere im Bereich „Éducation aux médias“, sind fünf Bereiche mit verschiedenen transversalen Kompetenzen aufgeführt: (1) Auswahl und Nutzung von Medienangeboten, (2) kreative Gestaltung und Verbreitung eigener Medien, (3) Verständnis und Bewertung von Mediengestaltungen, (4) Identifikation und Reflexion von Medieneinflüssen sowie (5) Anerkennung und Beurteilung der Bedingungen für die Medienproduktion und -verbreitung (MENFP, 2011). Bisher konzentrierten sich diese fünf Kompetenzen auf die Informations- und Kommunikationstechnologie (IKT) Anwendungskompetenz und Medienbildung. Allerdings hat das Bildungsministerium in den Unterlagen zum Schuljahr 2019/20



angekündigt, dass zusätzlich zur IKT-Anwendungskompetenz und Medienbildung auch die Informatikbildung in die Lehrpläne integriert werden soll (Reuter, 2020).

Der luxemburgische Medienkompass wurde seit 2017 vom SCRIPT entwickelt, basierend auf dem Referenzrahmen „DigComp“ (Version „DigComp 2.1“ von 2018). Er wurde an den luxemburgischen Kontext angepasst und soll zusammen mit unterstützenden Ressourcen, Unterrichtsmaterialien, einem Medienpass und Lehrerfortbildungen Orientierung und Unterstützung für Lehrkräfte bieten. Dies umfasst nicht nur technische Fertigkeiten, sondern auch die Fähigkeit, zugrunde liegende Prinzipien kritisch zu verstehen und zu reflektieren (Reuter, 2020). Der Medienkompetenzrahmen umfasst insgesamt 16 Kompetenzen, die in fünf übergeordnete Bereiche unterteilt sind: (1) Informationen und Daten, (2) Kommunikation und Zusammenarbeit, (3) Erstellen von Inhalten, (4) Datenschutz und Sicherheit und (5) Die digitale Welt. Zusätzlich wird besonderer Wert auf die „4K“-Kompetenzen gelegt: Kommunikation, Kollaboration, Kreativität und kritisches Denken (SCRIPT, 2022). Im dritten Kompetenzbereich des Medienkompasses wird das konzeptionelle Denken des CT in der Teilkompetenz „3.4 Modellieren, strukturieren, kodieren“ behandelt. Diese Begriffe betonen die proaktive Vorgehensweise und verdeutlichen, dass CT weniger auf reines Kodieren abzielt, sondern auch auf Problemlösungsfähigkeiten, die in verschiedenen Bereichen Anwendung finden und lange vor dem eigentlichen Programmieren beginnen (Reuter, 2020). Dabei werden vier Etappen hervorgehoben: Problemanalyse, Planung, Umsetzung und Überprüfung (SCRIPT, 2022).

### **2.2.3. Bisher identifizierte Herausforderungen**

Da informatisches und algorithmisches Denken eine neue Kompetenz im luxemburgischen Schulkontext ist, fehlen Umsetzungsstudien. Ein bereits identifiziertes Problem in der Studie von Hennico (2021) ist die Unklarheit des Lehrpersonals über die Integration von CT im Unterricht. Obwohl 85,4% der Lehrkräfte Interesse an der Integration zeigten, konnten nur 41,9% es

tatsächlich umsetzen. Insgesamt gaben 68,8% an, CT nicht in ihrem Unterricht zu verwenden, dies aber nach einer entsprechenden Schulung in Erwägung zu ziehen. 22,6% zweifelten an ihrem Selbstvertrauen in diesem Bereich und 35,5% fühlten sich nicht in der Lage, CT umzusetzen. Die Mehrheit (93,5%) war jedoch bereit, sich weiterzubilden (Hennico, 2021).

Luxemburg stellt außerdem einen besonderen Fall dar, da das luxemburgische Bildungsministerium im Gegensatz zu vielen anderen Ländern aus politischen Gründen bei offiziellen Pressekonferenzen das Konzept des CT unter dem Namen „Coding“ eingeführt hat. Dies bringt viel Verwirrung auf, da der Begriff ‚Kodéiren‘ (Kodieren) fälschlicherweise mit dem Begriff ‚Computational Thinking‘ in Luxemburg gleichgesetzt wird. Obwohl die Bildungsakteure in den Medien bemüht sind, diese ungünstige Namensgebung zu klären, scheint das Missverständnis bestehen geblieben zu sein. In der Tat assoziierten 19,5% der Lehrkräfte CT mit digitaler Mediennutzung oder Programmierung, während 17,5% die Frage unbeantwortet ließen. Eine weitere Klärung des Begriffs und die Förderung eines präzisen Sprachgebrauchs sind also nötig (Hennico, 2021).

### **2.3. ISTE Essential Conditions**

Die International Society for Technology in Education (ISTE) hat 14 „Essential Conditions“ formuliert, um die optimale Nutzung von Technologie im schulischen Kontext zu unterstützen (Crompton, 2017). Diese Bedingungen sind entscheidend für die effektive Nutzung von Bildungstechnologie zur Unterstützung des Lernens und bieten einen forschungsbasierten Rahmen für Pädagogen und Schulleitungen, um ISTE-Standards zu implementieren und systemweite Veränderungen herbeizuführen (ISTE, 2023). Die 14 Essential Conditions umfassen: „shared vision, empowered leaders, implementation planning, consistent and adequate funding, equitable access, skilled personnel, ongoing professional learning, technical support, curriculum framework, student-centered learning, assessment and evaluation,

engaged communities, support policies, supportive external context“ (Crompton, 2017, S.1-2).

Im Jahr 2000 wurde die erste Gruppe von 10 Essential Conditions von Schul- und Bezirksadministratoren in Zusammenarbeit mit der ISTE-Fakultät entwickelt. 2007 wurden diese Bedingungen von Tausenden an Pädagogen überarbeitet und die endgültigen 14 Essential Conditions 2009 veröffentlicht (Crompton et al., 2016). Diese Bedingungen bieten einen anerkannten Leitfaden für die Planung und Messung des Fortschritts bei der Integration von Technologie im Unterricht.

Die 14 Essential Conditions bilden die grundlegende Struktur dieser Bachelorarbeit. Obwohl sie ursprünglich für die Technologieintegration entwickelt wurden, dienen sie als flexibles Rahmenwerk, das auf andere pädagogische Kontexte übertragbar ist. In dieser Arbeit wurden sie spezifisch an den CT-Kontext angepasst und dienen als Analysegerüst, um systematisch die förderlichen und hinderlichen Faktoren bei der Integration von CT in die K-12-Bildung zu identifizieren.

## **3. Methode**

### **3.1. Forschungsansatz**

Das Ziel dieser Forschungsarbeit ist eine systematische Auswertung von Daten aus verlässlichen Quellen zur Integration von CT in die K-12 Bildung. Die zentrale Frage lautet: „Welche Faktoren fördern oder behindern die erfolgreiche Implementierung von Computational Thinking als Schulfach in Grundschulen?“. Die Klärung dieser Frage ist entscheidend, um Schlussfolgerungen zu ziehen und zur Optimierung der CT-Integration in den Lehrplänen von Bildungseinrichtungen beizutragen. Die Ergebnisse können Schulen und Bildungseinrichtungen in Luxemburg, sowie auch im Ausland dienen. Indem die Erfolgsfaktoren identifiziert werden, können konkrete Empfehlungen formuliert werden, um CT effektiv in den Unterricht zu implementieren und die Vorteile dieses Bildungsansatzes bestmöglich zu nutzen. Gleichzeitig ermöglicht die Analyse potenzieller Hindernisse, gezielt Strategien zu entwickeln, um Herausforderungen zu bewältigen und Ressourcen optimal einzusetzen. Dies erhöht die Chancen auf eine erfolgreiche und reibungslose Integration von CT in den Bildungsalltag. Diese umfassende Untersuchung trägt somit wesentlich zur Modernisierung und Verbesserung des Bildungssystems bei, indem sie praxisorientierte Lösungen und Ansätze für eine zeitgemäße Bildung liefert.

### **3.2. Methoden der Datenanalyse**

Für die Analyse der förderlichen und hinderlichen Faktoren im Kontext der Integration von CT in die Grundschulbildung wurde sich an den 14 Essential Conditions von ISTE inspiriert. Obwohl die Essential Conditions ursprünglich für die effektive Implementierung und Integration von Technologie in Bildungseinrichtungen entwickelt wurden, erweisen sie sich als flexibler Ansatz, der weit über den Bereich der Bildungstechnologien hinausgeht. Hier ist es wichtig zu betonen, dass CT nicht als Bildungstechnologie betrachtet wird, sondern als umfassendes, interdisziplinäres Kompetenzfeld. Die

Anwendung von CT-Prinzipien kann zwar durch Technologien erfolgen, dies ist jedoch nicht zwingend erforderlich. Diesbezüglich wurden die ursprünglichen Beschreibungen der 14 Bedingungen konkret an den CT-Kontext adaptiert und leicht umgeschrieben.

Das Framework bietet einen abstrakten und allgemeinen Rahmen, der für weitreichende systemische Anpassungen in verschiedenen pädagogischen Kontexten anwendbar ist. Die Bedingungen unterstützen eine breite Palette pädagogischer Veränderungen und Verbesserungen, indem sie einen Leitfaden für die Planung, Umsetzung und Bewertung verschiedener Bildungsprojekte bereitstellen. Dabei werden zahlreiche interne und externe Faktoren sowie lokale bis internationale Aspekte berücksichtigt, die sowohl die schulinterne Organisation und unmittelbare Umsetzung als auch die externe Unterstützung und Maßnahmen umfassen. Durch diese umfassende Herangehensweise und den vielseitigen, flexiblen Ansatz kann die interdisziplinäre Natur von CT in einem breiten, interaktiven pädagogischen Kontext berücksichtigt werden. Dies ermöglicht eine ganzheitliche und systematische Betrachtung der CT-Integration und fördert eine tiefgreifende und nachhaltige Analyse im Kontext dieser Arbeit. Im Folgenden werden die einzelnen Bedingungen detaillierter erklärt (Crompton, 2017):

1. **Gemeinsame Vision:** Proaktive Führung bei der Entwicklung einer gemeinsamen Vision für CT unter allen Bildungsbeteiligten, einschließlich Lehrern und Unterstützungspersonal, Schul- und Bezirksadministratoren, Lehrerbildnern, Schülern, Eltern und der Gemeinschaft.
2. **Befähigte Führungskräfte:** Stakeholder auf allen Ebenen werden befähigt, Führungskräfte zu sein und Veränderungen im Bereich CT zu bewirken.
3. **Umsetzungsplanung:** Ein systematischer Plan, der auf eine gemeinsame Vision für die Effektivität der Schule und das Lernen der Schüler durch die Integration von CT und (digitalen) Lernressourcen ausgerichtet ist.

4. Konstante und angemessene Finanzierung: Laufende Finanzierung zur Unterstützung der Infrastruktur, des Personals, der (digitalen) Ressourcen und der Personalentwicklung im Bereich CT.
5. Gleicher Zugang: Robuster und zuverlässiger Zugang zu aktuellen und aufkommenden Technologien und (digitalen) Ressourcen für CT, mit Konnektivität für alle Schüler, Lehrer, Mitarbeiter und Schulführer.
6. Qualifiziertes Personal: Pädagogen, Unterstützungspersonal und andere Führungskräfte sind qualifiziert in der Auswahl und dem effektiven Einsatz geeigneter Ressourcen für CT.
7. Laufende berufliche Weiterbildung: Berufliche Weiterbildungspläne und -möglichkeiten im Bereich CT mit festgelegter Zeit zum Üben und Teilen von Ideen.
8. Technischer Support: Konstante und zuverlässige Unterstützung für die Wartung, Erneuerung und Nutzung von Ressourcen für CT und (digitale) Lernressourcen.
9. Lehrplan-Rahmen: Inhaltsstandards und zugehörige (digitale) Lehrplanressourcen, die mit dem Lernen und Arbeiten im Bereich CT übereinstimmen und dieses unterstützen.
10. Schülerzentriertes Lernen: Planung, Unterricht und Bewertung, die auf die Bedürfnisse und Fähigkeiten der Schüler im Bereich CT ausgerichtet sind.
11. Bewertung und Evaluierung: Kontinuierliche Bewertung von Lehre, Lernen und Führung sowie Bewertung des Einsatzes von CT und (digitalen) Ressourcen.
12. Engagierte Gemeinschaften: Partnerschaften und Zusammenarbeit innerhalb der Gemeinschaften zur Unterstützung und Finanzierung des Einsatzes von CT und (digitalen) Lernressourcen.

13. Unterstützende Richtlinien: Richtlinien, Finanzpläne, Rechenschaftsmaßnahmen und Anreizstrukturen zur Unterstützung des Einsatzes von CT und anderen (digitalen) Ressourcen für das Lernen und den Betrieb von Bezirksschulen.

14. Unterstützender externer Kontext: Politiken und Initiativen auf nationaler, regionaler und lokaler Ebene zur Unterstützung von Schulen und Lehrerbildungsprogrammen bei der effektiven Umsetzung von CT zur Erreichung von CT-Standards.

Für jede dieser Bedingungen werden die gefundenen hinderlichen und förderlichen Aspekte notiert.

### **3.3. Analyseschritte**

In dieser Bachelorarbeit wurde das Ziel verfolgt, ein umfassendes Verständnis der förderlichen und hinderlichen Aspekte der Integration von CT im Bildungskontext zu entwickeln. Eine systematische Herangehensweise wurde angewandt, indem gezielt relevante Literatur für die 14 Essential Conditions gesucht wurde. Die Literatur umfasste eine breite Auswahl an empirischen Studien, Forschungsarbeiten, Berichten, lehrerrzentrierter Forschung und Literaturübersichten, die verschiedene Aspekte von CT beleuchten und diverse Erfolgs- sowie Hindernisfaktoren identifizieren. Der Fokus lag darauf, hauptsächlich wissenschaftliche, primäre, empirische, peer reviewed und aktuelle Quellen zu verwenden. Die Suchstrategie für wissenschaftliche Studien zur Integration von CT im K-12 Bildungsbereich basierte auf der gezielten Kombination präziser Schlagwörter, wie „Computational Thinking“ in Verbindung mit „K-12“, „implementation“, „effective and hindering factors“, „barriers“, „curriculum integration“, „teacher training“, „unplugged“, „inclusion“ und „assessment“. Hierbei kamen renommierte Suchmaschinen wie Google Scholar, ERIC und a-z.lu zum Einsatz. Dazu wurden auch Literaturverzeichnisse bereits gefundener Studien berücksichtigt und als Inspirationsquelle für weitere relevante Quellen genutzt. Durch diese

erweiterte Herangehensweise konnte sichergestellt werden, dass auch Studien, die möglicherweise nicht unmittelbar über die initialen Suchbegriffe gefunden wurden, in die systematische Auswertung einbezogen werden konnten. Zudem wurde auf die geografische Diversität der Studien geachtet, wobei Quellen aus verschiedenen Kontinenten einbezogen wurden, insbesondere Europa, Nordamerika und Asien. Dies ermöglichte, unterschiedliche Bildungsansätze, kulturelle Kontexte und Implementierungsstrategien zu untersuchen und zu vergleichen.

Anschließend wurden die gesammelten Daten analysiert und unter die entsprechende Bedingung geordnet. Jede der 14 Bedingungen wurde in Erfolgsfaktoren und Hindernisse unterteilt, die weiter in thematische Unterkategorien gegliedert wurden. Wo möglich, wurden konkrete Beispiele herangezogen, um die Forschungsergebnisse zu veranschaulichen und zu präzisieren. Diese Methodik ermöglichte eine strukturierte und systematische Übersicht, was den Vergleich und kritische Bewertung erheblich erleichterte. Basierend darauf konnten gezielte Schlussfolgerungen und Empfehlungen zur Optimierung der Einführung von CT in Grundschulen gezogen werden. Empfehlungen von anderen Autoren wurden im Analyseteil vermieden, da sie aufgrund der begrenzten Validität der Daten nicht ausreichend unterstützt werden konnten.



## **4. Forschungsergebnisse**

Im Folgenden werden die Ergebnisse der Forschung im Rahmen der 14 Bedingungen präsentiert. Jede der 14 Bedingungen wird einzeln eingeführt und die Hauptergebnisse dazu kurz erläutert. Im Anschluss werden zuerst die identifizierten Erfolgsfaktoren detailliert in den jeweiligen thematischen Unterkategorien erklärt, gefolgt von den identifizierten Hindernissen. Ziel dieser Präsentation ist es, ein umfassendes und strukturiertes Bild der förderlichen und hinderlichen Faktoren zu vermitteln, die die Integration von CT in die Grundschulbildung beeinflussen.

### **4.1. Gemeinsame Vision**

Die erste Bedingung analysiert die Rolle einer gemeinsamen CT-Vision. Die Analyse zeigt, dass die Entwicklung eines gemeinsamen Verständnisses entscheidend für den Erfolg der CT-Bildung ist, sie jedoch vor Herausforderungen durch unterschiedliche CT-Begrifflichkeiten und den variierenden Status der CT-Bildungspolitik weltweit steht.

#### **4.1.1. Erfolgsfaktoren**

##### **4.1.1.1. Gemeinsames Verständnis**

Balanskat et al. (2018) betonen die Bedeutung der Erarbeitung eines umfassenden Schulkonzepts, das eine gemeinsame Vision und ein einheitliches Verständnis innerhalb der Schule beinhaltet. Dabei ist das Engagement von Lehrkräften, Schulleitungen, Bildungseinrichtungen und politischen Entscheidungsträgern entscheidend, um Veränderungen voranzutreiben und sicherzustellen, dass CT/Programmierung ein fester Bestandteil des Curriculums wird, der in verschiedenen Unterrichtskontexten umgesetzt wird. Dazu stellten Bocconi et al. (2022) in ihrer Mehrfachfallstudie fest, dass konkrete Unterrichtsbeispiele zur Schaffung eines gemeinsamen Verständnisses zwischen Lehrern, politischen Entscheidungsträgern und anderen

Interessengruppen beitragen können. Dies kann wiederum die mangelnde Einigkeit über eine gemeinsame CT-Definition adressieren.

#### **4.1.2. Hindernisse**

##### **4.1.2.1. Diverse CT-Begrifflichkeiten und Bedeutungen**

Verschiedene Länder und Regionen benutzen unterschiedliche Begriffe, um CT-Konzepte und -Praktiken zu beschreiben (Hsu et al., 2019). Die Verwendung des Begriffs ‚Computational Thinking‘ in Bildungspolitiken und Lehrplänen variiert international. Während manche Länder wie Dänemark keinen spezifischen Ausdruck für CT haben (Bocconi et al., 2018), verwenden andere Länder wie Finnland und Norwegen ‚algorithmic thinking‘ und Estland ‚technological literacy‘ (Information Technology Foundation for Education [HITSA], 2014). In einigen Ländern, wie England und Polen, wird CT oft mit Informatik und IKT verbunden (Syslo & Kwiatkowska, 2015). Politiken in Finnland, betonen eher Programmierung als CT (Bocconi et al., 2018), während in Hongkong meist Programmierung und systematisches Problemlösen im Fokus stehen (The Curriculum Development Council, 2017). Moderne Initiativen und Bildungsrichtlinien nutzen zunehmend den Begriff ‚Coding‘ (z.B. United Arab Emirates Ministry of Education, 2015). Diese unterschiedlichen Begrifflichkeiten zwischen den Ländern spiegeln die vielschichtigen politischen Diskussionen und Maßnahmen wider, die auf verschiedenen Ebenen in diesem Bereich stattfinden.

Bocconi et al. (2018) weisen auf die erhebliche Komplexität von CT hin, die das Erreichen eines gemeinsamen Verständnisses erschweren und dadurch die Umsetzungspläne beeinträchtigen könnte. Folglich plädieren Selby und Woollard (2014) dafür, eine robuste Definition von CT zu entwickeln. Dabei ist es ebenso wichtig, klar zu definieren, was CT nicht ist, wie die endgültige Definition selbst. Interessanterweise argumentieren Voogt et al. (2015), dass es vorteilhafter ist, Ähnlichkeiten und Beziehungen in den Diskussionen über CT herauszustellen, anstatt eine ultimative Definition zu suchen. Durch die Identifizierung dieser Ähnlichkeiten und Beziehungen kann eine

präzisere Beschreibung dessen erfolgen, was bei CT wichtig ist und wie es in den K-12-Bereich integriert werden kann. Ebenso argumentiert Guzdial (2011), dass es weniger darauf ankommt, eine exakte und strenge Definition von CT zu haben. Vielmehr legt er den Fokus auf konkrete Implikationen im Bildungskontext, wie das Lehren und Messen des informatischen und algorithmischen Denkens.

#### **4.1.2.2. Diverser Status von CT Educational Policy Initiatives**

Die weltweite Implementierung der Bildungspolitik für CT variiert stark und reflektiert die unterschiedlichen politischen, kulturellen und bildungsspezifischen Gegebenheiten der jeweiligen Länder. In Ländern wie England und vielen europäischen Staaten ist CT seit mehreren Jahren ein obligatorischer Teil des Lehrplans. In asiatischen Nationen wie Japan, Südkorea und China wird CT als Schlüsselkompetenz des 21. Jahrhunderts gefördert und als Motor für technologische sowie wirtschaftliche Fortschritte betrachtet. Entwicklungsländer in Lateinamerika und Afrika hinken jedoch hinterher, da CT dort nicht in die öffentlichen politischen Diskussionen integriert ist und hauptsächlich durch universitäre Initiativen und Stiftungen vorangetrieben wird. Dies führt zu einem technologischen und wirtschaftlichen Rückstand gegenüber Ländern, die CT aktiv fördern, und könnte die sozioökonomischen Unterschiede zwischen entwickelten und weniger entwickelten Ländern weiter verschärfen (Belmar, 2022).

## **4.2. Befähigte Führungskräfte**

Die zweite Bedingung analysiert die Befähigung von Führungskräften im CT-Wandel. Die Ergebnisse verdeutlichen, dass das Einbeziehen von Bildungsakteuren in den Prozess der Implementierung und eine professionelle Entwicklung der Schulleiter vorteilhaft sind. Jedoch wird die Befähigung von Führungskräften zur Förderung der CT-Bildung durch mangelnde Priorisierung und eine geringe Akzeptanz unter wichtigen Interessengruppen gehemmt.

## **4.2.1. Erfolgsfaktoren**

### **4.2.1.1. Einbeziehung der Beteiligten**

Nach Toyama (2015) hängt der Erfolg der informatischen Bildung von der Leidenschaft aller Beteiligten ab, einschließlich Führungskräften wie Gouverneuren oder Direktoren, Lehrern, Schülern und Eltern. Es wird somit empfohlen, alle Bildungsbeteiligten aktiv in den Prozess der Implementierung von CT einzubinden (Bocconi et al., 2022; Toyama, 2015). Sowohl Lehrkräfte als auch Lernende sollen ihre Erfahrungen im Bereich der Programmierung teilen können (Egbert et al., 2021).

Lehrkräfte spielen dabei eine zentrale Rolle, denn der Erfolg von Bildungsreformen hängt maßgeblich davon ab, wie effektiv Lehrpersonen diese Änderungen im Unterricht anwenden. Es ist unerlässlich, das Engagement, die Motivation und die Bedenken der Lehrkräfte zu berücksichtigen, um eine erfolgreiche Implementierung zu gewährleisten. Die Bedürfnisse der Lehrkräfte müssen erkannt und angehört werden, um adäquate pädagogische Unterstützung zu leisten, anstatt eine undurchführbare Top-Down-Anweisung zur Durchsetzung des Informatikunterrichts landesweit vorzugeben (Kim & Kim, 2018). Dies ist besonders wichtig, da das Engagement der Lehrkräfte direkte Auswirkungen auf verschiedene Aspekte hat, von der Rekrutierung bis zur Umsetzung (Li, 2020).

### **4.2.1.2. Professionelle Entwicklung der Schulleiter**

Ein spezielles Schulungsprogramm für Schulleiter und Schulräte ist essenziell, da deren Unterstützung direkt die Umsetzung und Nachhaltigkeit der Informatikbildung beeinflusst. Schulleiter spielen eine Schlüsselrolle bei der Schaffung eines förderlichen Umfelds in ihren Schulen, das Innovationen in der Informatikbildung unterstützt (Kim & Kim, 2018). Laut einer Studie von Choi et al. (2016) verbessert sich die Einstellung der Schulleiter zur Informatikbildung durch Ausbildungskurse zunehmend, was wiederum das Bildungssystem positiv beeinflusst. Schulleiter, die von der Wichtigkeit der

Informatikbildung überzeugt sind, sorgen dafür, dass Ressourcen und Unterstützung für Informatiklehrkräfte verfügbar sind. Auch laut Kim und Kim (2018) sind Schulleiter, die an diesen Weiterbildungsmaßnahmen teilnehmen, oft engagierte Befürworter der Informatikbildung und fördern die professionelle Entwicklung ihrer Lehrkräfte in diesem Bereich. Dies ist in Korea besonders prägnant, wo Schulleiter höchste Autorität und Anerkennung erhalten.

## **4.2.2. Hindernisse**

### **4.2.2.1. Geringes Verständnis und mangelnde Priorisierung**

Ein wesentliches Hindernis, welches in der Fallstudie von Bocconi et al. (2022) identifiziert wurde, ist die mangelnde Priorisierung der CS<sup>3</sup>-Bildung durch Schulleitungen und Führungsteams. Laut einer Studie von Google aus dem Jahr 2015 gaben lediglich 18% der Verwaltungspersonen an, dass Informatik in den Klassen 1 bis 6 an ihren Schulen oberste Priorität habe. Dies äußert sich oft in unzureichenden Zeitplänen, Budgets und Ressourcen für die effektive Implementierung und Skalierung von CT-Bildungsinitiativen. Kim und Kim (2018) identifizierten diese mangelnde Priorisierung ebenfalls unter fachfremden Lehrkräften.

Die Herausforderung, Akzeptanz unter wichtigen Interessengruppen wie Fakultätsmitgliedern, Lehrern und Verwaltungspersonal für die Implementierung von CT in Lehrerausbildungsprogrammen zu gewinnen, wurde in den Experteninterviews von Li (2020) deutlich. Sowohl Lehrer als auch Verwaltungspersonal waren oft unzureichend motiviert. Dabei ist es schwierig, genügend Anreize für jene zu bieten, die von Anfang an nicht an CT interessiert sind, und es gibt nur wenige Möglichkeiten für Bestrafungen (Li, 2020). Staatliche Entscheidungsträger stehen vor der Herausforderung, dass sie aufgrund ihrer Distanz zum schulischen Alltag oft nicht in der Lage sind,

---

<sup>3</sup> Computer Science

detaillierte und praxisnahe Implementierungsempfehlungen zu geben, die Lehrkräfte für ihre Arbeit benötigen (Mills et al., 2021). Nach Bocconi et al. (2022) liegt die mangelnde Priorität daran, dass sich Schulleitungen und Führungsteams um zahlreiche andere Anliegen kümmern müssen.

### **4.3. Umsetzungsplanung**

Die dritte Bedingung analysiert die systematische Umsetzungsplanung von CT, worauf auf allgemeine und spezifische Umsetzungsansätze eingegangen wird. Allgemein wird die Wichtigkeit einer frühkindlichen und schrittweisen Vermittlung von CT-Fähigkeiten, sowie der Erfolg von Lerngerüsten festgestellt. Spezifische Umsetzungsansätze schließen Programmierung, Bildungsrobotik, Lernspiele, unplugged Aktivitäten, sowie die TIPP&SEE Methode ein, wobei auch die Einführung in STEAM<sup>4</sup>- und Informatikfächer vorteilhaft ist. Jedoch können andere Aspekte wie eine zu lange Dauer von CT-Aktivitäten, ein überfüllter Klassenraum oder technische Probleme die Umsetzung behindern.

#### **4.3.1. Erfolgsfaktoren**

##### **4.3.1.1. Integrationsansätze**

Die Einbindung von CT/Programmierung in den Lehrplan wird sowohl durch die spezifischen Ziele ihrer Integration als auch durch praktische Überlegungen bestimmt (Balanskat et al., 2018). Bocconi et al. (2022) offenbarten in ihrer Analyse verschiedene Ansätze zur Vermittlung von CT-Fähigkeiten. In der Grundschulbildung wird CT über drei Hauptansätze in die Lehrpläne integriert: fächerübergreifend, separat oder innerhalb anderer Fächer. In der Grundschule, wo Lehrer mehrere Fächer unterrichten, bietet die fächerübergreifende Integration von CT-Fähigkeiten erhebliche Vorteile. CT-Konzepte können nahtlos in verschiedene Fächer eingebettet werden,

---

<sup>4</sup> science, technology, engineering, arts, mathematics

was das Verständnis und die Relevanz für die Schülerinnen und Schüler erhöht. Dabei ermöglicht ein gemischter Ansatz, wo CT als fächerübergreifendes Thema in den frühen Klassen (1 und 2) und als eigenständiges Fach ab der dritten Klasse eingeführt wird, eine schrittweise Einführung und Anwendung von CT-Fähigkeiten in verschiedenen Disziplinen. Wichtig zu beachten ist die klare Zuweisung von Verantwortlichkeiten an die Lehrer, um sicherzustellen, dass informatische Konzepte effektiv in den Lehrplan integriert werden (Bocconi et al., 2021).

#### **4.3.1.2. Frühkindliche Vermittlung**

Die Entwicklung von CT-Fähigkeiten entstehen bei Kindern nicht zufällig während der Grundschulbildung, sondern erfordern gezielte Anweisungen und pädagogische Maßnahmen zur Förderung (Rodríguez-Martínez et al., 2020). Forscher sind sich einig, dass das frühzeitige Erlernen von CT einen wichtigen Beitrag zum Fähigkeiten- und Entwicklungsstand von Kindern darstellt (Grover & Pea, 2018). Ein früher Beginn des Informatikunterrichts kann die Entwicklung von CT-Fähigkeiten signifikant fördern, da sie die Motivation zum Erlernen des Programmierens der Schülerinnen und Schüler steigern und eine positive Einstellung zur Informatik unterstützen (Tran, 2019). Dabei soll die Informatikausbildung bereits in der frühen Grundschule zwischen 5 und 7 Jahren beginnen und in der Sekundarschule und darüber hinaus fortgeführt werden (Arfé et al., 2020). Jedoch mangelt es an klaren Richtlinien darüber, in welchem Alter Interventionen zur Förderung der CT-Fähigkeiten angemessen sind (Rijke et al., 2018). Forschung in frühkindlichen Bildungseinrichtungen mit Kibo-Robotik-Kits zeigte jedoch, dass junge Lernende von 3 bis 5 Jahren grundlegende CT-Fähigkeiten wie das Erstellen von Sequenz, Wiederholungen, Bedingungen und Fehlerbehebung erwerben konnten (Bers et al., 2019). Die Studie von Saxena et al. (2020) zeigte, dass fast alle Kinder der Klassen K1 bis K3 (3 bis 6 Jahre) nach 10 Stunden CT-Training die Fähigkeiten zur Mustererkennung und Sequenzierung beherrschten. Aufgaben zur Algorithmusgestaltung konnten die K1-Schüler jedoch nur teilweise abschließen, während

die anderen Kinder in der Regel das angestrebte Leistungsniveau erreichen. Schlussendlich ist die Forschung, die sich auf junge Altersgruppen konzentriert, immer noch zu begrenzt, um gültige Schlussfolgerungen ziehen zu können (Arfé et al., 2020).

#### **4.3.1.3. Altersgerechte und schrittweiser Aufbau**

Eine altersgerechte und schrittweise Einführung von CT im Bildungssystem ist von grundlegender Bedeutung. Die CT-Aufgaben sollten dabei dem Alter der Kinder entsprechen und weder zu schwierig noch zu einfach sein (Rijke et al., 2018). Um dies zu gewährleisten, müssen erreichbare und angemessene Kompetenzniveaus für junge Schüler bekannt sein (Moore et al., 2020). Schülerinnen und Schüler verschiedener Altersgruppen zeigen nämlich unterschiedliche Niveaus an CT-Fähigkeiten auf (Moore et al., 2020; Rijke et al., 2018). Mit zunehmendem Alter werden die Kinder geschickter und bekommen höhere Fähigkeitsniveaus in den Bereichen des CT-Denkens, der Selbstwirksamkeit beim Programmieren und des reflektierenden Denkens (Durak et al., 2019). Auch Rijke et al. (2018) weisen eine positive Korrelation zwischen Alter und Abstraktionsvermögen hin. Hier ist es jedoch wichtig zu betonen, dass das logische Verständnis von Kindern eine auf Erfahrung (Vygotsky, 1994) und Biologie (Bjorklund, 2018) basierende Fähigkeit ist und nicht nur eine Frage des Alters (Piaget, 1952).

#### **4.3.1.4. Lerngerüst/ Scaffolding**

Bezüglich Lerngerüsten (Scaffolding) zeigten Xu et al. (2023), dass das von Lehrern unterstützte Lerngerüst sehr effektiv zur CT-Förderung ist. Dieses zielt darauf ab, Problemlösungsfähigkeiten und Denkvermögen zu fördern, die ohne effektive Lernunterstützung nicht möglich sind. Beispielsweise förderten Lerngerüste die Kooperation und Beteiligung der Schülerinnen und Schüler, erleichterten das Problemlösen und verbesserten das höherstufige Denken (Reiser, 2004, zitiert nach Xu et al., 2023, S.6638). Zudem kann ein Tutor den Lernprozess der Kinder durch ein interaktives Austauschsystem unterstützen, indem er ihre Aufmerksamkeit lenkt, die Aufgabe



vereinfacht, den Problemlöseprozess steuert, wichtige Aspekte hervorhebt, Frustration mindert und Lösungen demonstriert (Wood et al., 2006). Basu et al. (2017) fanden heraus, dass Schülerinnen und Schüler, die adaptives Scaffolding erhielten, bessere Ergebnisse bei der Modellierungsleistung und -strategie sowie beim Lernen von Naturwissenschaften und CT erzielten als ihre Mitschüler. Schließlich ist das von Lehrern unterstützte Lerngerüst besonders zielgerichtet, informativ und zeitnah, weshalb es den größten Einfluss auf CT hat (Xu et al., 2023).

#### **4.3.1.5. Gezielte Umsetzungsansätze**

##### **4.3.1.5.1. Programmierung**

Obwohl Programmierung ein wesentlicher Bestandteil von CT ist, geht CT über das reine Programmieren hinaus (Resnick et al., 2009). Programmieren dient jedoch als Grundlage für die Entwicklung von CT-Fähigkeiten, die auf andere Wissensbereiche, einschließlich der STEAM-Fächer, übertragbar sind (Rojas & García, 2020). Nach Moreno-Léon et al. (2018) gilt Programmieren als effektivste Methode, um CT-Fähigkeiten zu erlernen (Moreno-Léon et al., 2018). Der Programmierunterricht kann die Entwicklung von CT bei K-12-Schülern signifikant fördern, insbesondere durch strukturierte und problemorientierte Ansätze (Xu et al., 2023). Es fördert kreatives Denken, kritisches Denken und Problemlösungsfähigkeiten (Wong & Cheung, 2018). Studien zeigen, dass visuelle Blockprogrammiersprachen besonders effektiv in der Förderung von CT sind (Chiazzese et al., 2018; Merino-Armero et al., 2022). Das Verständnis von Konzepten durch einen aktiven Ansatz, projektbasiertes Lernen, Nutzen, Motivation und Engagement unterstreicht die Bedeutung und Wirksamkeit der Implementierung einer visuellen Programmiersprache in der Grundschule (Sáez-López et al., 2016). Auch eine Mischung aus textbasierter und blockbasierter Programmierung verbessert die CT-Fähigkeiten signifikant und steigert das Selbstbewusstsein sowie die Freude am Programmieren (Deng et al., 2019).

##### **4.3.1.5.2. STEAM-Fächer und Informatik**

Die Einführung von CT zeigt besonders in den Bereichen der STEAM-Bildung (Wissenschaft, Technologie, Ingenieurwesen, Kunst und Mathematik) sowie der Informatik ein großes Potenzial (Merino-Armero et al., 2022). STEAM steht für eine interdisziplinäre Bildungsstrategie, die darauf abzielt, Schüler in den genannten Fächern zu fördern und kreative Problemlösungsfähigkeiten zu entwickeln. Mehrere Autoren (Henderson et al., 2007; Swaid, 2015, zitiert nach Merino-Armero et al., 2022, S.426) betonen die Notwendigkeit, CT in diesen Bereichen zu integrieren, um die Forschung und das Verständnis weiter voranzutreiben. Interessant ist dabei, dass die Informatik das Verständnis der Wissenschaft fördert und umgekehrt (Goodman & Dekhtyar, 2014, zitiert nach Merino-Armero et al., 2022, S.426). Diese wechselseitige Beziehung bildet einen hervorragenden Ausgangspunkt, um CT in Wissenschafts- und STEAM-Fächern zu implementieren. Darüber hinaus bieten Informatikunterricht und -kurse die notwendigen Werkzeuge und Ressourcen, um CT effektiv zu entwickeln und zu fördern (Merino-Armero et al., 2022).

#### **4.3.1.5.3. Bildungsrobotik**

Zu lernen, wie man programmiert, ist ein in hohem Maße konzeptueller Prozess. Die Untersuchungen von Min und Kim (2020) zeigten, dass physischer Computing-Unterricht mit Robotern die CT-Konzepte bei Schülerinnen und Schülern durch praktische Anwendungen konkretisiert und ihre Perspektiven in diesem Bereich durch die Integration realer Situationen erweitert werden. Auch andere Forschungen belegten, dass Bildungsrobotik im Programmierunterricht als geeignetes Mittel zur Entwicklung von CT-Fähigkeiten dienen kann (Ardito et al., 2020; Hsiao et al., 2019; Noh & Lee, 2020; Shen et al., 2020, zitiert nach Lee et al., 2022, S.7). Chiazzese et al. (2019) identifizierten die Wirksamkeit der Robotik bei der Förderung kognitiver Fähigkeiten im CT-Bereich, wobei konkret der Einsatz der Robotik-Kits wie WeDo 2.0 half.

Interessanterweise zeigten Merino-Armero et al. (2022) jedoch, dass Robotikkurse nur eine geringe Wirkung auf die Entwicklung von CT-

Fähigkeiten haben. Allerdings steigt ihre Effektivität deutlich, wenn sie im Vorschulalter eingesetzt werden.

#### **4.3.1.5.4. Lernspiele**

Forschung zeigte, dass das Entwerfen und Spielen von Bildungsspielen ein wichtiger Erfolgsfaktor für die Förderung der Entwicklung von CT-Fähigkeiten bei Schülern ist, besonders bei jüngeren Lernenden (Allsop, 2019; Altanis et al., 2018; Chiazese et al., 2018; Garneli & Chorianopoulos, 2018; Gomes et al., 2018; Zhao & Shute, 2019, zitiert nach Lee et al., 2022, S.8). Diese Aktivitäten steigern nicht nur die Motivation der Schülerinnen und Schüler, mehr über CT zu lernen, sondern verbessern auch ihr Verständnis geometrischer Prinzipien und ihre sozialen Fähigkeiten (Altanis et al., 2018). Darüber hinaus zeigten Schülerinnen und Schüler, die an der Entwicklung von Spielen mit Scratch und Alice beteiligt waren, signifikante Fortschritte in ihrer Fähigkeit zur Anwendung von Programmierkonzepten und setzten metakognitive Fähigkeiten wie Planungs-, Überwachungs- und Bewertungsstrategien ein. Zudem demonstrierten sie Fähigkeiten in Zusammenarbeit, Kommunikation, Ausdauer, Problemlösung und Kreativität (Allsop, 2019).

#### **4.3.1.5.5. Unplugged Aktivitäten**

In einer Meta-Analyse von Chen et al. (2023) wurde festgestellt, dass unplugged Aktivitäten eine positive Rolle bei der Förderung der CT-Fähigkeiten von Schülerinnen und Schüler spielen können. Hinsichtlich der Lernmittel zeigten sich die positivsten Effekte auf die CT-Fähigkeiten bei der Verwendung von Brettspielen und Papieraktivitäten. Roboter- und Blockaktivitäten ohne Computerunterstützung vermittelten den Schülern ein intuitives Erfolgserlebnis, was wiederum ihr Interesse und ihre Motivation steigerte und somit die Entwicklung von CT-Fähigkeiten förderte (Sun et al., 2021). Die Untersuchung von Busuttil und Formosa (2020) stellte ebenfalls fest, dass das Unplugged-Computing eine wirksame Lehrmethode darstellt, welche starkes Engagement und aktive Beteiligung der Lernenden fördert

sowie die Teamarbeit und Kollaboration stärkt. Dazu ermöglichen Unplugged-Aktivitäten es Lernenden, ihre CT-Fähigkeiten zu entwickeln, ohne umfangreiche Programmierkenntnisse zu benötigen (Kazimoglu et al., 2012).

#### **4.3.1.5.6. TIPP&SEE**

Es gibt bereits spezifische Implementierungsansätze, die zur Förderung und Integration von CT-Fähigkeiten entwickelt wurden. Ein spezifischer Ansatz ist zum Beispiel die TIPP&SEE-Methode, die insbesondere in Grundschulen zur Vermittlung der Programmiersprache Scratch angewendet wird. TIPP&SEE fördert das Verständnis der Programmierkonzepte durch gezielte Exploration und Änderung von Code, wobei „TIPP“ für die Schritte steht, ein Projekt zu verstehen, und „SEE“ für das Erkunden des Codes (Bocconi et al., 2022). Die experimentelle Forschung von Salac et al. (2020) dokumentierte, dass Schülerinnen und Schüler, die die TIPP&SEE-Strategie anwendeten, statistisch signifikante Verbesserungen ihrer Leistungen erzielten. Dies unterstreicht das Potenzial der TIPP&SEE-Strategie als effektive Lernmethode für CS und CT.

#### **4.3.2. Hindernisse**

##### **4.3.2.1. Dauer und Zeit**

Merino-Armero et al. (2022) legten dar, dass die Dauer einer Lerneinheit einen wesentlichen Einfluss auf die Entwicklung und das Erlernen von CT hat. Etwa eine Stunde pro Sitzung schien optimal zu sein, um die Konzentration und Effektivität der Intervention zu maximieren. Dies deutet darauf hin, dass kürzere Interventionszeiten effektiver sein können, weil sie wahrscheinlich weniger belastend für die Lernenden sind. CT ist jedoch allgemein ein komplexer Denkprozess, der sich über die Zeit hinweg entwickelt, und kann nicht durch übermäßig lange oder intensive Sitzungen beschleunigt werden. Längere Interventionszeiten können insgesamt zu einer deutlicheren Verbesserung der CT-Fähigkeiten führen, was die Analyse von Xu et al. (2023) darlegte. Beispielsweise erforderte die

Entwicklung von kritischem Denken, im Gegensatz zu subjektivem Wissen, eine schrittweise Akkumulation. Fähigkeiten im Problemlösen, ein wesentlicher Bestandteil von CT, ließen sich nicht kurzfristig entwickeln oder signifikant verbessern, sondern benötigten eine ausgedehnte pädagogische Praxis.

Beim Einsatz von Werkzeugen zeigte sich, dass die Wirkung von nicht-digitalen Aktivitäten auf CT mit der Zeit nachlässt, was darauf hindeutet, dass Interventionen, die länger als einen Monat dauern, an Effektivität verlieren können (Chen et al., 2023). Dies steht im Einklang mit Chauhan (2017), der herausfand, dass längere technologische Lerninterventionen (über 6 Monate) bei Grundschulern weniger wirksam sind als kurzfristige Maßnahmen.

#### **4.3.2.2. Lehr- und Lernumgebungen**

In Lehr- und Lernumgebungen können Probleme wie geringes Engagement und abweichendes Verhalten der Schülerinnen und Schüler auftreten (Mills et al., 2024). In Programmierklassen wurden negative Verhaltensweisen wie geringes Engagement (Franetovic, 2016) und Ablenkung (Egbert et al., 2021) beobachtet. Laut Tonbuloğlu und Tonbuloğlu (2019) wurden Mängel im Ablauf des CT-Unterrichts hauptsächlich durch Schwierigkeiten bei der Zeitplanung und Teamarbeit in überfüllten Klassenräumen, ablenkende und übermäßig detaillierte Unterrichtsmaterialien sowie demotivierende Faktoren verursacht. Weitere Bedenken betrafen das Klassen- oder Workshop-Umfeld selbst. Beispielsweise beeinträchtigte ein steriler Computerraum, der von der Bibliothek getrennt war, die Entwicklung von Videospielprojekten, da die Geräusche der Spiele nicht mit der erforderlichen Ruhe der Bibliothek vereinbar waren und eine dynamische Umgebung verhinderten (Parry et al., 2020).

## **4.4. Kontinuierliche und ausreichende Finanzierung**

Die vierte Bedingung analysiert die kontinuierliche und ausreichende Finanzierung für die CT-Implementierung. Die Analyse ergibt, dass Investitionen in Lehrerfortbildungen und Ressourcen vorteilhaft sind. Jedoch erschweren mangelnde Zugänglichkeit zu Materialien die CT-Umsetzung im Unterricht.

### **4.4.1. Erfolgsfaktoren**

#### **4.4.1.1. Investitionen**

Schulen die Möglichkeit zu geben, in Fortbildungen, Ressourcen und Anleitungen für Lehrkräfte zu investieren, kann den Prozess der CT-Implementierung erleichtern (Balanskat et al., 2018). In den von den Autoren besuchten Schulen hat sich gezeigt, dass solche Möglichkeiten das Engagement erhöhen und Lehrkräfte dazu motivieren, erforderliche Anpassungen vorzunehmen.

### **4.4.2. Hindernisse**

#### **4.4.2.1. Mangelnde Zugänglichkeit**

Kim und Kim (2018) identifizierten in ihrer Studie unzureichendes Material zu Lehrstrategien und Inhalten im Informatikunterricht, sowie einen Mangel an physischer Infrastruktur. In einer weiteren Untersuchung von Kale et al. (2018) wurde hervorgehoben, dass spezialisierte technologische Ausrüstungen wie Spiele und Robotik-Kits in amerikanischen Schulen schwer zugänglich waren. Dies stellte eine erhebliche Barriere für Lehrkräfte dar, die Robotik in den CT-Unterricht einbinden möchten, da ein erweiterter Zugang zu solcher Hardware erforderlich wäre. Ein weiteres Problem betraf die Installation von Software auf Schulcomputern. Viele Lehrer berichteten, dass sie zwar die Erlaubnis hatten, Softwareanfragen zu stellen, die Umsetzung jedoch oft langsam oder erst in den Schulferien durchgeführt wurde. Lehrpersonen mussten ihre technologischen Bedürfnisse somit gut im Voraus

planen, sodass spontane Integration von neuen Tools während des laufenden Semesters nahezu unmöglich war.

## **4.5. Gerechter Zugang**

Die fünfte Bedingung analysiert den gleichberechtigten Zugang zu CT für alle Beteiligten. Die Analyse ergibt, dass eine Verpflichtung der Kernelemente des Informatiklehrplans und unplugged-Aktivitäten für einen gleichberechtigten CT-Zugang vorteilhaft sind. Jedoch stellen sozioökonomische Hürden und Geschlechterunterschiede signifikante Barrieren dar, die durch Stereotypen und strukturelle Benachteiligungen verschärft werden.

### **4.5.1. Erfolgsfaktoren**

#### **4.5.1.1. Verpflichtung von Kernelementen**

In ihrer Studie zeigten Bocconi et al. (2022), dass die Verpflichtung von bestimmten Kernelementen des Informatiklehrplans den Zugang zu diesem Bildungsbereich gerechter und inklusiver gestalten kann. An Schulen, an denen Informatik obligatorisch ist, haben alle Schülerinnen und Schüler gleichermaßen Zugriff darauf. Probleme mit dem Zugang entstanden erst, wenn Informatik als optionales Fach angeboten wurde. Zum Beispiel ist in Kroatien Informatik in den fünften und sechsten Klassen obligatorisch. In den siebten und achten Klassen, wo es als Wahlfach gilt, wählten im Schuljahr 2020/21 dennoch 74% beziehungsweise 72,2% der Schülerinnen und Schüler dieses Fach.

#### **4.5.1.2. Unplugged-Aktivitäten**

Während plugged-in Aktivitäten wie Programmieraufgaben am Computer der Mainstream ist, erweist sich nicht digitales (unplugged) Werkzeug besonders wichtig für Schulen ohne angemessene technologische Ressourcen, Internetverbindungen oder sogar Strom (Brackmann et al., 2017). Die „unplugged“-Pädagogik bietet diverse Vorteile zur Förderung einer inklusiven CT-Bildung (Chen et al., 2023), da sie durch ihre Kosteneffizienz,

Unabhängigkeit vom Einsatz von Computern, kein Bedarf an IKT-Kenntnissen der Lehrkräfte und einfache Implementierung eine geeignete Alternative darstellt (Busutil & Formosa, 2020; Minamide et al., 2020). Nicht-digitale Aktivitäten bieten die Möglichkeit, mit alltäglichen Materialien, die in fast jedem Klassenzimmer vorhanden sind (z.B. Spielkarten, Marker, Papier, usw.), eine bereichernde Lernerfahrung zu schaffen (Chen et al., 2023).

## **4.5.2. Hindernisse**

### **4.5.2.1. Sozioökonomische Hürden**

Die Herausforderungen bei der Implementierung von Informatikbildung umfassen ungleiche Bildungschancen, -leistungen und den Zugang zu Werkzeugen. Untersuchungen zeigen, dass sozioökonomisch benachteiligte und unterrepräsentierte Schüler nicht frühzeitig genug mit problemorientiertem Unterricht in Kontakt kommen, um ihre Fähigkeiten ausreichend zu entwickeln (Bass et al., 2016). Diese Ungleichheiten wurden während der COVID-19-Pandemie verschärft (Martin et al., 2020).

Schulen in ressourcenarmen Gegenden bieten seltener Informatikkurse an als wohlhabendere Schulen (Banilower et al., 2018; Code.org et al., 2020). In Entwicklungsländern und Regionen mit großen sozioökonomischen Unterschieden fehlt oft die notwendige Infrastruktur (Kim & Kim, 2018). Marginalisierte Schülergruppen, einschließlich solcher aus einkommensschwachen oder ländlichen Gebieten sowie Englischlernende und neurodiverse Schüler, haben in den USA seltener Zugang zu Informatikressourcen (Code.org et al., 2020). Die International Computer and Information Literacy Study (ICILS) 2018 zeigt, dass Schüler aus weniger privilegierten Verhältnissen geringere CT-Kompetenzen aufweisen als ihre privilegierteren Mitschüler (Karpinski et al., 2021). Pietros (2022) fand ebenfalls heraus, dass ein höherer Anteil von Familien unterhalb der Armutsgrenze negativ mit der Nutzung von CT korreliert.



#### **4.5.2.3. Geschlecht**

Obwohl einige Studien keine signifikanten Unterschiede im CT zwischen Jungen und Mädchen feststellten (z. B. Chiazzese et al., 2018; Noh & Lee, 2020; Taylor & Baek, 2019; Witherspoon et al., 2017), berichteten andere von Geschlechtsunterschieden (z. B. Kong et al., 2019; Lee et al., 2017). Einige Studien zeigen, dass Jungen in CT-Aufgaben besser abschneiden. Allsop (2019) stellte fest, dass Jungen in der Verwendung von Programmierkonstruktionen in Alice und Scratch erfolgreicher waren. Atmatzidou und Demetriadis (2016) fanden heraus, dass Mädchen zwar das gleiche CT-Niveau erreichten, aber mehr Zeit benötigten.

Andererseits belegten Durak et al. (2019) jedoch, dass Mädchen höhere Fähigkeiten im CT und im reflektierenden Denken zur Problemlösung haben. Mädchen machten größere Fortschritte in ihren Programmierkenntnissen (Witherspoon et al., 2018) und überholten Jungen ab einem Alter von zehn Jahren in abstrakten Denkfähigkeiten (Rijke et al., 2018). Zudem sind Mädchen kreativer, sowohl im kreativen Denken als auch in der informatischen und algorithmischen Kreativität (Israel-Fishelson et al., 2021).

Schließlich stellten Papavlasopoulou et al. (2019) fest, dass Mädchen in ihren Kompetenzen oder ihrem Verhalten den Jungen nicht unterlegen sind, aber unterschiedliche Herangehensweisen und Sichtweisen auf das Programmieren haben. Ardito et al. (2020) zeigten, dass Jungen und Mädchen durch verschiedene Faktoren beeinflusst werden und sich auf unterschiedliche Aspekte der CT-Fähigkeiten konzentrieren. Jungen fokussieren sich eher auf operative Elemente wie das Konstruieren und Programmieren von Robotern, während Mädchen größeren Wert auf Gruppendynamik legen. Angeli und Valanides (2020) fanden heraus, dass Jungen stärker von aktivitätsbasierten, kinästhetischen Methoden profitieren, während Mädchen größere Fortschritte durch kollaborative Schreibübungen zeigen.

#### **4.5.2.4. Stereotypen**

Schülerinnen und Schüler, die marginalisierten Gruppen angehören, wählen seltener Informatikkurse, auch wenn sie die Möglichkeit dazu haben (Villavicencio et al., 2018). Nach Code.org et al. (2020) und National Center for Women in Computing (2018) gibt es in den USA eine deutliche Unterrepräsentation von Mädchen und ethnischen Minderheiten in Informatikkursen und -prüfungen, was die mangelnde Diversität in der Informatikbranche widerspiegelt. In Großbritannien waren im Jahr 2020 nur 18% von den 78.459 Schülern, die Informatik auf GSCE-Niveau wählten, Mädchen (Bocconi et al., 2022). Auch in der Analyse von Lee et al. (2022) fiel auf, dass sich männliche Schüler häufiger für Informatik entschieden als weibliche. Die ungleiche Beteiligung junger Frauen und farbiger Schülerinnen und Schüler an Informatikkursen ist nicht auf mangelnde Fähigkeiten zurückzuführen, sondern vielmehr auf soziale und strukturelle Barrieren, einschließlich Stereotypen darüber, wer in der Informatik erfolgreich sein kann (Mills et al., 2021). Guggemos (2021) zeigte, dass Motivation CT und geschlechtsspezifische Unterschiede beeinflusst. Intrinsische Werte, frühere naturwissenschaftliche Erfahrungen, außerschulische Aktivitäten, sowie die Einstellung von Freunden und wichtigen Bezugspersonen, insbesondere die Ansichten der Mütter über den Wert der Naturwissenschaften für Frauen, haben bedeutende Auswirkungen auf die beruflichen Präferenzen von 15- bis 18-jährigen Mädchen in naturwissenschaftlichen Feldern (Jacobs et al., 1998).

#### **4.6. Fachkundiges Personal**

Die sechste Bedingung analysiert das fachkundige Personal zur CT-Umsetzung. Die Analyse zeigt, dass ein grundlegendes CT-Verständnis, spezifisches Fachwissen und Selbstvertrauen die CT-Integration verbessern. Vielen Lehrkräften mangelt es jedoch an der nötigen Ausbildung und Erfahrung in CT, sodass sie sich unsicher im Umgang mit CT fühlen und Missverständnisse aufweisen. Zudem stellt die Rekrutierung von fähigem Lehrpersonal eine große Herausforderung dar.

## **4.6.1. Erfolgsfaktoren**

### **4.6.1.1. Konzeptuelles Verständnis und Fachwissen**

Für eine effektive CT-Integration benötigen Lehrpersonen sowohl grundlegendes CT-Verständnis als auch fachliches Wissen in dem Fach, in dem sie CT integrieren möchten (Caskurlu et al., 2021). Erstens ist wichtig, dass Lehrkräfte verstehen, was CT ist, welche Konzepte und Praktiken es umfasst und wie diese mit ihrem Fachbereich verknüpft sind (Chalmers, 2018; Fernández et al., 2018; ISTE, 2020; Kong et al., 2020; Ketelhut et al., 2020, zitiert nach Caskurlu et al., 2021, S.138). Ein fundiertes Fachwissen der Lehrkräfte im Bereich CT ermöglicht es ihnen, die Bedeutung dieses Denkansatzes sowohl für den schulischen Erfolg der Schülerinnen und Schüler als auch für deren zukünftige berufliche Laufbahnen zu erkennen (Bower et al. 2017; Haines et al., 2019; ISTE, 2020; Song, 2017, zitiert nach Caskurlu et al., 2022, S.138). Hug et al. (2018) stellten heraus, dass desto mehr grundlegendes CT-Wissen Lehrpersonen haben, umso eher sie bereit sind, CT-Aktivitäten in ihren Unterricht zu integrieren.

Zweitens benötigen Lehrpersonen Kompetenzen wie Kenntnisse der allgemeinen Pädagogik (allgemeines Verständnis von Theorien, Bildungsprinzipien und Unterrichtsprozessen), Kenntnisse der Schülerinnen und Schüler und ihrer Merkmale, Kenntnisse des schulischen Kontextes (wie Kultur, Umfeld und verfügbare Ressourcen) sowie fachdidaktische Kenntnisse (Caskurlu et al., 2021). Diese sind nach Shulman (1986) von enormer Bedeutung, da sie eine adäquate Anpassung des Unterrichts an die Merkmale der Schülerinnen und Schüler und das präsentierte Thema erlauben.

Das Technological Pedagogical Content Knowledge (TPACK)-Rahmenwerk von Mishra und Koehler (2006) bietet dabei wertvolle Unterstützung, indem es die Integration von Technologie in den Unterricht erleichtert (Koh & Chai, 2014; Mouza et al., 2017; Willermark, 2018, zitiert nach Kong et al., 2020). Dabei hilft das TPACK-Framework, geeignete Werkzeuge und Methoden auszuwählen, um CT erfolgreich in den Fachunterricht zu entwickeln (Kong et al., 2020).

#### **4.6.1.2. Erfahrung und Selbstvertrauen**

Pietros (2022) fand in seiner Analyse heraus, dass Variablen wie Lehrerfahrung und Selbstvertrauen signifikant für die Anwendungshäufigkeit von CT-Konzepten und Methoden im Unterricht waren. Erstens stellte die Autorin fest, dass Lehrpersonen mit mehr Lehrerfahrung sich eher an naturwissenschaftlichen Aktivitäten beteiligen, die CT-Ansätze und -Fähigkeiten wie Abstraktion, Fehlerbehebung, Muster und Zusammenarbeit umfassen. Zudem führen Lehrpersonen, die sich sicher fühlen, positive CT-Ergebnisse ihrer Schülerinnen und Schüler erzielen zu können, eher die entsprechende Praxis durch. Das Vertrauen in CT wächst, wenn Lehrkräfte explizite CT-Lehrerfahrungen sammeln (Chalmers, 2018).

#### **4.6.2. Hindernisse**

##### **4.6.2.1. Mangel an qualifizierten Lehrkräften**

Allgemein besteht ein weitverbreiteter Mangel an Lehrkräften, die ausreichend in der Vermittlung von CT und Informatik geschult sind (Belmar, 2022; Bocconi et al., 2022; Kim & Kim, 2018; Seow et al., 2019; Xing & Zhang, 2020). Den meisten Lehrpersonen auf K-12 Ebene mangelt es an technischem, inhaltlichem und pädagogischem Wissen CT zu unterrichten (Gal-Ezer & Stephenson, 2014). CT ist für viele Lehrkräfte eine neue Disziplin, in der sie weder formalen Unterricht noch eine Anleitung zum Lehren erhalten haben (Bower et al., 2017). Vor allem haben Lehrkräfte Schwierigkeiten damit, Schülerinnen und Schüler, Unterrichtsmaterialien und verschiedene Geräte bei der Durchführung des Informatikunterrichts effektiv zu verwalten (Lockwood & Mooney, 2017; Strawhacker et al., 2018; Tonbuloğlu & Tonbuloğlu, 2019, zitiert nach Lee et al., 2022, S.10). Dieser Mangel an Lehrkräften mit dem nötigen Hintergrund und den erforderlichen Fähigkeiten stellt nach mehreren Berichten und Forschungsliteratur das größte Hindernis für die erfolgreiche Eingliederung von CT-Kompetenzen in die verpflichtenden Lehrpläne dar (Caeli & Yadav, 2020; Grover & Pea,

2018; Royal Society, 2017; Webb et al., 2017; Zhang et al., 2020, zitiert nach Bocconi et al., 2022).

#### **4.6.2.2. Unsicherheiten und Unklarheiten**

Kang, Donovan und McCarthy (2018) stellten fest, dass Lehrkräfte im Vergleich zu anderen Unterrichtsinhalten weniger Wissen und Selbstvertrauen in der Vermittlung von CT und Mathematik haben. Fast zwei Drittel der von Kale et al. (2018) befragten Lehrkräfte waren nicht mit CT-Konzepten und den zugehörigen Technologien vertraut, besonders in ländlichen Grundschulen und städtischen Sekundarschulen. Viele Lehrkräfte fühlen sich unsicher im Umgang mit neuen Unterrichtsmaterialien (Bower et al., 2017). Dies verdeutlicht, dass die digitale Kluft zwar in Bezug auf den Zugang zu Technologien kleiner werden könnte, aber in den Fähigkeiten und der Anwendung bestehen bleibt (von van Dijk, 2005, zitiert nach Kale et al., 2018). Dazu sind Lehrpersonen oft besorgt über ihr Wissen in Informatik und Programmierung (Sentance & Csizmadia, 2017). Ihre geringe Erfahrung in der Kodierung führt zu Unsicherheit, Stress und begrenzten Möglichkeiten, als Vorbild zu fungieren (Egbert et al., 2021). Viele Lehrkräfte zögern, Neues auszuprobieren, akzeptieren ihren geringeren Wissensstand gegenüber einigen Schülern nicht und scheuen sich, aus Fehlern zu lernen (Bocconi et al., 2022). Aus diesen Erkenntnissen kann geschlussfolgert werden, dass Lehrpersonen Unterstützung benötigen, um zu verstehen, wie CT ihre fachlichen Lernziele fördern und wie es effektiv in den Unterricht integriert werden kann (Caskurlu et al., 2021; Sands et al., 2018).

#### **4.6.2.3. Missverständnisse**

Sands et al. (2018) wiesen darauf hin, dass Lehrpersonen Missverständnisse in Bezug auf CT haben, indem sie es beispielsweise mit „Mathematik machen“ verwechseln oder den Einsatz digitaler Werkzeuge fälschlicherweise als CT-Unterricht betrachten. Auch Bower et al. (2017) stellten heraus, dass viele Lehrkräfte unter CT die Nutzung von Technologien im Allgemeinen, das Programmieren von Computern oder das Denken in der Art

eines Computers verstehen. Daher besteht die Notwendigkeit, das Bewusstsein einiger Lehrkräfte für CT und verwandte Konzepte zu stärken, um sie besser auf die neuen Lehrplanänderungen vorzubereiten (Bower et al., 2017).

#### **4.6.2.4. Anerkennung und Karriere**

Die Herausforderung der Lehrerrekutierung wurde als ein Hauptproblem bei der Implementierung von CT in Lehrerausbildungsprogrammen identifiziert. Insbesondere stellt die Lizenzierung ein entscheidendes Hindernis dar. Viele Bundesstaaten in den USA bieten beispielsweise keine Zertifizierung für Informatiklehrer an. Dies senkt die Motivation für Lehrpersonen, einen Abschluss in CT oder CS zu erwerben. Auch stellen die geringe Anzahl von Stellenangeboten für Informatiklehrer an Grundschulen und die hohen Kosten für zusätzliche Ausbildung weitere Hindernisse dar (Li, 2020). Darüber hinaus verfügen viele Länder nicht über spezielle Programme zur Rekrutierung von Lehrkräften für Informatik. Einerseits möchten Fachkräfte mit entsprechenden Informatikkenntnissen nicht im Bildungsbereich arbeiten, da die Lehrergehälter nicht konkurrenzfähig sind. Andererseits bevorzugen Lehrpersonen mit tiefgehenden Informatikkenntnissen und einem starken Interesse am Fach oft Karrierewege, in denen sie ihre Fachkenntnisse professionell weiterentwickeln können, statt in der Lehre zu bleiben (Bocconi et al., 2022).

#### **4.7. Kontinuierliche Weiterbildung**

Die siebte Bedingung analysiert die kontinuierlichen Weiterbildungsangebote im Bereich CT. Die Befunde zeigen, dass gezielte Schulungen und fortlaufende professionelle Entwicklungsprogramme sowohl die pädagogischen Fähigkeiten als auch das Vertrauen der Lehrpersonen stärken. Dennoch mangelt es an effizienter vorberuflicher sowie berufsbegleitender Vorbereitung von Lehrern auf CT-Inhalte. Zudem stellt die Änderung bestehender Programme eine Herausforderung dar.

## **4.7.1. Erfolgsfaktoren**

### **4.7.1.1. CT-(Weiter)Bildung**

Lehrkräften müssen gezielte Schulungen und Unterstützung angeboten werden, um sie auf die neuen Herausforderungen vorzubereiten (Balanskat et al., 2018). Studien zeigten, dass der Zugang zu gut strukturierten professionellen Lernmöglichkeiten die Fähigkeiten der Lehrpersonen, Informatik effektiv zu unterrichten, signifikant erhöhen kann (Dong et al., 2019; Jaipal-Jamani & Angeli, 2017; Zhao et al., 2020, zitiert nach Pietros, 2022, S.113). Einerseits verbessert die Integration von Modulen zum algorithmischen Denken in die Lehrerausbildung die Lernergebnisse der angehenden Lehrkräfte erheblich (Yadav et al., 2011, 2014). In der Studie von 2014 stellten die Autoren fest, dass Lehrkräfte in Ausbildung, die spezielle CT-Module durchliefen, eher bereit waren, CT in ihren zukünftigen Unterricht zu integrieren als jene, die diese Module nicht hatten. Dabei genügte schon eine kurze Maßnahme, um das Verständnis der Lehrkräfte von CT zu ändern. Andererseits verbessern gut gestaltete Fortbildungsprogramme das Vertrauen und die Kompetenz von Lehrkräften in der Anwendung von CT im Unterricht (Dong et al., 2019). Bower et al. (2017) merkten an, dass Lehrkräfte, die an Fortbildungsworkshops teilnahmen, in kurzer Zeit ihre grundlegenden Kenntnisse in den Bereichen CT-Inhalte, Didaktik und Technologie verbesserten. Zudem profitierten sie deutlich von einem gesteigerten Selbstvertrauen im Umgang mit diesen Themen.

Kong, Lai und Sun (2020) identifizierten in ihrer Studie folgende vier Faktoren für eine erfolgreiche Lehrerfortbildung, welche auch für CT gelten: Erstens erfordert effektive Lehrerfortbildung anhaltende Lernphasen, nicht nur die Teilnahme an einem Workshop (Garet et al., 2001, zitiert nach Kong et al., 2020, S.3). Zweitens sollten Lehrer aktive Teilnehmer und keine passiven Empfänger von Wissen in Entwicklungsprogrammen sein (Darling-Hammond et al., 2017, zitiert nach Kong et al., 2020, S.3). Drittens ist eine Verbindung zum schulischen Kontext notwendig, wo Lehrer praktische Erfahrungen sammeln und reflektieren können. Dies funktioniert am besten

in einer Gemeinschaft, die Austausch und gegenseitige Unterstützung fördert (Garet et al., 2001; Sentance & Humphreys, 2018, zitiert nach Kong et al., 2020, S.3). Viertens reicht reines Fachwissen nicht aus; Lehrer müssen auch geeignete pädagogische Ansätze erlernen, die das Lernen der Schülerinnen und Schüler fördern. Daher sollte sich die Lehrerentwicklung auf das pädagogische Fachwissen (PCK) konzentrieren, welches die lehrrelevanten Aspekte des Inhalts betont (Shulman, 1986; Hubbard, 2018; Hwang et al., 2018; Saeli et al., 2011, zitiert nach Kong et al., 2020, S.3).

#### **4.7.2. Hindernisse**

##### **4.7.2.1. Hürden in der Lehrerausbildung**

Forschung zeigte, dass es an Unterstützung für die berufliche Weiterentwicklung der Lehrkräfte mangelt (z.B. Bers et al., 2019; Israel et al., 2015; Lindsay & Hounsell, 2017; Pinto-Llorente et al., 2018, zitiert nach Mills et al., 2024, S.16). Es ist unklar, welches Wissen und welche Kompetenzen für Lehrpersonen zur Integration und unterstützten Vermittlung von CT-Praktiken und -Werkzeugen erforderlich sind (Rich et al., 2019). Dieses unzureichende Verständnis über das Wissen und die Fähigkeiten von Lehrkräften führt zu Unsicherheiten bezüglich der Inhalte, die in Lehrerfortbildungen und Ausbildungsprogramme integriert werden sollen (Caskurlu et al., 2021). Historisch betrachtet fehlen in den meisten Lehrerausbildungsprogrammen spezifische Kurse und Vorbereitung auf das Lehren und Bewerten von CT, wobei die Lücke in vielen Programmen weiterhin besteht (Koshy et al., 2021). Viele Bildungssysteme weltweit sind nicht auf die Entwicklung der Fähigkeiten und des Selbstvertrauens vorbereitet, die für ein effektives Lehren und Lernen im Bereich der CS erforderlich sind (Vegas & Fowler, 2020).

##### **4.7.2.2. Mangelnde effektive Fortbildungen**

Menekse (2015) zeigte in ihrer Studie über berufliche Entwicklung auf, dass die meisten professionellen Entwicklungsprogramme drei Hauptprobleme



haben: Erstens fehlt die Zusammenarbeit zwischen Hochschuleinrichtungen und lokalen Schulorganisationen, zweitens sind die Angebote in der Regel kurzfristig und bieten keine fortlaufende Unterstützung, und drittens liegt der Fokus nicht eindeutig auf dem fachspezifischen pädagogischen Wissen in der Informatik.

Eine Studie von Rich et al. (2021) zu fortlaufenden Weiterbildungen für Lehrkräfte (fünftägige Sitzungen über ein Jahr verteilt) ergab, dass strukturierte Zeiten notwendig sind, um das Codieren effektiv zu lehren. Cabrera et al. (2018) stellten fest, dass punktuelle Fortbildungen allein keinen ausreichenden Bestand an qualifizierten Informatiklehrern sichern können.

#### **4.7.2.3. Änderungen der Programme**

Delyser et al. (2018) deuteten darauf hin, dass die Einführung neuer Inhalte in bestehende Programme eine große Hürde darstellt, da die Programme bereits zahlreiche Akkreditierungs- und Zulassungsanforderungen erfüllen müssen. Li (2020) identifizierte in ihren Experteninterviews, dass die Implementierung von CT-Programmen in der Lehrerbildung aufgrund begrenzter Zeit und Ressourcen auf Hindernisse stoßen könnte. Dazu mangelt es an qualifizierten Lehrerbildnern an Hochschulen, die CT in ihre Lehre einfließen lassen können (Li, 2020). Universitäten stehen selbst vor der Herausforderung, Lehrkräfte für den Informatikunterricht adäquat auszubilden und zu zertifizieren. In den USA zum Beispiel benötigen die Lehrerbildner Unterstützung, um Lehrpläne zu entwickeln, die zukünftige Lehrkräfte auf die Integration von algorithmischem Denken in ihren Unterricht vorbereiten (CSTA, 2013).

### **4.8. Technische Unterstützung**

Die achte Bedingung analysiert die technische Unterstützung für CT. Die Analyse zeigt, dass eine direkte Unterstützung an Schulen durch IKT-pädagogische Berater vorteilhaft ist. Es bestehen nämlich oft technologische Herausforderungen für das Lernen und Lehren von CT.

## **4.8.1. Erfolgsfaktoren**

### **4.8.1.1. Direkte Unterstützung an Schulen**

Ein von Lehrpersonen geschätzter Ansatz ist die direkte Unterstützung und Schulung an den Schulen, zum Beispiel durch IKT-pädagogische Berater. Diese lokalen Initiativen sind ebenfalls effektiv, um die positive Haltung der Lehrkräfte zur Lehre von CT/Programmierung zu stärken. Sie werden ermutigt, Neues zu wagen und aus Fehlern zu lernen, während sie sich auf die Unterstützung durch Kollegen verlassen (Balanskat et al., 2018).

## **4.8.2. Hindernisse**

### **4.8.2.1. Technologische Faktoren**

In der Forschung von Mills et al. (2024) wurden technologische Faktoren häufig als Hindernisse für das Lehren und Lernen von Programmierung und CT identifiziert. Lehrkräfte stießen auf Probleme mit Computerhardware, Software oder Internetverbindungen, die ihre technische Expertise überstiegen (Baytak & Land, 2010; Egbert et al., 2021; Khanlari, 2016, zitiert nach Mills et al., 2024, S.17). Externe technische Schwierigkeiten umfassten schlechte WLAN-Verbindungen, sowie Authentifizierungsprobleme für Sicherheit und Datenschutz (Gaeta et al., 2019). Lernende berichteten von Plattformen (z.B. 'Story-Writing-Coding'), die nur begrenzte Aktionen zuließen, was zu weniger beeindruckenden kodierten Geschichten führte (Price & Price-Mohr, 2018). Weitere Studien fanden heraus, dass Unerfahrenheit mit der Online-Plattform Lernende davon abhielten, den Remix-Prozess zu erkunden (Kafai et al., 2010). Auch kann ein übermäßiges Engagement in Technologie problematisch sein (Egbert et al., 2021). Forschung zeigte, dass die Erfolgsmaßnahme der Neuartigkeit der Technologie für die Lernenden nach zwei Programmzyklen bereits abnahm (Gaeta et al., 2019).

## **4.9. Lehrplanrahmen**

Die neunte Bedingung analysiert den Lehrplanrahmen zur CT-Integration. Die Analyse zeigt, dass die Herstellung von Verbindungen zwischen CT und den Lehrplanzielen, sowie eine gewisse Flexibilität in der Lehrplangestaltung positiv auf die CT-Integration wirken. Andererseits bestehen jedoch Herausforderungen durch einen bereits vollen Lehrplan, zeitliche Einschränkungen, mangelnder Fokus auf theoretische Grundlagen sowie eine zu frühe Lehrplaneinführung.

### **4.9.1. Erfolgsfaktoren**

#### **4.9.1.1. Verbindungen**

Balanskat et al. (2018) fanden in ihrer Studie heraus, dass die Spezifizierung der Verbindung zwischen CT/Programmierung und weiteren digitalen Fähigkeiten sowie den allgemeinen Lehrplanzielen dazu beitragen kann, die Gesamtkonsistenz des Curriculums zu sichern. Zudem steigert ein expliziter CT-Lehrplan das Selbstvertrauen der Lehrkräfte in ihre Fähigkeiten, CT zu unterrichten, was wiederum die Häufigkeit der CT-Praxis erhöht (Chalmers, 2018).

#### **4.9.1.2. Flexibilität in der Lehrplangestaltung**

Lehrpersonen reagieren positiv auf offene Curricula, die ihnen Autonomie bei ihrer Umsetzung ermöglichen, sofern diese mit detaillierten Anleitungen, Unterrichtsbeispielen und hochwertigen Lehrmaterialien kombiniert werden. Dies ist zum Beispiel in Polen und Großbritannien der Fall, wo erfolgreich mit eher offenen und nicht sehr vorschreibenden Lehrplänen gearbeitet wird (Bocconi et al., 2022). Die Autoren stellten außerdem fest, dass Informatiklehrpläne, die oft flexibel gestaltet sind und Fähigkeiten wie Problemlösung und logisches Denken fördern, Lehrmethoden zur Förderung der Selbstständigkeit der Schülerinnen und Schüler unterstützen. Zu diesen

Methoden gehören personalisiertes Lernen, projektbasiertes Arbeiten und Teamarbeit (Bocconi et al., 2022).

## **4.9.2. Hindernisse**

### **4.9.2.1. Konkurrenz im Lehrplan**

Bocconi et al. (2022) identifizierten in ihrer Umfrage, dass die Konkurrenz mit anderen Lehrplanprioritäten eine Herausforderung für CT-Integration auf Grundschul- sowie Sekundarschulebene darstellt. Der bereits volle Lehrplan erschwert die Integration von CT, weil es schwierig ist, zusätzliche CT-Lerninhalte hinzuzufügen (Israel et al., 2015).

### **4.9.2.2. Zeitmangel für CT-Entwicklung**

Es gibt generell einen Mangel an Unterrichtsstunden für die Entwicklung von CT (Bocconi et al., 2022; Kim & Kim, 2018; So et al., 2020). Auch Mills et al. (2024) identifizierten Zeitbeschränkungen als eine bedeutende Barriere für die effektive Vermittlung von Programmierfähigkeiten, besonders in Schulen und Hochschulen. Ein Drittel der von ihnen untersuchten Studien zeigte, dass die im Lehrplan verfügbare Zeit oft nicht ausreichte. Ein Beispiel hierfür war ein zehnwöchiges Programm für Fünftklässler, das darauf abzielte, die Auswirkungen praktischer Programmierübungen zu untersuchen. Es stellte sich heraus, dass nicht genügend Zeit zur Verfügung stand, um die CT-Fähigkeiten der Schülerinnen und Schüler zu verbessern (Tonbuloğlu & Tonbuloğlu, 2019). In Klassen der 2. Klasse, in denen mit Ozobot und Ozoblockly programmiert wurde, führte der hohe Zeitaufwand für die Vorbereitung und Organisation der Unterrichtsmaterialien zu Schwierigkeiten (Egbert et al., 2021). Weitere Lehrkräfte berichteten, dass sie beim Einbeziehen des CT-Denkens in ihre Lehrmethoden auf Hindernisse wie begrenzte Zeit für Planung und Unterricht stießen (Israel et al., 2015). Durch die zeitlichen Begrenzungen werden ebenfalls die Möglichkeiten zur Demonstration von Fachkenntnissen der Schülerinnen und Schüler eingeschränkt. So hatten zum Beispiel Fünftklässler, die ein Spiel zur Ernährung

programmierten, nicht genügend Zeit, alle geplanten Funktionen umzusetzen (Baytak & Land, 2010).

#### **4.9.2.3. Fokus im Lehrplan**

Forscher identifizieren die Herausforderung, dass theoretische Grundlagen und Konzepte im Informatiklehrplan im Vergleich zu den praktischen Erfahrungen eher in den Hintergrund treten. Der Fokus im Informatikunterricht richtet sich nicht primär auf den Lehrplan an sich, sondern vielmehr darauf, wie dieser Lehrplan implementiert und von den Schülerinnen und Schülern aufgenommen wird. Dabei sind die essenziellen theoretischen Grundlagen und Konzepte sowohl für Lehrkräfte als auch für Schüler von großer Bedeutung, um das Verständnis des CT-Konzepts zu ermöglichen (Bocconi et al., 2022).

#### **4.9.2.4. Zu frühe Lehrpläneinführung**

Der Erfolg einer Bildungsreform wird maßgeblich dadurch bestimmt, wie effektiv Lehrkräfte die Neuerungen in der Unterrichtspraxis anwenden können. Für den Erfolg ist entscheidend, dass Lehrpersonen den überarbeiteten Lehrplan für den Informatikunterricht vollumfänglich begreifen. Dazu zählen das Verständnis der Intentionen, Bedeutungen und Ziele des Lehrplans sowie die Kenntnis der erforderlichen Methoden, um diese Ziele präzise und verantwortungsvoll erreichen zu können (Kim & Kim, 2018). Bocconi et al. (2022) stellten in ihrer Studie fest, dass eine effektive Verknüpfung von CT-Fähigkeiten mit bestehenden Lehrplänen Zeit erfordert, um einen fundierten Integrationsprozess sicherzustellen. Dies schließt angemessene Weiterbildungen für Lehrkräfte und die Einführung wirksamer neuer Lehrmethoden mit ein. In vielen europäischen Ländern, darunter Großbritannien und Polen, wurden jedoch Informatiklehrpläne eingeführt, obwohl es an vollständig ausgebildeten Lehrkräften mangelte. Dieses Problem wird oft als „Henne-Ei-Problem“ beschrieben, da keine Mittel für die Lehrerausbildung bereitgestellt wurden, bevor der Lehrplan offiziell eingeführt wurde. In Großbritannien dauerte es mehrere Jahre, bis ausreichende

Schulungsmöglichkeiten für Lehrer verfügbar waren, was dazu führte, dass Lehrpersonen gelegentlich unvorbereitet einen neuen Lehrplan unterrichten mussten (Bocconi et al., 2022).

## **4.10. Schülerzentriertes Lernen**

Die zehnte Bedingung analysiert das schülerzentrierte Lernen im CT-Kontext. Die Ergebnisse unterstreichen, dass schülerzentriertes Lernen im Bereich des CT von verteilter Expertise und selbstgesteuertem Lernen profitiert, wobei die Formulierung von Aufgabestellungen wichtig ist. Jedoch empfinden viele Lernende kognitive Schwierigkeiten, unter anderem bei komplexen Programmieraufgaben.

### **4.10.1. Erfolgsfaktoren**

#### **4.10.1.1. Verteilte Expertise unter Lernenden**

Eine verteilte Expertise in Bildungsprogrammen für Programmierung und CT ist wesentlich für erfolgreiche Lernumgebungen in diesem Bereich (Mills et al., 2024). Verteilte Expertise bedeutet, dass kognitive Fähigkeiten und Wissen innerhalb eines Netzwerks von Lernenden geteilt werden. Dies tritt auf, wenn Lernende in Gruppen arbeiten, einander Feedback geben und sich unterstützen (Bass et al., 2016). Solche Lernsituationen ermöglichen den Schülerinnen und Schülern, sich als Novizen-Experten in bestimmten Wissensgebieten zu etablieren und anderen in ihrem Fachgebiet zu helfen. Diese Art der Wissenskonstruktion geschieht sowohl zufällig als auch gezielt herbeigeführt (Brown et al., 1993). Besonders effektiv ist das Programmierenlernen, wenn Lernende als aktive Gestalter ihres Wissens einbezogen werden, wie bei gemeindebasierten Digitaldesign-Projekten (Calabrese Barton & Tan, 2018). In Fällen, in denen Schülerinnen und Schüler zusätzliche Lernunterstützung benötigten, hat sich die Nachahmung erfahrenerer Mitschüler als hilfreich erwiesen (Bargagna et al., 2019, zitiert nach Mills et al., 2024, S.8). Dabei soll die Paarung von Schülerinnen und Schülern so strukturiert werden, dass sie einen konstruktiven und unterstützenden

Dialog fördert, der auf die jeweiligen Aufgaben ausgerichtet ist (DeVane et al., 2016).

#### **4.10.1.2. Eigenverantwortliches Lernen**

Lernende erzielen deutliche Fortschritte in der Programmierung, wenn sie eigene Kodierungsprojekte entwickeln. Entscheidend dabei sind hohe Eigenverantwortung und Wahlmöglichkeiten für die Studierenden, kombiniert mit zeitnahen und klaren Anleitungen. In selbstgesteuerten, interessenbasierten Lernkontexten verbesserten Schülerinnen und Schüler ihre CT-Fähigkeiten und ihre informatischen Praktiken (Mills et al., 2024). Die Arbeit an realen Problemen bildet den Kern einer erfolgreichen Informatikbildung, wobei das Erschaffen etwas Eigenes dazugehört. Dabei können die Schülerinnen und Schüler ihre Fehler im Prozess finden und korrigieren (Bocconi et al., 2022). Starre Lehrstile hingegen schränken die Motivation und das Engagement der Schülerinnen und Schüler ein, indem sie zu spezifischen Projekten statt einer breiteren Auswahl zwingen (Hughes & Morrison, 2018).

#### **4.10.2. Hindernisse**

##### **4.10.2.1. Entdeckendes Lernen**

Obwohl Papert (1980) argumentierte, dass Schülerinnen und Schüler durch entdeckendes Lernen am besten lernen, zeigen Forschungen, dass diese Methode aufgrund chaotischer Abläufe und mangelnder Struktur problematisch sein kann. Beispielsweise neigten Lernende, die die Programmiersprache Logo verwendeten, oft zu zufälligem und impulsivem Programmieren. Ein strukturierter und geführter Ansatz hingegen fördert zielgerichteteres Verhalten, ein besseres Verständnis von Konzepten und höheres Denkvermögen (Ratcliff & Anderson, 2011). Darüber hinaus kann zu viel Freiheit einige Schülerinnen und Schüler mit Lernschwierigkeiten lähmen, während zu viel Hilfe sie daran hindert, Fehler zu machen und Risiken im Problemlösen zu nehmen (Hughes & Morrison, 2018).

#### **4.10.2.2. Kognitive Schwierigkeiten**

Mills et al. (2024) identifizierten kognitive Herausforderungen, die Schülerinnen und Schüler bei der Entwicklung von CT-Fähigkeiten erleben. Forscher fanden Lernhindernisse wie mangelndes Fachwissen und Schwierigkeiten, dieses Wissen in das Programmieren zu integrieren (Tonbuloğlu & Tonbuloğlu, 2019). Lernende hatten Schwierigkeiten, erlernte Fähigkeiten und Konzepte von einem Kontext auf einen anderen zu übertragen, was das domänenspezifische Problemlösen und den Transfer von Lerninhalten erschwerte (Barak & Zadok, 2009; Falloon, 2016). Die Motivation der Schüler sank, wenn sie die Verbindung zwischen den erlernten Lektionen und dem Informatikunterricht nicht erkennen konnten und das Gefühl hatten, am Ende des Unterrichts nichts gelernt zu haben (Tonbuloğlu & Tonbuloğlu, 2019). Sentance und Csizmadia (2017) berichteten ebenfalls von fehlendem Fachwissen, Geduld und Widerstandsfähigkeit bei den Schülerinnen und Schülern.

#### **4.10.2.3. Komplexe Programmieraufgaben**

Hindernisse im Lernfortschritt von Lernenden und Lehrpersonen im CT-Bereich umfassen kognitiv basierte Herausforderungen, die oft komplexer sind als fehlende Fähigkeiten, Wissen, Erfahrung und Kreativität (Mills et al., 2024). Die häufigste Lernschwierigkeit (15,8 %) betraf die Komplexität von Programmieraufgaben (Bargagna et al., 2019). Dies umfasste oft Aktivitäten, die für junge Anfänger zu anspruchsvoll waren, wie zum Beispiel die Nutzung globaler Variablen zur Punktevergabe in einem Computerspiel (Denner et al., 2012). Zudem führten Schwierigkeiten beim Verstehen von Anweisungen dazu, dass die Teilnehmer komplexere Programmieraufgaben und CT-Denken weniger geschickt umsetzten (Bargagna et al., 2019; Frydenberg, 2015). Die Komplexität einiger Programmiersoftware führte dazu, dass Lernende Schwierigkeiten hatten, wichtige Programmierkonzepte anzuwenden oder sinnvoll zu kombinieren (Theodosiou & Karasavvidis, 2015). Probleme beim Programmieren bewirkten in manchen Fällen sogar, dass



Lernende die Arbeit aufgaben (Pinto-Llorente et al., 2018, zitiert nach Mills et al., 2024, S. 18). Auch andere Forscher stellten fest, dass viele Schülerinnen und Schüler Herausforderungen mit Programmierkonzepten oder -werkzeugen empfinden (Allsop, 2019; Chang, 2019; Durak et al., 2019; Garneli & Chorianopoulos, 2018, zitiert nach Lee et al., 2022, S.10). In einer Untersuchung von Chang aus dem Jahr 2019 zeigte sich, dass es Grundschüler innerhalb der Scratch-Plattform schwerfiel, abstraktere und komplexere CT-Konzepte wie Bedingungen, Schleifen und Variablen in die Aktivität zu integrieren.

## **4.11. Bewertung und Evaluation**

Die elfte Bedingung analysiert die Bewertung und Evaluation von CT im Unterricht. Die Analyse ergibt, dass es bereits validierte und effektive Bewertungsmethoden und -tools gibt. Jedoch stellen die Messbarkeit von CT, die Bewertung von Praktiken gegenüber Wissen, sowie die Bewertung von CT in anderen Fächern bedeutende Herausforderungen dar. Zudem beeinträchtigen fehlende Klarheit in der Definition und unzureichende Validierung von Bewertungsinstrumenten die Effektivität der Bewertungsprozesse.

### **4.11.1. Erfolgsfaktoren**

#### **4.11.1.1. Vielseitige Bewertungswerkzeuge**

Formative und iterative Tools, wie Dr. Scratch (Moreno-León & Robles, 2015) und der CT Pattern Graph (Koh et al., 2014), sind darauf ausgelegt, Lernenden unmittelbares und automatisiertes Feedback zu bieten, was eine schnelle Verbesserung ihrer Fähigkeiten ermöglicht. Der Portfolio-basierte Ansatz hingegen ermöglicht die ganzheitliche Betrachtung über die durch Projekte oder Arbeitsprodukte erworbenen CT-Fähigkeiten der Schülerinnen und Schüler und kann ihnen hilfreiches Feedback für ihr zukünftiges Lernen liefern (Tang et al., 2020).

Eine wirksame Strategie zur Überprüfung der Gültigkeit einer Leistungsaufgabe besteht außerdem darin, sie mit den von den Schülerinnen und

Schülern gegebenen Erklärungen zu ihrem Arbeitsprozess zu verknüpfen (Weintrop et al., 2021a). Tang et al. (2020) betonten in ihrer Studie, dass Interviews zur Messung von CT die Möglichkeit bieten, die Denkprozesse der Lernenden besser zu verstehen. Besonders mit modernen Tools wie automatischer Transkription und Analysetechnologie könnten Interviews eine noch bedeutendere Rolle bei der Erforschung der CT-Denkweise von Schülerinnen und Schülern spielen.

Der Computational Thinking Test (CTt) von Román-González et al. (2017a) wurde mehrfach als CT-Bewertungstool validiert (Román-González, 2015; Román-González et al., 2017b; Román-González et al., 2018b, 2018a; Wiebe et al., 2019). Es handelt sich dabei um 28 Multiple-Choice-Aufgaben, die insgesamt sieben verschiedene CT-Konzepte abdecken. Da der CTt sich auf die Altersgruppe von zehn bis 16-jährigen Kindern limitiert, haben Tsarava et al. (2022) eine Anpassung des Tests entwickelt, die sich an acht bis zehnjährigen Kinder richtet. In ihrer Studie belegen sie, dass diese „verkürzte deutsche Version des CTt“ (S.4) in der entsprechenden Altersgruppe zuverlässig das CT misst.

Auch die von Korkmaz et al. (2017) entwickelte fünfstufige CT-Skala mit 19 Items ist ein Bewertungsinstrument, das darauf abzielt, die Fähigkeiten im CT messen. Diese Skala wurde umfassend auf Konstruktvalidität und Reliabilität geprüft (durch Faktorenanalysen, Reliabilitätskoeffizienten und Test-Retest-Verfahren) und erwies sich als gültiges und zuverlässiges Messinstrument in fünf Faktoren wie Kreativität, algorithmisches Denken, Kooperativität, kritisches Denken und Problemlösen. Obwohl die Skala an Universitätsstudenten getestet wurde, präsentiert dieses Messinstrument nach Tang et al. (2020) einen soliden Prozess der Reliabilitäts- und Validitätsanalysen im CT-Bereich und ist somit hier erwähnenswert.

Grover (2017) schlussfolgert, dass die Verwendung verschiedener Bewertungsstrategien ein umfassenderes Bild des Lernens und der CT-Fähigkeiten der Schülerinnen und Schüler ermöglicht. Nicht alle Lernziele lassen sich mit denselben Aufgaben oder Bewertungsmethoden erfassen, was besonders für CT mit seinen vielfältigen Aspekten gilt (Weintrop et al., 2021a).

## **4.11.2. Hindernisse**

### **4.11.2.1. Schwierigkeiten in der Messbarkeit**

Die Bewertung von CT ist herausfordernd. Wegen der relativen Neuheit des Konzepts, des Mangels an Konsens über dessen Definition und der vielfältigen Facetten von CT ist es schwierig, genau zu spezifizieren, was gemessen werden soll (Weintrop et al., 2021a). Cutumisu et al. (2019) stimmen dem zu und betonen, dass die Uneinigkeit darüber, welche spezifischen Konzepte, Praktiken und Perspektiven zu CT gehören, die Entwicklung von Bewertungsinstrumenten erschwert. Da CT mehr ist als nur eine Ansammlung einzelner Kompetenzen und das Verständnis abstrakter Konzepte (So et al., 2020), bleibt die Bewertung und Quantifizierung von Denk- und Programmierfähigkeiten, Konzepten, Einstellungen und Praktiken im Bereich CT eine bedeutende Herausforderung (Bocconi et al., 2022). Die fehlende Klarheit wird besonders deutlich, wenn Entwickler von Assessments vorgeben, über die Programmierung hinausgehende Konzepte wie Abstraktion, Generalisierung, Problemlösung und Mustererkennung zu bewerten. Dazu stellten Tang et al. (2020) in ihrer Studie fest, dass CT und Programmierung oft vermischt werden, was zu Problemen bei der Bewertung führt. Darüber hinaus fehlt es an Instrumenten zur Bewertung von CT für verschiedene Dimensionen und konstitutive Elemente wie Kreativität (Mishra & Henriksen, 2012; Repenning et al., 2015).

### **4.11.2.2. Bewertung Endprodukt vs. Lernprozess**

In der Informatik gibt es eine lange Tradition der Entwicklung automatischer Bewertungssysteme, insbesondere für große Einführungskurse auf postsekundärer Ebene (Weintrop et al., 2021b). Piech et al. (2012) argumentieren jedoch, dass die Betrachtung des Endprodukts keine validen Rückschlüsse auf das Wissen der Schülerinnen und Schüler zulässt, im Gegensatz zur Bewertung des Prozesses der CT-Anwendung bei der Programmerstellung. Diese Entwicklungsprozesse liefern wichtige Informationen über das Verständnis der Konzepte, die über das Endprodukt hinausgehen. Die

Durchführung solcher Bewertungen im großen Maßstab ist jedoch schwierig (Weintrop et al., 2021b).

#### **4.11.2.3. Bewertung in anderen Fächern**

Die Bewertung von CT im Zusammenhang mit anderen Fächern ist komplex. Da CT oft in diesen Fächern integriert unterrichtet wird, ist es schwierig, das informatische und algorithmische Denken separat zu bewerten, ohne gleichzeitig das Fachwissen zu testen. Dennoch ist dies notwendig, da CT als wichtige Fähigkeit anerkannt ist und integriert unterrichtet werden soll, um Bildungsstandards zu erfüllen. Lehrkräfte müssen sicherstellen, dass sie die richtigen Fähigkeiten bewerten und dass fachfremdes Wissen die Ergebnisse nicht verfälscht (Weintrop et al., 2021b).

#### **4.11.2.4. Traditionelle Sichtweisen**

Der Erfolg von Informatikinitiativen wird oft nur durch die Einschreibung in AP<sup>5</sup>-Computerwissenschaftskurse und den Erfolg in diesen Prüfungen gemessen, was besonders für marginalisierte Lernende wenig relevant sein kann (Code.org et al., 2020). Diese traditionelle Ausrichtung auf höhere Bildung und spezifische Disziplinen kann dazu führen, dass das Bildungssystem systemische Ungleichheiten verstärkt und das Potenzial der Informatikbildung nicht vollständig ausschöpft (Vakil, 2018; Washington, 2020).

#### **4.11.2.5. Fehlende Validation**

Laut Cutumisu et al. (2019) hinkt die Forschung zur Bewertung von CT hinterher. Die meisten der Bewertungsmethoden, die in formalen und informellen Kontexten eingesetzt werden, sind noch nicht ausreichend validiert, um ihre Integration in den K-12-Lehrplan und ihre Akzeptanz durch Bildungsforscher zu rechtfertigen.

---

<sup>5</sup> Advanced Placement (AP) Computer Science

## **4.12. Engagierte Gemeinschaften**

Die zwölfte Bedingung analysiert die Rolle engagierter Gemeinschaften in der CT-Integration. Die Untersuchung zeigt, dass verteilte Expertise innerhalb von Gemeinschaften, Forschungspraxispartnerschaften und Online-Communities vorteilhaft sind, um Lernumgebungen zu stärken und Lernende aktiv in ihre eigene Bildung einzubeziehen. Jedoch stellen soziale Kommunikationsprobleme, insbesondere in Online-Umgebungen, eine Herausforderung dar.

### **4.12.1. Erfolgsfaktoren**

#### **4.12.1.1. Unterstützende Gemeinschaften**

Ein wesentliches Element erfolgreicher Lernumgebungen für Kodierung und informatisches und algorithmisches Denken ist die Entwicklung und Förderung unterstützender Gemeinschaften. Diese Gemeinschaften der Praxis nutzen das vielfältige und verteilte Fachwissen von Personen, die als fähige Novizen in sowohl Online- als auch Offline-Lernkontexten fungieren, und das sowohl in formellen als auch in informellen Bildungssituationen (Mills et al., 2024). Zu den Kernaspekten gehört die gemeinsame Nutzung von Aufgaben, Ideen, Fähigkeiten, Wissen und Denkstrategien über die Grenzen der Bildung hinaus, um Gemeinschaften, die Industrie und interdisziplinäre Lernumgebungen miteinzubeziehen. In diesen Kontexten werden die Lernenden ermutigt, Fragen zu stellen, Unterstützung zu suchen und anderen zu helfen (Bass et al., 2016; Calabrese Barton & Tan, 2018; Fessakis et al., 2013; Gestwicki & McNely, 2012; Israel et al., 2015; Kafai et al., 2010; Ward et al., 2014, zitiert nach Mills et al., 2024, S.26). Die Zusammenarbeit mit Praxisgemeinschaften, Bildungszentren und die Unterstützung durch Lehrkraftkollegen erweist sich als wesentlich, um offizielle Fortbildungsangebote zu erweitern. Besonders in den Anfangsjahren der Einführung neuer Informatiklehrpläne sind lokale Organisationen oft

entscheidend, um Schulungen anzubieten, wie etwa in Großbritannien beobachtet wurde (Bocconi et al., 2022).

#### **4.12.1.2. Forschungspraxispartnerschaften**

Forschungspraxispartnerschaften (RPP) sind langfristige Kooperationen zwischen Forschern und Lehrkräften, die sich auf die praktische Umsetzung und kontinuierliche Verbesserung von Lehrmethoden, insbesondere im Bereich des CT, konzentrieren (Coburn et al., 2021; Coburn & Penuel, 2016, zitiert nach Pietros, 2022, S.86). Diese Partnerschaften erweisen sich als effektive Strategie zur Integration von CT im naturwissenschaftlichen Unterricht, da sie den Austausch und die Anwendung wissenschaftlicher Erkenntnisse in der Praxis fördern. Lehrkräfte erhalten durch RPPs die Möglichkeit, aktuelle wissenschaftliche Ressourcen zu nutzen und Herausforderungen in ihrem Unterrichtsumfeld direkt anzugehen, während sie gleichzeitig Risiken eingehen und neue Lehrmethoden erproben können. Dieser Ansatz ermöglicht es Forschern, ihre Theorien und Methoden direkt in der Praxis zu testen und weiterzuentwickeln, während Praktiker in ihrem Fachgebiet durch die gemeinsame Arbeit und den regelmäßigen Austausch gestärkt werden (Mulvey et al., 2020; Penuel et al., 2013, zitiert nach Pietros, 2022, S.86).

#### **4.12.1.3. Online-Gemeinschaften für Lernende**

Mills et al. (2024) stellten fest, dass Lernende in der Programmierung motivierter, selbstbewusster und stolzer auf ihre Leistungen sind, wenn sie ihre CT-Projekte einer echten Online-Community präsentieren können. Insbesondere junge Designer interaktiver Medien im Alter von 7 bis 18 Jahren schätzten diese Möglichkeit, um ihre Arbeiten einem Publikum vorzustellen und sich über ihre Erfahrungen auszutauschen. Diese Interaktionen stärkten ihre Identität als Programmierer und brachten ihnen Anerkennung aus der Community ein (Brennan, 2016, zitiert nach Mills et al., 2024, S.11). Gee (2003) betrachtet unterstützende Online-Gemeinschaften, wie Gaming-Communities, als "Affinitätsgruppen" für junge Menschen mit

ähnlichen Programmierinteressen. Diese Gruppen fördern effektive Teamarbeit und projektbasierte Arbeitsumgebungen, die einen Austausch weit über das Klassenzimmer hinaus ermöglichen. Beispielsweise teilten Schülerinnen und Schüler im Alter von 14 bis 18 Jahren nach der Schule ihre Spiele, Animationen und Geschichten in der Scratch-Community und erhielten Anerkennung für ihre Unterstützung und Beiträge (Kafai et al., 2012). Online-Gemeinschaften bieten zudem die Möglichkeit, Code zu remixen, wobei Jugendliche im Alter von 11 bis 13 Jahren unterschiedlich kreativ tätig waren. Dies förderte die Entwicklung von Programmier- und Problemlösungsfähigkeiten in der Schule, insbesondere für Anfänger (Fields et al., 2015; Hagge, 2017; Literat & Kligler-Vilenchik, 2018, zitiert nach Mills et al., 2024, S.12).

Die Forschung hat jedoch auch Herausforderungen in Online-Umgebungen aufgezeigt, wie den Umgang mit negativen Kommentaren und politischen Beiträgen. Manche Schülerinnen und Schüler entschieden sich später dafür, ihre Beiträge wieder zu entfernen (Literat & Kligler-Vilenchik, 2018, zitiert nach Mills et al., 2024, S.11). Trotz gemischter Ergebnisse bezüglich der Art und des Levels der Unterstützung durch Online-Communities, empfanden viele Teilnehmer die Online-Kommunikation als generell ermutigend, und einige änderten ihre Projekte basierend auf konstruktivem Online-Feedback (Fields et al., 2015, zitiert nach Mills et al., 2024, S.12).

#### **4.12.2. Hindernisse**

##### **4.12.2.1. Soziale Kommunikationsprobleme bei Lernenden**

Soziale Kommunikationsprobleme sowohl in Online- als auch Offline-Umgebungen wurden als wesentliche Einschränkungen im Kontext des Programmierens identifiziert (Mills et al., 2024). Insbesondere in Online-Communities führten diese Schwierigkeiten zu Enttäuschungen und Verwirrung bei den Studierenden, wenn ihre Beiträge unbeachtet blieben. Dies trug dazu bei, dass Studierende zögerten, ihre Arbeiten zu teilen, oder sie sogar löschten, um negativer Aufmerksamkeit aus der Community zu entgehen

(Brennan, 2016; Literat & Kligler-Vilenchik, 2018, zitiert nach Mills et al., 2024, S.18). Persönliche Projekte oder Erzählungen von Schülerinnen und Schülern, die auf Plattformen wie Scratch online geteilt werden sollten, wurden häufig aufgrund nachträglicher Bedenken hinsichtlich der Online-Teilung aufgegeben (Burke, 2012).

Wettbewerb unter Freunden und Meinungsverschiedenheiten in Teams stellten weitere soziale Herausforderungen dar (Fessakis et al., 2013; Pinto-Llorente et al., 2018, zitiert nach Mills et al., 2024, S.18). In Offline-Einstellungen beeinträchtigte die Dynamik in Gruppenarbeiten den Lernprozess, insbesondere wenn hoch motivierte Studierende intolerant gegenüber weniger motivierten Kommilitonen wurden (Franetovic, 2016, zitiert nach Mills et al., 2024, S.18).

Die Untersuchungen zeigten, dass 40,8% der festgestellten Hindernisse unvorhergesehene Probleme für Studierende bei der Online-Kommunikation sind. Dies ist ein Hinweis auf die Notwendigkeit einer sorgfältigen Überwachung und Unterstützung von Online-Interaktionen (Mills et al., 2024).

#### **4.13. Unterstützende Richtlinien**

Die 13. Bedingung analysiert die Rolle von unterstützenden Richtlinien für die Integration von CT. Die Untersuchung zeigt jedoch, dass es an politischen Leitlinien fehlt oder diese inkonsistent sind.

##### **4.13.1. Hindernisse**

###### **4.13.1.1. Mangelnde oder inkonsistente politische Leitlinien**

Es fehlt an klaren politischen Leitlinien, die die Implementierung von CT im Bildungssystem unterstützen. Bocconi et al. (2016) betonen, dass eine systematische Integration von CT klare Visionen und spezifische Ziele erfordert. Ohne solche Richtlinien wird die Einbindung von CT in den Bildungsprozess erheblich erschwert, da CT weit mehr als nur einige Stunden Programmierung umfasst und eine durchdachte Strategie benötigt. Eine weitere Studie hebt hervor, dass politische Rahmenbedingungen und



Unterstützung entscheidend sind, um soziale Ungleichheiten im Zugang zu CT-Bildung zu verringern und sicherzustellen, dass alle Schüler von diesen wichtigen Fähigkeiten profitieren können (International Association for the Evaluation of Educational Achievement [IEA], 2021). Kim und Kim (2018) stellten ebenfalls fest, dass inkonsistente Richtlinien zur Informatikbildung die Umsetzung eines einheitlichen Lehrplans und die Förderung von CT in Südkorea behindern.

#### **4.14. Unterstützender externer Kontext**

Die 14. Bedingung analysiert die Rolle eines unterstützenden externen Kontextes für die CT-Implementierung. Die Ergebnisse verdeutlichen, dass die CT-Implementierung maßgeblich von nationaler und internationaler Zusammenarbeit in der Bildungspolitik profitiert.

##### **4.14.1. Erfolgsfaktoren**

###### **4.14.1.1. Internationale Zusammenarbeit**

Nationale und internationale Kooperationen sind entscheidend für die Implementierung von CT im Bildungsbereich (Balanskat et al., 2018). Dabei nimmt die Zusammenarbeit zwischen Autoren, Institutionen und Ländern stetig zu. Dies ist vor allem auf die Bemühungen von Wissenschaftlern aus den USA sowie mehreren europäischen und asiatischen Ländern zurückzuführen, die starke nationale und internationale Verbindungen pflegen (Tekdal, 2021). Solche internationalen Forschungsk Kooperationen fördern die Qualität der Forschung und die Entwicklung neuer Technologien (Kim, 2006).

## **5. Diskussion**

### **5.1. Interpretation der Ergebnisse**

Die vorliegende Arbeit bietet durch die Analyse der Erfolgsfaktoren und Hindernisse bei der Implementierung von CT in der Grundschulbildung wertvolle Einblicke. Obwohl bereits bekannte Daten herangezogen werden, integriert diese Arbeit sie in einen neuen, umfassenden Kontext, was vergleichende Analysen ermöglicht. Dadurch lassen sich konkrete Schlussfolgerungen ziehen, die zur Verbesserung und Weiterentwicklung von CT-Bildungsprogrammen weltweit beitragen können.

Erstens bietet die vorliegende Arbeit den Vorteil einer umfassenden Betrachtung von vielfältigen Faktoren, die die CT-Implementierung beeinflussen. Während viele Studien isolierte Aspekte betrachten (z.B. CT-Bewertung, CT in bestimmten Fächern, unplugged Aktivitäten, Herausforderungen für Lehrkräfte, usw.), verknüpft diese Arbeit zahlreiche Einflussfaktoren in einem globalen Kontext. Sie integriert Forschungsergebnisse aus Europa, Amerika und Asien und ermöglicht so einen umfassenden Überblick über die globalen Trends und Herausforderungen in der CT-Bildung. Dadurch können Muster und Zusammenhänge erkannt werden, die in isolierten Studien nicht sichtbar wären. Zweitens wird eine holistische Analyse der Erfolgsfaktoren und Hindernisse bei der CT-Integration im Grundschulkontext ermöglicht. Einzelne Studien analysieren oft entweder Erfolgsfaktoren oder Hindernisse, jedoch selten beide zusammen. Die vorliegende Arbeit schlägt somit eine Brücke zwischen isolierten Studien und einem globalen Verständnis von CT. Die Kombination von theoretischen Rahmenbedingungen mit praktischen Beispielen bietet zudem tiefere Einsichten und konkrete Handlungsempfehlungen.

In Bezug auf die Forschungsfrage „Welche Faktoren fördern oder behindern die erfolgreiche Implementierung von Computational Thinking als Schulfach in Grundschulen?“ konnten mehrere wichtige Schlussfolgerungen gegeben werden. Erfolgsfaktoren und Hindernisse wurden in verschiedenen

Bereichen identifiziert. Im Folgenden werden die zentralen Aspekte diskutiert, die für die erfolgreiche Implementierung und langfristige Wirksamkeit von CT-Bildungsprogrammen besonders relevant sind. Diese Aspekte wurden ausgewählt, da sie meiner Meinung nach besonders wichtig sind und den größten Einfluss auf die Praxis haben.

Erstens stellen die vielfältigen Begrifflichkeiten und Konzepte rund um CT weltweit einen interessanten Aspekt dar (Hsu et al., 2019). Diesbezüglich könnten einerseits die Klärung und Vereinheitlichung der Terminologie rund um CT und CT-Konzepten ein wesentlicher Schritt sein, um ein einheitliches Verständnis zu erarbeiten. Eine klare Definition und Abgrenzung dieser Begriffe könnten unter anderem Missverständnisse vermeiden, Lehrinhalte und Bewertungen klarer gestalten, eine konsistente Verwendung in Lehrplänen und Bildungspolitiken sicherstellen und Lehrkräfte besser auf die Vermittlung der Inhalte vorbereiten. Gleichzeitig wird die Notwendigkeit und Effizienz einer strengen Definition hinterfragt, da es für den Bildungskontext wichtiger wäre, konkrete Einsatzmöglichkeiten von CT zu identifizieren (z.B. Guzdial, 2011; Voogt et al., 2015). Dies sollte somit in zukünftiger Forschung genauer untersucht werden. Wichtiger ist jedoch, dass ein gemeinsames Verständnis von CT unter allen Bildungsakteuren erreicht wird, unabhängig davon, ob eine strenge Definition vorliegt oder nicht. Dieses grundlegende Verständnis von CT ist essenziell, um Veränderungen voranzutreiben (Balanskat et al., 2018).

In diesem Zusammenhang sollten Regierungen Maßnahmen ergreifen, um eine einheitliche Auffassung von CT zu fördern und eine kohärente Vision für dessen Implementierung zu entwickeln. Eine umfassende Diskussion über CT, sowohl öffentlich als auch innerhalb des Bildungssektors, könnte diese Bemühungen unterstützen. Bocconi et al. (2022) empfehlen ebenfalls, dass Regierungen Aufklärungsinitiativen fördern sollten, die die Vorteile von CT-Fähigkeiten für Schüler hervorheben und eine gemeinsame Integrationsvision unterstützen. Zur besseren Diagnose der Situation sollten Regierungen auf Experten für Technologie und Bildung zurückgreifen (Belmar,

2022) und im Bildungsministerium ein spezielles Department für Informatikbildung etablieren (Kim & Kim, 2018).

Ein wichtiger und höchst interessanter Befund ist die große Anzahl an Lehrpersonen, die sich in der CT-Lehre unvorbereitet und unfähig fühlen (z.B. Kang et al., 2018). Dies stellt wahrscheinlich eines der größten Hindernisse für die Implementierung von CT dar, da Lehrpersonen eine zentrale Rolle bei der Umsetzung spielen. Ebenso beunruhigend ist der Mangel an effektiven Ausbildungsprogrammen und qualifizierten Lehrbildnern an Hochschulen, die zukünftige Lehrkräfte adäquat vorbereiten können, wie von Li (2020) festgestellt. Diesbezüglich betonen Kim und Kim (2018), die Bedenken und Neigungen aller Stakeholder, einschließlich Lehrkräften, zu berücksichtigen. Folglich empfiehlt sich ein offener Dialog, der die emotionalen Bedürfnisse und Anliegen der Lehrpersonen adressiert und gründlich aufnimmt. Außerdem soll die Schaffung eines Netzwerks für professionelle Zusammenarbeit, sowohl online als auch offline, für den Austausch von Ressourcen, Erfahrungen und Lösungen ermöglicht werden. Dies stärkt auch nach Bocconi et al. (2022) das Selbstvertrauen der Lehrkräfte und fördert innovative Lehrmethoden.

Darüber hinaus sollten gezielte Maßnahmen von Schulen und Regierungen die Informatikbildung systematisch fördern und langfristig sichern. Konkret kann hier ein detaillierter Unterstützungsplan für Lehrkräfte und Schulen hilfreich sein, der konkrete Unterstützungsmaßnahmen wie Ressourcen und Peer-Support bietet (Bocconi et al., 2022). Wesentlich sind auch gezielte, umfassende Schulungen mit praxisnahen Beispielen, die sowohl pädagogisches CT-Fachwissen als auch konkrete Umsetzungsbeispiele vermitteln. Universitäten und Hochschulen sollten somit ihre Lehrpläne hinterfragen und bei Bedarf anpassen, um diesen Mängeln entgegenzuwirken. Dabei wird die Zusammenarbeit mit Informatikabteilungen von Li (2020) angeraten, um Fakultätsressourcen und Expertise zu teilen. Auch Unternehmen wie Google sollten nach Xing und Zhang (2020) Lehrmaterialien (z.B. Videos) und eine finanzielle Unterstützung für die Lehrerausbildung bereitstellen.

Die Kooperation zwischen Bildungsministerien, der Industrie und Organisationen wie Graswurzelgruppen wird für die Entwicklung effektiver Trainingsprogramme empfohlen.

Ein weiterer wichtiger Aspekt sind die ungleichen Bedingungen und Zugangsmöglichkeiten zu CT, die durch sozioökonomische, geografische oder andere benachteiligende Faktoren entstehen (z.B. Code.org et al., 2020; Mills et al., 2021). Um Chancengleichheit zu gewährleisten, sollten Regierungen eine nachhaltige Finanzierung sicherstellen, die alle Schulen mit geeigneter CT-Ausstattung versorgt. Bei finanziellen Hindernissen sollten kostengünstige und alternative Werkzeuge eingesetzt werden. Unplugged-Aktivitäten sind hierbei besonders wertvoll, da sie ohne spezielle technische Ausrüstung oder Vorkenntnisse durchgeführt werden können (Busuttill & Formosa, 2020; Minamide et al., 2020). Angesichts der oft falsch vertretenen Annahme, dass CT-Unterricht den Einsatz digitaler Werkzeuge erfordert (Sands et al., 2018), ist die positive Wirkung nicht-digitaler Methoden zur Förderung von CT besonders bemerkenswert.

Zudem bestehen klare Unklarheiten bezüglich der Geschlechterrollen in der CT-Lehre (z.B. Kong et al., 2019). Die unterschiedlichen und teilweise widersprüchlichen Perspektiven unterstreichen die dringende Notwendigkeit weiterer Forschung. Klare Richtlinien und Strategien zur Förderung von Geschlechtergleichheit, Gerechtigkeit und Inklusion in der Informatikbildung sind erforderlich. Diese sollten Maßnahmen, gezielte Ressourcen und Programme umfassen, die einen geschlechtsneutralen und vorurteilsfreien Zugang zu CT fördern. Mills et al. (2021) schlagen diesbezüglich das Universal Design for Learning vor. Darüber hinaus empfiehlt Belmar (2022) eine Verpflichtung zum CT-Unterricht für alle Schülerinnen und Schüler.

Richtlinien und Strategien sind zwar grundlegend, aber ihre tatsächliche Wirkung entfaltet sich erst durch die Lehrkräfte und deren praktische Umsetzung. Deshalb ist es entscheidend, dass Lehrpersonen sich der Geschlechterstereotypen in der CT-Bildung bewusst sind, diese kritisch reflektieren und konsequent vermeiden, da solche Stereotype erhebliche

Barrieren für Lernende darstellen können. Dieses Bewusstsein sollte auch in den Ausbildungs- und Weiterbildungsprogrammen für Lehrkräfte thematisiert und gefördert werden.

Die Analyse verdeutlicht die dringende Notwendigkeit für validierte formative und summative Bewertungstools (z.B. Cutumisu et al., 2019). Ohne Bewertungsinstrumente ist es schwierig, die Fortschritte und Fähigkeiten der Schülerinnen und Schüler im CT-Bereich zuverlässig zu messen und gezielte Fördermaßnahmen zu entwickeln (Grover, 2017). Den Lehrkräften sollten effektive Richtlinien und Methoden zur CT-Bewertung zur Verfügung gestellt werden, damit sie den Lernfortschritt der Schüler überwachen und aufkommende Missverständnisse erkennen und angehen können. Dabei sollten detaillierte Kriterien für die Bewertung von CT-Fähigkeiten helfen, die sowohl das Verständnis der Schülerinnen und Schüler für grundlegende CS-Konzepte als auch ihre CT-Fähigkeiten umfassen. Bocconi et al. (2022) stimmen dem überein und raten auch eine kontinuierliche Bewertung von Unterricht, Lernen und Führung an, um den Erfolg der Informatikbildung zu messen und zu verbessern.

Ein interessanter Befund ist, dass eine frühzeitige Einführung von CT-Fähigkeiten vorteilhaft ist (z.B. Tran, 2019). Junge Menschen sollten bereits früh praktische Erfahrungen sammeln, um ihre Lernfähigkeiten im Laufe der Zeit zu entwickeln (Bass et al., 2016). Dies widerspricht der weit verbreiteten Annahme, dass komplexe CT-Konzepte erst für ältere Schülerinnen und Schüler geeignet sind. Obwohl weiterer Forschungsbedarf besteht (Arfé et al., 2020), sind diese Erkenntnisse wertvoll. Sie weisen auf das Potenzial einer frühen Förderung analytischer Fähigkeiten hin und geben Anlass, die aktuellen Bildungsansätze zu überdenken und gegebenenfalls anzupassen. Regierungen und Bildungsinstitutionen sollten sich dessen bewusst sein und geeignete Maßnahmen zur frühkindlichen CT-Förderung ergreifen.

Ein weiterer bedeutender Befund ist die Wichtigkeit nationaler und internationaler Zusammenarbeit (Balanskat et al., 2018). Der länderübergreifende Austausch von Erfahrungen und Best Practices ist dringend zu empfehlen. Trotz der unterschiedlichen Bedingungen in jedem Land gibt es die Möglichkeit, erfolgreiche Erkenntnisse zu sammeln und sie unter Berücksichtigung der jeweiligen Situation zu nutzen. Xing und Zhang (2020) betonen auch, dass Vertreter des Bildungsministeriums CT-bezogene Kurse in fortgeschritteneren Ländern besuchen sollten, um wertvolle Rückmeldungen für die Politikgestaltung zu erhalten. Außerdem ermöglicht das Internet es Bildungsexperten weltweit, an öffentlichen Gemeinschaften teilzunehmen, um Erfahrungen auszutauschen und Fragen zu klären. Alternativ können Wettbewerbsgruppen gebildet werden, in denen Lehrkräfte gemeinsam Lehrpläne entwickeln und deren Umsetzung vergleichen. Darüber hinaus kann die Industrie als Vermittler dienen, um internationale Verbindungen zu fördern (Xing & Zhang, 2020).

Zusammenfassend zeigt die Analyse der Herausforderungen und Erfolgsfaktoren bei der Implementierung von CT in Grundschulen, dass eine umfassende und kohärente Strategie erforderlich ist. Die erfolgreiche Implementierung von CT als Schulfach in Grundschulen erfordert ein gemeinsames Verständnis, gut vorbereitete Lehrkräfte, valide Bewertungstools, Chancengleichheit bei der Ausstattung und dem Zugang, frühzeitige Förderung der CT-Fähigkeiten sowie nationale und internationale Zusammenarbeit. Nur durch diese koordinierten Maßnahmen können die identifizierten Hindernisse überwunden und eine nachhaltige Integration von CT in den Bildungsprogrammen gewährleistet werden. Es ist notwendig, sich kritisch mit bestehenden Konzepten auseinanderzusetzen und auch oft vernachlässigte Aspekte in der aktuellen Forschung und Praxis zu berücksichtigen.

## **5.2. Grenzen der Arbeit**

Die vorliegende Analyse zur Implementierung von CT in Grundschulen stößt in mehreren Bereichen an ihre Grenzen, die es wichtig zu erkennen und zu

reflektieren gilt, um die Ergebnisse und Empfehlungen in den richtigen Kontext zu setzen. Erstens mussten alle Quellen von Englisch ins Deutsche übersetzt werden, was die Interpretation beeinflussen kann. Während des Übersetzungsprozesses besteht das Risiko, dass spezifische terminologische Feinheiten verloren gehen. Dies kann zu Ungenauigkeiten führen, da manche Fachbegriffe und Konzepte im Deutschen keine direkten Entsprechungen haben. Ein Beispiel hierfür ist der Begriff „Computational Thinking“, der im Deutschen unterschiedlich übersetzt werden kann, etwa als „informatisches und algorithmisches Denken“, „rechnerisches Denken“ oder „informatisches Denken“.

Zweitens kann die Natur der untersuchten Quellen die Studie erheblich einschränken. Die Einbeziehung sekundärer Datenquellen kann zwar effizient sein und breite Einblicke ermöglichen, reduziert jedoch die Kontrolle über die Qualität und Konsistenz der Daten. Dazu wurden einige Studien nur über kurze Zeiträume durchgeführt, was die Zuverlässigkeit und Generalisierbarkeit der Ergebnisse beeinträchtigen kann. Folglich können die Ergebnisse der Analyse durch die Genauigkeit und Relevanz der ursprünglichen Datenquellen beeinflusst und potenziell verzerrt sein. Es wurde jedoch versucht, vor allem qualitativ hochwertige und zuverlässige Quellen zu verwenden.

Drittens spiegeln die analysierten Daten möglicherweise nicht die volle Vielfalt und Komplexität globaler Bildungssysteme wider. Studien, die in bestimmten kulturellen oder bildungspolitischen Kontexten durchgeführt wurden, könnten Ergebnisse liefern, die nicht universell auf andere Kontexte übertragbar sind. Besonders in einem Land wie Luxemburg, das spezifische bildungspolitische Strukturen und Herausforderungen aufweist, könnte dies bedeuten, dass einige Empfehlungen der Studie angepasst werden müssen, um den lokalen Gegebenheiten gerecht zu werden.

Viertens kann die Gleichsetzung von CT mit Programmierkenntnissen, wie sie oft in der Bildungspraxis vorkommt, zu einer eingeschränkten Auffassung von CT führen. Diese Tendenz übersieht wichtige Aspekte wie



logisches Denken, Problemlösung und kreatives Denken. Dadurch könnten nicht alle relevanten Erfolgsfaktoren und Hindernisse bei der Analyse vollständig erfasst worden sein, was die Tiefe und Genauigkeit der Untersuchung beeinträchtigt und zu unvollständigen oder verzerrten Ergebnissen führt.

Fünftens ist zu beachten, dass viele Herausforderungen und Erfolgsfaktoren noch nicht ausreichend durch empirische Forschung validiert sind, weshalb sie im Analyseteil trotz ihrer Relevanz nicht aufgegriffen wurden (z.B. differenzierte Instruktionen, Schulinfrastruktur, usw.). Die Notwendigkeit weiterer Forschung ist daher entscheidend, um die Wirksamkeit von CT-Implementierungsstrategien zu untermauern.

Sechstens werden in dieser Analyse verschiedene potenziell erfolgreiche Aspekte nicht behandelt, da sie allgemeiner Natur sind und nicht spezifisch auf den CT-Kontext ausgerichtet (z.B. Fehlerakzeptanz, Universal Design for Learning, usw.). Diese sind jedoch auch im Rahmen der CT-Bildung von erheblicher Relevanz und sollten nicht vernachlässigt werden.

Zusammenfassend lässt sich sagen, dass die genannten Limitationen der Analyse die Notwendigkeit betonen, die Ergebnisse mit Vorsicht zu interpretieren und die Empfehlungen im Lichte weiterführender, spezifisch angepasster Forschung zu evaluieren.

## **5.3. Implikationen**

Im Folgenden werden die Implikationen für Forschung und Praxis vorgestellt, die aus den Ergebnissen dieser Studie abgeleitet wurden.

### **5.3.1. Implikationen für zukünftige Forschung**

Die vorliegende Arbeit beleuchtet vielfältige Aspekte der CT-Implementierung im Grundschulbereich. Es ist jedoch wichtig zu beachten, dass es zahlreiche weitere Faktoren gibt, die über die hier dargestellten hinausgehen. Zudem existieren viele zusätzliche Daten, die für die einzelnen Bedingungen

relevant wären, jedoch aus organisatorischen Gründen nicht in dieser Arbeit berücksichtigt werden konnten.

Zukünftige Forschungen sollten daher die Aspekte, die in dieser Arbeit nicht oder nur unzureichend behandelt wurden, genauer untersuchen (z.B. Elternbeteiligung und Unterstützung, schulinterne Kollaboration und Kommunikation, differenzierte Lehrmethoden, usw.). Diese Untersuchungen könnten dazu beitragen, ein umfassenderes Verständnis der Erfolgsfaktoren und potenziellen Hürden bei der Implementierung von CT zu entwickeln.

Zweitens sollte besondere Aufmerksamkeit Bereichen geschenkt werden, in denen es an validen und einheitlichen Erkenntnissen in der Literatur mangelt. Dazu gehört beispielsweise eine genauere Analyse der Geschlechterrollen im CT-Kontext. Weiterhin müssen Bewertungsmethoden entwickelt werden, um CT entweder als ganzheitliches Maß oder als Reihe von Teilkompetenzen im Kontext authentischer Problemlösungen in allen Fächern und Disziplinen bewerten zu können. Zukünftige Forschungen sollten auch untersuchen, wie CT-Programme benachteiligte Gruppen, wie sozioökonomisch benachteiligte Schüler oder Mädchen, unterstützen können.

Angesichts des luxemburgischen Kontexts ist es besonders wichtig, mehr lokale Studien zur Effektivität von CT-Programmen durchzuführen. Solche Untersuchungen, die speziell auf Luxemburg zugeschnitten sind, können ein tieferes Verständnis der lokalen Implementierungsprozesse und ihrer Wirksamkeit bieten, wodurch passgenauere Schlussfolgerungen gezogen werden können. Dabei sollten Aspekte wie die Effektivität der Lehrerfortbildungen in Luxemburg, die tatsächliche CT-Umsetzung in der Praxis, die Wirksamkeit des eingesetzten CT-Materials oder die Erfolgsfaktoren und Herausforderungen im Zugang zu spezialisierten Lehrkräften analysiert werden. Zudem könnte die Effektivität der in dieser Arbeit vorgeschlagenen Maßnahmen im Kontext Luxemburgs validiert und optimiert werden.

### **5.3.2. Implikationen für die Praxis in Luxemburg**

Diese Arbeit untersucht die fördernden und hindernden Faktoren bei der Implementierung von CT in Grundschulen und zielt darauf ab, spezifische Empfehlungen für den luxemburgischen Schulkontext zu entwickeln. Im Folgenden werden konkrete Handlungsempfehlungen für die wichtigsten Bildungsakteure in Luxemburg vorgestellt. Durch die Umsetzung dieser Empfehlungen könnte die Integration von CT in den luxemburgischen Schulkontext effektiv unterstützt und verbessert werden.

Erstens sollten Bildungspolitiker öffentliche Diskussionen über CT durch Informationskampagnen und Veranstaltungen intensiv fördern, um mehr Aufmerksamkeit und Bewusstsein zu schaffen. Dabei ist es besonders wichtig, den Unterschied zwischen CT und Coding zu verdeutlichen, da die Einführung von CT unter dem Namen „Coding“ zu viel Verwirrung geführt hat. Eine enge und gleichberechtigte Zusammenarbeit zwischen Bildungspolitikern, Schulleitungen und Lehrkräften sollte unterstützt werden, um deren tatsächliche Bedürfnisse, Bedenken und Ideen zu erfassen. Das Ziel ist es, gemeinsam zu lernen und durch Kooperation Bedeutung und Wissen zu entwickeln. Diese Vorgehensweise kann zu einer höheren Akzeptanz und zur erfolgreichen Erreichung der Bildungsziele führen.

Konkrete Richtlinien und gesetzliche Bestimmungen, die zum Beispiel Mindestzeiten für den Informatikunterricht oder zu erreichende Kompetenzen im Bilan, sollten ebenfalls geschaffen werden, um CT einen festen Platz im Lehrplan zu sichern. Solche Vorgaben können die Verbindlichkeit erhöhen und sicherstellen, dass Lehrkräfte regelmäßig CT-Aktivitäten durchführen. Darüber hinaus ist es für Bildungspolitiker entscheidend, ausreichende finanzielle Unterstützung für Universitäten und Schulen sicherzustellen. Finanzielle Mittel und der Zugang zu CT-Infrastruktur spielen eine Schlüsselrolle für den Erfolg der CT-Bildung. In den verschiedenen Regionen des Landes müssen die Verantwortlichen garantieren, dass alle Schulen gleichberechtigten Zugang zu den erforderlichen Ressourcen für die Integration von CT haben.

Für das SCRIPT ist die Entwicklung und Anpassung von konkretem CT-Material, insbesondere unplugged Werkzeugen, von Bedeutung. Außerdem sollten eine kontinuierliche Überprüfung und Bewertung von CT-Praktiken und Materialien gewährleistet werden, um mit der schnellen technologischen Entwicklung Schritt halten zu können.

In Luxemburg, wo traditionell mit Schulbüchern gearbeitet wird, bietet sich beispielsweise eine Anpassung oder Neugestaltung dieser Bücher an, um CT-Unterrichtseinheiten direkt zu integrieren. Diese Integration würde CT nahtlos in den bestehenden Lehrplan einbetten und den Zugang sowie die Anwendung von CT für Lehrkräfte und Lernenden erleichtern. Interdisziplinäre Projekte, die Mathematik, Naturwissenschaften und Sprachen mit informatischen Grundlagen verbinden, könnten hierbei eine strukturierte und kontextualisierte Integration ermöglichen.

Für den Bachelor in Educational Sciences (BScE) an der Universität Luxemburg ist die Anpassung des Curriculums fundamental. Spezifische Module oder Kurse zu CT und dessen Didaktik sollten integriert werden. Obwohl CT offiziell seit einigen Jahren Teil der Lehrerausbildung ist, sollte mehr Wert auf pädagogisches Fachwissen und Methoden für die CT-Bildung gelegt werden. Dabei könnten konkrete Umsetzungsbeispiele zukünftige Lehrkräfte besser auf die Praxis vorbereiten. Dies sollte durch eine erhöhte finanzielle Unterstützung der Regierung sichergestellt werden.

Das Institut de Formation de l'Éducation Nationale (IFEN) sollte die Lehrerfortbildungen stärken. Eine Verpflichtung oder Appell zur Teilnahme an einer Mindestanzahl von CT-Weiterbildungskursen könnte zum Beispiel förderlich sein, um Lehrpersonen für CT zu sensibilisieren. Allgemein sollten die Fortbildungskurse eine fundierte Ausbildung in der Anwendung von Programmierwerkzeugen sowie im Verständnis und in der Vermittlung von CT-Konzepten bieten. Die Programme sollten praxisorientiert sein und konkrete Umsetzungsbeispiele vermitteln, die von den Lehrkräften in verschiedenen Unterrichtsfächern angewendet werden können.

Schließlich ist es entscheidend, dass Lehrpersonen sich ihrer fundamentalen Rolle bei der Integration von CT bewusstwerden. Sie sollten offen für Veränderungen im Lehrplan sein und bereit, sich kontinuierlich weiterzubilden. Zudem ist es wichtig, dass sie eigene Fehler akzeptieren und als Lernmöglichkeiten nutzen, um ihre Unterrichtspraxis stetig zu verbessern. Die erfolgreiche Umsetzung von CT im Unterricht erfordert eine praxisorientierte Herangehensweise und die Bereitschaft, bei Herausforderungen aktiv Unterstützung zu suchen. Außerdem sollten Lehrpersonen darauf achten, keine Stereotypen zu fördern, die die Einstellungen und das Selbstbild der Schülerinnen und Schüler negativ beeinflussen könnten. Stattdessen sollten sie eine inklusive Lernumgebung schaffen, in der alle Schüler, unabhängig von Geschlecht, Hintergrund oder Fähigkeiten, gleichermaßen unterstützt und ermutigt werden.

## 6. Fazit

Die systematische Analyse der Einflussfaktoren zur Implementierung von CT als Schulfach in Grundschulen ergab mehrere wichtige Schlussfolgerungen. Erfolgsfaktoren und Hindernisse wurden in verschiedenen Bereichen identifiziert.

Die Untersuchung zeigte, dass klare Terminologie, gezielte Schulungen, nachhaltige Finanzierung und internationale Zusammenarbeit entscheidend für den Erfolg von CT-Bildungsprogrammen sind. Sowohl strukturelle als auch inhaltliche Maßnahmen sind notwendig, um die erfolgreiche Integration von CT in Grundschulen zu fördern. Durch gezielte Schulungen, den Einsatz geeigneter Lehrmittel und die Schaffung unterstützender Netzwerke kann die CT-Bildung nachhaltig verbessert werden. Die Förderung von Chancengleichheit und Geschlechtergerechtigkeit sowie die Betonung der frühen Vermittlung von CT-Fähigkeiten sind zentrale Elemente für die Weiterentwicklung der Informatikbildung.

Regierungen, Bildungsinstitutionen und Lehrkräfte müssen gemeinsam daran arbeiten, diese Erkenntnisse umzusetzen, um eine zukunftsfähige und inklusive CT-Bildung sicherzustellen. Konkret in Luxemburg sollten Bildungspolitiker öffentliche Diskussionen über CT fördern und gesetzliche Bestimmungen sowie finanzielle Unterstützung sicherstellen. Das SCRIPT sollte konkretes CT-Material entwickeln und kontinuierlich bewerten, während die Anpassung von Schulbüchern eine nahtlose Integration ermöglichen könnte. Für den BScE an der Universität Luxemburg ist eine Curriculum-Anpassung notwendig, um praktische Umsetzungsbeispiele zu integrieren. Das IFEN sollte CT-Lehrerfortbildungen intensivieren, dringende Empfehlungen aussprechen und praxisorientierte Programme anbieten.

Schlussendlich müssen Lehrpersonen offen für Lehrplanänderungen sein, kontinuierlich lernen und eine inklusive Lernumgebung schaffen. Diese Maßnahmen zielen darauf ab, die Integration von CT im luxemburgischen Bildungssystem effektiv zu unterstützen und zu verbessern.

## 7. Literaturverzeichnis

Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-Computer Interaction*, 19, 30–55.

Altanis, I., Retalis, S., & Petropoulou, O. (2018). Systematic design and rapid development of motion-based touchless games for enhancing students' thinking skills. *Education Sciences*, 8(1), 18–32.

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, 105, 105954. <https://doi.org/10.1016/j.chb.2019.03.018>

Ardito, G., Czerkawski, B., & Scollins, L. (2020). Learning computational thinking together: Effects of gender differences in collaborative middle school robotics program. *TechTrends*, 64(3), 373–387.

Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education*, 148. <https://doi.org/10.1016/j.compedu.2020.103807>

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661–670. <https://doi.org/10.1016/j.robot.2015.10.008>

Balanskat A., Engelhardt K., Licht A.H., (2018). Strategies to include computational thinking in school curricula in Norway and Sweden - European Schoolnet's 2018 Study Visit. European Schoolnet.

Balanskat, A., & Engelhardt, K. (2015). Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe. European Schoolnet.

Banilower, E. R., Smith, P. S., Malzahn, K. A., Plumley, C. L., Gordon, E. M., & Hayes, M. L. (2018). Report of the 2018 NSSME+. Horizon Research, Inc.

Barak, M., & Zadok, Y. (2009). Robotics projects and learning concepts in science, technology and problem solving. *International Journal of Technology and Design Education*, 19(3), 289–307. <https://doi.org/10.1007/s10798-007-9043-3>

- Bargagna, S., Castro, E., Cecchi, F., Cioni, G., Dario, P., Dell’Omo, M., Di Lieto, M. C., Inguaggiato, E., Martinelli, A., Peci, C., & Sgandurra, G. (2019). Educational robotics in Down Syndrome: A feasibility study. *Technology, Knowledge and Learning*, 24(2), 315–323. <https://doi.org/10.1007/s10758-018-9366-z>
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is Involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54. <https://doi.org/10.1145/1929887.1929905>
- Bass, K. M., Hu Dahl, I., & Panahandeh, S. (2016). Designing the game: How a project-based media production program approaches STEAM career readiness for underrepresented young adults. *Journal of Science Education and Technology*, 25(6), 1009–1024. <https://doi.org/10.1007/s10956-016-9631-7>
- Basu, S., Biswas, G., & Kinnebrew, J. S. (2017). Learner modeling for adaptive scaffolding in a Computational Thinking-based science learning environment. *User Modeling and UserAdapted Interaction*, 27(1), 5–53. <https://doi.org/10.1007/s11257-017-9187-0>
- Baytak, A., & Land, S. M. (2010). A case study of educational game design by kids and for kids. *Procedia-Social and Behavioral Sciences*, 2(2), 5242–5246. <https://doi.org/10.1016/j.sbspro.2010.03.853>
- Belmar, H. (2022). Review on the teaching of programming and CT in the world. *Frontiers in Computer Science*, 10. 3389/fcomp.2022.997222
- Bers, M. U., González-González, C., & Armas-Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, 138, 130–145. <https://doi.org/10.1016/j.compedu.2019.04.013>
- Bjorklund, D. F. (2018, January). How children invented humanity a metatheory for cognitive development (or “piaget is dead” revisited). *Child Development*, 89(6), 2288–2302. <https://doi.org/10.1111/cdev.13019>
- Bocconi, S., Chiocciariello, A., & Earp, J. (2018). The Nordic approach to introducing computational thinking and programming in compulsory education. Report prepared for the Nordic@BETT2018 Steering Group. <https://doi.org/10.17471/54007>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing Computational Thinking in Compulsory Education. Implications for policy and practice. JRC Science for Policy Report. <https://doi.org/10.2791/792158>



- Bocconi, S., Chiocciariello, A., Kamyliis, P., Dagiené, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M. A., Jasuté, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). Reviewing Computational Thinking in compulsory Education. JRC Publications Repository. <https://doi.org/10.2760/126955>
- Bower, M., Wood, L., Lai, J., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the Computational Thinking Pedagogical Capabilities of School Teachers. *Australian Journal of Teacher Education*, 42(3), 53–72. <https://doi.org/10.14221/ajte.2017v42n3.4>
- Brackmann, C., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. <https://doi.org/10.1145/3137065.3137069>.
- Brown, A. L., Ash, D., Rutherford, M., Nakagawa, K., Gordon, A., & Campione, J. C. (1993). Distributed expertise in the classroom. In G. Salomon (Eds.), *Distributed cognitions: Psychological and educational considerations* (pp.188–228). Cambridge University Press.
- Bundy, A. (2007). Computational Thinking is Pervasive.
- Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education*, 4(2), 121–135.
- Busuttil, L., & Formosa, M. (2020). Teaching computing without computers: Unplugged computing as a pedagogical strategy. *Informatics in Education*, 19(4), 569–587. <https://doi.org/10.15388/infedu.2020.25>
- Cabrera, K., Morreale, P., & Li, J. J. (2018). Computer science+ education: An assessment of CS professional development. *Journal of Computing Sciences in Colleges*, 33(3), 141–147.
- Calabrese Barton, A., & Tan, E. (2018). A longitudinal study of equity-oriented STEM-rich making among youth from historically marginalized communities. *American Educational Research Journal*, 55(4), 761–800. <https://doi.org/10.3102/0002831218758668>
- Caskurlu, S., Yadav, A., Dunbar, K., & Santo, R. (2021). Professional development as a bridge between teacher competencies and computational thinking integration. In A. Yadav & U., Berthelsen (Eds.), *Computational thinking in education* (pp. 136-150). Routledge.
- Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, 17, 93–100.

Chang, C. H. (2019). Does the learning of computational thinking concepts interact with the practice of digital curation in children? A preliminary case study. *Journal of Educational Media and Library Sciences*, 56(1), 45–68.

Chauhan, S. (2017). A meta-analysis of the impact of technology on learning effectiveness of elementary students. *Computers & Education*, 105, 14–30. <https://doi.org/10.1016/j.compedu.2016.11.005>

Chen, P., Yang, D., Metwally, A.H.S. et al. Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. *IJ STEM Ed* 10, 47 (2023). <https://doi.org/10.1186/s40594-023-00434-7>

Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V., & Tosto, C. (2019). Educational robotics in primary school: Measuring the development of computational thinking skills with the Bebras tasks. *Informatics*, 6(4), 43. <https://doi.org/10.3390/informatics6040043>

Chiazzese, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2018). Engaging primary school children in computational thinking: Designing and developing videogames. *Education in the Knowledge Society*, 19(2), 63–81.

Choi, S., Kim, H., Sung, J., Seo, Y., & Choi, Y. (2016). A study of activating Gyeonggido SW education. Korea: Gyeonggido institute of education.

Code.org, CSTA, & ECEP Alliance. (2020). 2020 State of computer science education: Illuminating disparities. Abgerufen am 22.05.2024, von <https://advocacy.code.org/stateofcs>

Computer Science Teachers Association (CSTA) (2013). Bugs in the system: Computer science teacher certification in the U.S.

Crompton, H. (2017). ISTE standards for educators: A guide for teachers and other professionals. International Society for Technology in Education.

Crompton, H., Ulmer, L., Butler, B., & Giannakos, M. (2016). Introducing ISTE's New Essential Conditions Rubric. The International Society for Technology.

Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, 28(6), 651–676. <https://doi.org/10.1007/s10956-019-09799-3>

Delyser, L. A., Goode, J., Guzdial, M., Kafai, Y. B., & Yadav, A. (2018). Priming the Computer Science Teacher Pump: Integrating Computer Science Education into Schools of Education.

Deng, W., Pi, Z., Lei, W., Zhou, Q., & Zhang, W. (2019). Pencil Code improves learners' computational thinking and computer learning attitude. *Computer Application in Engineering Education*, 28(1), 90–104.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, 58(1), 240–249. <https://doi.org/10.1016/j.compedu.2011.08.006>

Denning, P.J., & Tedre, M. (2019) *Computational Thinking*. MIT Press Essential Knowledge Series.

DeVane, B., Steward, C., & Tran, K. M. (2016). Balancing expression and structure in game design: Developing computational participation using studio-based design pedagogy. *Educational Technology*, 56(3), 42–47. <https://www.jstor.org/stable/44430492>.

Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2019). PRADA: A practical model for integrating Computational thinking in K–12 education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 906–912). <https://doi.org/10.1145/3287324.3287431>

Durak, H. Y., Yilmaz, F. G. K., & Bartın, R. Y. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Educational Technology*, 10(2), 173–197.

Egbert, J., Shahrokni, S. A., Abobaker, R., & Borysenko, N. (2021). "It's a chance to make mistakes": Processes and outcomes of coding in 2nd grade classrooms. *Computers & Education*, 168, 104173. <https://doi.org/10.1016/j.compedu.2021.104173>

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. on the iPad. *Journal of Computer Assisted Learning*, 32(6), 576–593. <https://doi.org/10.1111/jcal.12155>

Frydenberg, M. (2015). Achieving digital literacy through game development: An authentic learning experience. *Interactive Technology and Smart Education*, 12(4), 256–269. <https://doi.org/10.1108/ITSE-08-2015-0022>

Gaeta, E., Beltrán-Jaunsaras, M. E., Cea, G., Spieler, B., Burton, A., García-Betances, R. I., Cabrera-Umpiérrez, M. F., Brown, D., Boulton, H., & Arredondo Waldmeyer, M. T.

(2019). Evaluation of the Create@School game-based learning-teaching approach. *Sensors*, 19(15), 3251. <https://doi.org/10.3390/s19153251>

Gal-Ezer, J., & Stephenson, C. (2014). A tale of two countries: Successes and challenges in K-12 computer science education in Israel and the United States. *ACM Transactions on Computing Education*, 14(2), 8.

Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1), 20–20. <https://doi.org/10.1145/950566.950595>

Google. (2015). Searching for Computer Science: Access and Barriers in U.S. K–12 Education.

Grover, S. (2017). Assessing Algorithmic and Computational Thinking in K-12: Lessons from a Middle School Classroom. In Rich, P., & Hodges, C. (Eds.), *Emerging Research, Practice, and Policy on Computational Thinking*. Educational Communications and Technology: Issues and Innovations (pp.269-288). Springer. [https://doi.org/10.1007/978-3-319-52691-1\\_17](https://doi.org/10.1007/978-3-319-52691-1_17)

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43.

Grover, S., & Pea, R. (2018). Computational Thinking: A competency whose time has come. In S. Sentance, E. Barendsen, & S. Carsten (Eds.), *Computer Science Education: Perspectives on Teaching and Learning in School* (pp. 19–38). Bloomsbury Publishing.

Guggemos, J. (2021). On the predictors of computational thinking and its growth at the high-school level. *Computers & Education*, 161. <https://doi.org/10.1016/j.compedu.2020.104060>

Guzdial, M. (2011). A Definition of Computational Thinking from Jeannette Wing. *Computing Education Blog* [Online]. Abgerufen am 21.05.2024, von <http://computinged.wordpress.com/2011/03/22/a-definition-ofcomputational-thinking-from-jeanette-wing/>

Harteis, C. (2018). *The Impact of Digitalization in the Workplace*. Springer.

Hennico, J. (2021). Factors affecting the implementation of computational thinking in the curriculum [Masterthesis]. Saarland University.

- Hsu, Y.-C., Irie, N. R., & Ching, Y.-H. (2019). CT Educational Policy Initiatives (CTEPI) Across the Globe. *TechTrends*, 63(3), 260–270. <https://doi.org/10.1007/s11528-019-00384-4>
- Hug, S., Eyerman, S., Cota, R., & Pontelli, E. (2018). Embedding K12 Professional Development Through Co-Teaching Experiences-Sustaining Computational Thinking in Interdisciplinary Courses. 1-6. 10.1109/RESPECT.2018.8491708.
- Hughes, J., & Morrison, L. (2018). The use of e-textiles in Ontario education. *Canadian Journal of Education*, 41(1), 356–384.
- Information Technology Foundation for Education (HITSA). (2014). ProgeTiger Programme 2015–2020.
- International Association for the Evaluation of Educational Achievement (IEA). (2021). Computational thinking, socioeconomic gaps, and policy implications.
- International Society for Technology in Education (ISTE), & Computer Science Teachers Association (CSTA). (2011). Operational definition of computational thinking for K–12 education.
- International Society for Technology in Education (ISTE). (2023). Essential conditions for effective tech use in schools. Abgerufen am 17. Januar 2024, von <https://iste.org/essential-conditions-for-effective-tech-use-in-schools>
- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 82, 263–279. <https://doi.org/10.1016/j.compedu.2014.11.022>
- Israel-Fishelson, R., Hershkovitz, A., Eguíluz, A., Garaizar, P., & Guenaga, M. (2021). The Associations Between Computational Thinking and Creativity: The Role of Personal Characteristics. *Journal of Educational Computing Research*, 58(8), 1415–1447. <https://doi.org/10.1177/0735633120940954>
- Jacobs, J. E., Finken, L. L., Griffin, N. L., & Wright, J. D. (1998). The career plans of science-talented rural adolescent girls. *American Educational Research Journal*, 35(4), 681–704.
- Kafai, Y. B., Fields, D. A., Roque, R., Burke, W. Q., & Monroy-Hernandez, A. (2012). Collaborative agency in youth online and offline creative production in Scratch. *Research and Practice in Technology Enhanced Learning*, 7(2), 63–87.

Kafai, Y. B., Fields, D., & Burke, W. Q. (2010). Entering the clubhouse: Case studies of young programmers joining the online Scratch communities. *Journal of Organizational and End User Computing*, 22(2), 21–35. <https://doi.org/10.4018/joeuc.2010101906>

Kale, U., Akcaoglu, M., Cullen, T., & Goh, D. (2018). Contextual factors influencing access to teaching computational thinking. *Computers in the Schools*, 35(2), 69– 87. <https://doi-org.uri.idm.oclc.org/10.1080/07380569.2018.1462630>

Kang, E. J. S., Donovan, C., & McCarthy, M. J. (2018). Exploring Elementary Teachers' Pedagogical Content Knowledge and Confidence in Implementing the NGSS Science and Engineering Practices. *Journal of Science Teacher Education*, 29(1), 9–29. <https://doi.org/10.1080/1046560X.2017.1415616>

Karpinski, Z., Biagi, F., & Di Pietro, G. (2021). Computational Thinking, Socioeconomic Gaps, and Policy Implications. IEA Compass: Briefs in Education. Number 12. International Association for the Evaluation of Educational Achievement.

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Social and Behavioural Sciences*, 47, 1991–1999. <https://doi.org/10.1016/j.sbspro.2012.06.938>

Kim, K. (2006). Measuring international research collaboration of peripheral countries: Taking the context into consideration. *Scientometrics*, 66(2), 231–240. <https://doi.org/10.1007/>

Kim, S., & Kim, H.Y. (2018). A CT Curriculum and Teacher Professional Development in South Korea. In Khine, M. (Eds.), *CT in the STEM Disciplines*. Springer, Cham. [https://doi.org/10.1007/978-3-319-93566-9\\_9](https://doi.org/10.1007/978-3-319-93566-9_9)

Koh, K. H., Basawapatna, A., Nickerson, H., & Repenning, A. (2014). Real-time assessment of computational thinking. *Visual Languages and Human-Centric Computing (VL/HCC)*, 2014 IEEE Symposium On (pp. 49–52). IEEE, Piscataway, NJ.

Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, 151, 103872.

Kong, S., & Abelson, H. (2019). *Computational Thinking Education*. Springer Nature.

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558–569. <https://doi.org/10.1016/j.chb.2017.01.005>

Koshy, S., Martin, A., Hinton, L., Scott, A., Twarek, B., & Davis, K., (2021). The computer science teacher landscape: Results from a nationwide survey. The Kapor Center and Computer Science Teachers Association. Abgerufen am 24.05.2024, von <https://www.kaporcenter.org/the-computer-science-teacher-landscape-results-of-a-nationwide-teacher-survey/>

Lee, J. M., Jung, Y. J., & Park, H. K. (2017). Gender differences in computational thinking, creativity, and academic interest on elementary SW education. *Journal of the Korean Association of Information Education.*,21(4), 381–391.

Lee, S. J., Francom, G. M., & Nuatomue, J. (2022). Computer science education and K-12 students' computational thinking: A systematic review. *International Journal of Educational Research*, 114, 102008. <https://doi.org/10.1016/j.ijer.2022.102008>.

Li, Q. (2020). Computational thinking and teacher education: An expert interview study. *Human Behavior and Emerging Technologies*, 1–15. <https://doi.org/10.1002/hbe2.224>

Liu, J., Lin, C. H., Hasson, E. P., & Barnett, Z. D. (2011). Introducing computer science to K-12 through a summer computing workshop for teachers. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 389-394). ACM.

Luxembourg Centre for Educational Testing (LUCET), Universität Luxemburg & Service de Coordination de la Recherche et de l'Innovation pédagogiques et technologiques. (2021). *Nationaler Bildungsbericht Luxemburg 2021*.

Martin, A., Koshy, S., Hinton, L., Scott, A., Twarek, B., & Davis, K. (2020). Teacher perspectives on COVID-19's impact on K–12 computer science instruction. Kapor Center.

Menekse, M. (2015). Computer science teacher professional development in the United States: A review of studies published between 2004 and 2014. *Computer Science Education*, 25(4), 325–350.

Merino-Armero, J. M., González-Calero, J. A., & Cózar-Gutiérrez, R. (2022). Computational thinking in K-12 education. An insight through meta-analysis. *Journal of Research on Technology in Education*, 54(3), 410–437. <https://doi.org/10.1080/15391523.2020.1870250>

Mills, K. A., Cope, J., Scholes, L., & Rowe, L. (2024). Coding and Computational Thinking Across the Curriculum: A Review of Educational Outcomes. *Review of Educational Research*, 0(0). <https://doi.org/10.3102/00346543241241327>

Mills, K., Coenraad, M., Ruiz, P., Burke, Q., & Weisgrau, J. (2021). CT for an Inclusive World: A Resource for Educators to Learn and Lead. *Digital Promise*. <https://doi.org/10.51388/20.500.12265/138>

Milmeister, F., & Baumann, A. (2021). Das one2one-Programm in Luxemburger Sekundarschulen: Vom Ausstattungsziel hin zu Kompetenzziele. In Luxembourg Centre for Educational Testing, Universität Luxemburg, Service de Coordination de la Recherche et de l'Innovation pédagogiques et technologiques (Eds.), *Nationaler Bildungsbericht Luxemburg 2021* (S.155-175). Athena-Verlag.

Min, S. H., & Kim, M. K. (2020). Developing children's computational thinking through physical computing lessons. *International Electronic Journal of Elementary Education*, 13(2), 183–198.

Minamide, A., Takemata, K., & Yamada, H. (2020). Development of computational thinking education system for elementary school class. In *Proceedings of the IEEE 20th international conference on advanced learning technologies (ICALT)* (pp. 22–23). <https://doi.org/10.1109/ICALT49669.2020.00013>

Ministère de l'Éducation nationale et de la Formation professionnelle (MENFP). (2011). *Plan d'études - Ecole fondamentale*.

Ministère de l'Éducation nationale, de l'Enfance et de la Jeunesse (MENJE). (2021). *Lettre circulaire de printemps 2021*.

Mishra, P., & Henriksen, D. (2012). Rethinking technology & creativity in the 21st century: On being in-disciplined. *TechTrends*, 56(6), 18–21. <https://doi.org/10.1007/s11528-012-0608-y>

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for integrating technology in teacher knowledge. *Teachers College Record*, 108(6), 1017–1054.

Moore, T. J., Brophy, S. P., Tank, K. M., Lopez, R. D., Johnston, A. C., Hynes, M. M., & Gajdzik, E. (2020). Multiple representations in computational thinking tasks: A clinical study of second-grade students. *Journal of Science Education and Technology*, 29(1), 19–34. <https://doi.org/10.1007/s10956-020-09812-0>



Moreno-León, J., & Robles, G. (2015). Dr. Scratch: A Web Tool to Automatically Evaluate Scratch Projects. In Proceedings of the Workshop in Primary and Secondary Computing Education (pp. 132–133). <https://doi.org/10.1145/2818314.2818338>

Moreno-León, J., Román-González, M., and Robles, G. (2018). "On computational thinking as a universal skill: a review of the latest research on this ability," in 2018 IEEE Global Engineering Education Conference (EDUCON) (Santa Cruz de Tenerife: IEEE), 1684–1689. <https://doi.org/10.1109/EDUCON.2018.8363437>

National Center for Women & Information Technology. (2018). By the numbers.

National Research Council (NRC), Division of Behavioral and Social Sciences and Education, Board on Science Education, Committee on a Conceptual Framework for New K-12 Science Education Standards, Keller, T., Quinn, H., & Schweingruber, H. (2012). A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas. National Academies Press.

Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, 68(1), 463–484.

Papavasopoulou, S., Sharma, K., & Giannakos, M. (2019). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*, 105. <http://dx.doi.org/10.1016/j.chb.2019.03.003>

Papert, S. (1971). Teaching children thinking (LOGO memo). Massachusetts Institute of Technology, A.I: Laboratory.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Parmentier, Y., Reuter, R., Higuete, S., Kataja, L., Kreis, Y., Duflot-Kremer, M., Laduron, C., Meyers, C., Busana, G., Weinberger, A., & et al. (2020). PIAF: Developing Computational and Algorithmic Thinking in Fundamental Education. In Proceedings of EdMedia + Innovate Learning 2020 (Vol. 1, S.315-322). Association for the Advancement of Computing in Education (AACE).

Parmentier, Y., Reuter, R., Higuete, S., Kataja, L., Kreis, Y., Duflot-Kremer, M., Laduron, C., Meyers, C., Busana, G., Weinberger, A., et al. (2021). PIAF: développer la Pensée Informatique et Algorithmique dans l'enseignement Fondamental: Référentiel de compétences. Version 2.

Parry, R., Howard, F., & Penfold, L. (2020). Negotiated, contested and political: the disruptive third spaces of youth media production. *Learning, Media, and Technology*, 45(4), 409–421. <https://doi.org/10.1080/17439884.2020.1754238>

Piaget, J. (1952). *The origins of intelligence in children*. International University Press.

Piech, C., Sahami, M., Koller, D., Cooper, S., & Blikstein, P. (2012). Modeling how students learn to program. In *Proceedings of the 43rd ACM technical symposium on computer science education* (pp. 153–160). ACM.

Pietros, J. (2022). *Computational thinking in elementary science [Dissertation]*. University of Rhode Island. <https://doi.org/10.23860/diss-pietros-jennifer-2022>

Price, C. B., & Price-Mohr, R. M. (2018). Stories children write while coding: a crossdisciplinary approach for the primary classroom. *Cambridge Journal of Education*, 48(6), 735–747. <https://doi.org/10.1080/0305764X.2017.1418834>

Ratcliff, C. C., & Anderson, S. E. (2011). Reviving the turtle: Exploring the use of Logo with students with mild disabilities. *Computers in the Schools*, 28(3), 241–255. <https://doi.org/10.1080/07380569.2011.594987>

Repenning, A., Grover, R., Gutierrez, K., Repenning, N., Webb, D. C., Koh, K. H., Nickerson, H., Miller, S. B., Brand, C., Horses, I. H. M., Basawapatna, A., & Gluck, F. (2015). Scalable game design. *ACM Transactions on Computing Education*, 15(2), 1–31. <https://doi.org/10.1145/2700517>

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., & Silver, J. (2009). Scratch. *Communications of the ACM*, 52(11), 60–67. <https://doi.org/10.1145/1592761.1592779>

Reuter, C. (2020). *Computational Thinking in der Grundschule [Masterthesis]*. Donau Universität Krems, Austria.

Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education*, 27(2), 165–205. <https://www.learntechlib.org/primary/p/207487/>

Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational Thinking in Primary School: An examination of Abstraction and Decomposition in Different Age Groups. *Informatics in Education*, 17(1), 77–92. <https://doi.org/10.15388/infedu.2018.05>

Rings, E. (2020). „Umstellung der Lehrer erfordert“ Interview Claude Meisch über die Herausforderungen bei der Einführung digitaler Kompetenzen in den Schulen. *Tageblatt*, 32.

Rodríguez-Martínez, J. A., Gonzalez-Calero, J. A., & Saez-Lopez, J. M. (2020). Computational thinking and mathematics using Scratch: An experiment with sixth-grade students. *Interactive Learning Environments*, 28(3), 316–327.

Rojas, A., & García, F. J. (2020). Assessment of computational thinking for learning computer programming in higher education. *Journal of Distance Education*, 20, 1–36. <https://doi.org/10.6018/red.409991>

Román-González, M. R. (2015). Computational thinking test: Design guidelines and content validation. In *Proceedings of EDULEARN15 conference*, July (pp. 2436–2444). <https://doi.org/10.13140/RG.2.1.4203.4329>

Román-González, M. R., Moreno-León, J., & Robles, G. (2017b). Complementary Tools for Computational Thinking Assessment. In S. C Kong, J Sheldon, and K. Y Li (Eds.), *Proceedings of International Conference on Computational Thinking Education - CTE 2017* (pp. 154–159). The Education University of Hong Kong.

Román-González, M. R., Pérez-González, J. C., & Jiménez-Fernández, C. (2017a). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior*, 72, 678-691.

Román-González, M. R., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018a). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-computer Interaction*, 18, 47–58. <https://doi.org/10.1016/j.ijcci.2018.06.004>

Román-González, M. R., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018b). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, 80, 441–459. <https://doi.org/10.1016/j.chb.2017.09.030>

Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two-year case study using “Scratch” in five schools. *Computers & Education*, 97, 129–141. <https://doi.org/10.1016/j.compedu.2016.03.003>

Salac, J., Thomas, C., Butler, C., Sanchez, A., & Franklin, D. (2020). TIPP&SEE: A Learning Strategy to Guide Students through Use - Modify Scratch Activities. *Proceedings of the*

51st ACM Technical Symposium on Computer Science Education, 79–85.  
<https://doi.org/10.1145/3328778.3366821>

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K–12: In-service teacher perceptions of computational thinking. In *Computational thinking in the STEM disciplines* (pp. 151-164). Springer, Cham.

Saxena, Anika & Lo, Chung Kwan & Hew, Khe & Wong, Gary. (2020). Designing Unplugged and Plugged Activities to Cultivate CT: An Exploratory Study in Early Childhood Education. *The Asia-Pacific Education Researcher*. 29. <https://doi.org/10.1007/s40299-019-00478-w>.

Selby, C., & Woollard, J. (2014). Refining an understanding of computational thinking.

Sembill, D., Schumacher, L., Wolf, K. D., Wuttke, E., & Santjer-Schnabel, I. (2001). Förderung der Problemlösefähigkeit und der Motivation durch Selbstorganisiertes Lernen. In K. Beck & V. Krumm (Eds.), *Lehren und Lernen in der beruflichen Erstausbildung* (S. 257–281). Leske & Budrich.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher’s perspective. *Education & Information Technologies*, 22(2), 469–495. <https://doi.org/10.1007/s10639-016-9482-0>

Sentance, S., & Humphreys, S. (2018). Understanding professional learning for computing teachers from the perspective of situated learning. *Computer Science Education*, 28(4), 345–370.

Seow, P., Looi, CK., How, ML., Wadhwa, B., Wu, LK. (2019). Educational Policy and Implementation of CT and Programming: Case Study of Singapore. In: Kong, SC., Abelson, H. (Eds.), *CT Education*. Springer. [https://doi.org/10.1007/978-981-13-6528-7\\_19](https://doi.org/10.1007/978-981-13-6528-7_19)

Service de Coordination de la Recherche et de l’Innovation pédagogiques et technologiques (SCRIPT). (2020). Einfach Kodieren.

Service de Coordination de la Recherche et de l’Innovation pédagogiques et technologiques (SCRIPT). (2022). *Medienkompass: Medienkompetent lehren und lernen*. Einfach digital.

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, 15, 4–14.

Sun, L., Guo, Z., & Hu, L. (2021). Educational games promote the development of students’ computational thinking: A meta-analytic review. *Interactive Learning Environments*. <https://doi.org/10.1080/10494820.2021.1931891>

- Syslo, M. M., & Kwiatkowska, A. B. (2015). Introducing a new computer science curriculum for all school levels in Poland. In A. Brodник & J. Vahrenhold (Eds.), *Informatics in schools: Curricula, competences, and competitions* (pp. 141–154). Springer.
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798. <https://doi.org/10.1016/j.compedu.2019.103798>
- Taylor, K., & Baek, Y. (2019). Grouping matters in computational robotic activities. *Computers in Human Behavior*, 93, 99–105.
- Tekdal, M. (2021). Trends and development in research on computational thinking. *Education And Information Technologies*, 26(5), 6499–6529. <https://doi.org/10.1007/s10639-021-10617-w>
- The Curriculum Development Council. (2017). *Technology education: Key learning area curriculum guide (Primary 1 – Secondary 6)*.
- Theodosiou, S., & Karasavvidis, I. (2015). Serious games design: A mapping of the problems novice game designers experience in designing games. *Journal of e-Learning and Knowledge Society*, 11(3), 133–148. <https://www.learntechlib.org/p/151929/>
- Tonbuloğlu, B., & Tonbuloğlu, I. (2019). The effect of unplugged coding activities on computational thinking skills of middle school students. *Informatics in Education*, 18(2), 403–426. <https://doi.org/10.15388/infedu.2019.19>
- Toyama, K. (2015). *Geek Heresy*. New York: PublicAffairs.
- Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research*, 57(1), 3–31.
- Tsarava, K., Moeller, K., Román-González, M., Golle, J., Leifheit, L., Butz, M., & Ninaus, M. (2022). A cognitive definition of computational thinking in primary education. *Computers & Education*, 179, <https://doi.org/10.1016/j.compedu.2021.104425>.
- United Arab Emirates Ministry of Education. (2015). *UAE K-12 computer science and technology standards [Electronic version]*. Dubai, United Arab Emirates: United Arab Emirates Ministry of Education.

Vakil, S. (2018). Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review*, 88(1), 26–52. <https://doi.org/10.17763/1943-5045-88.1.26>

Vegas, E., & Fowler, B. (2020). What do we know about the expansion of K–12 computer science education? Brookings.

Villavicencio, A., Fancsali C., Martin, W., Mark, J., & Cole, R. (2018). Computer science in New York City: An early look at teacher training opportunities and the landscape of CS implementation in schools. The Research Alliance for New York City Schools.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728.

Vygotsky, L. S. (1994). The development of academic concepts in school aged children. In Van der Veer, R., & Valsiner, J. (Eds.), *The Vygotsky reader* (pp. 355–370). Blackwell.

Washington, A. N. (2020). When twice as good isn't enough: The case for cultural competence in computing. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 213–219). <https://doi.org/10.1145/3328778.336679>

Weintrop, D., Rutstein, D. W., Bienkowski, M. & McGee, S. (2021a). Assessing computational thinking: an overview of the field. *Computer Science Education*, 31(2), 113–116. <https://doi.org/10.1080/08993408.2021.1918380>

Weintrop, D., Rutstein, D. W., Bienkowski, M. & McGee, S. (2021b). Assessing computational thinking. In A. Yadav & U., Berthelsen (Eds.), *Computational thinking in education* (pp. 90-111). Routledge.

Wenzlik, A. (2013). Schlüsselkompetenzen in der Kulturellen Bildung. kubi-online. Abgerufen am 12. Dezember 2023, von <https://www.kubi-online.de/artikel/schluesselkompetenzen-kulturellen-bildung>

Wiebe, E., Mott, B. W., London, J., Boyer, K. E., Aksit, O., & Lester, J. C. (2019). Development of a lean computational thinking abilities assessment for middle grades students. In *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 456–461). <https://doi.org/10.1145/3287324.3287390>

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>

- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>
- Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C., & Shoop, R. (2017). Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education*, 18(1). <https://doi.org/10.1145/3104982>
- Witherspoon, E. B., Schunn, C. D., Higashi, R. M., & Shoop, R. (2018). Attending to structural programming features predicts differences in learning and motivation. *Journal of Computer Assisted Learning*, 34(2), 115–128.
- Women In Digital Empowerment (WIDE). (o.D.). Digital skills in education. Digital Skills and Jobs Coalition Luxembourg. Abgerufen am 24.05.2024, von <https://digitalskills.lu/digital-skills-in-education/>
- Wong, G. K.-W., & Cheung, H.-Y. (2018). Exploring children’s perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, 28(4), 438–450.
- Wood, D., Bruner, J. S., & Ross, G. (2006). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2), 89–100. <https://doi.org/10.1111/j.1469-7610.1976.tb00381.x>
- World Economic Forum. (2023). *The Future of Jobs Report 2023*.
- Xing, Y., & Zhang, Y. (2020). International Comparative Study on CT Education in K-12. *The Educational Review, USA*, 4(8), 166–175. <https://doi.org/10.26855/er.2020.08.002>
- Xu, E., Wang, W., & Wang, Q. (2023). A meta-analysis of the effectiveness of programming teaching in promoting K-12 students’ computational thinking. *Education and Information Technologies*, 28(6), 6619–6644. <https://doi.org/10.1007/s10639-022-11445-2>
- Yadav, A., et al. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1), 1–16. <https://doi.org/10.1145/2994591>
- Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. (2011). Introducing Computational Thinking in Education Courses. 465-470. <https://doi.org/10.1145/1953163.1953297>