



Debugging im Unterricht

Prof. Dr. Tilman Michaeli
tilman.michaeli@tum.de
07.07.2023

Computing Education Research Group @TUM



Debugging im
Informatikunterricht



Data & AI Literacy



Kollaboration in
Programmier-
projekten



Digitale Bildung



Quanteninformatik



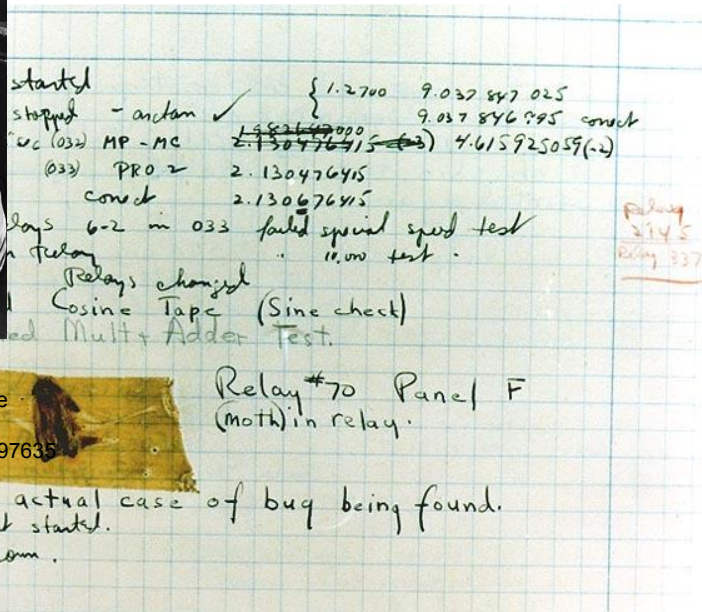
ComputingEducation
Research Group Munich



Mythos "Fehler in der Informatik"?

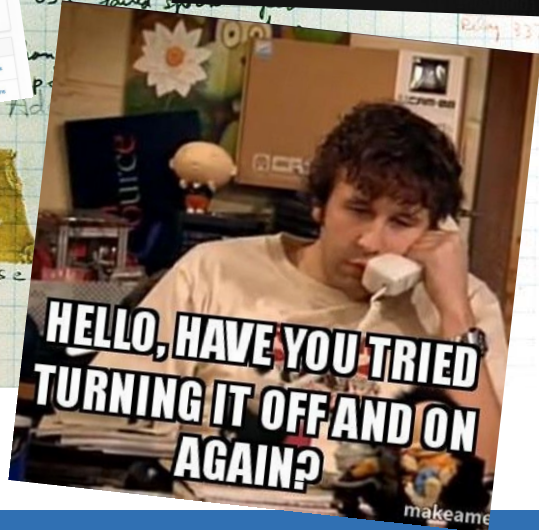
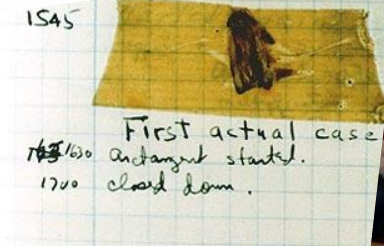
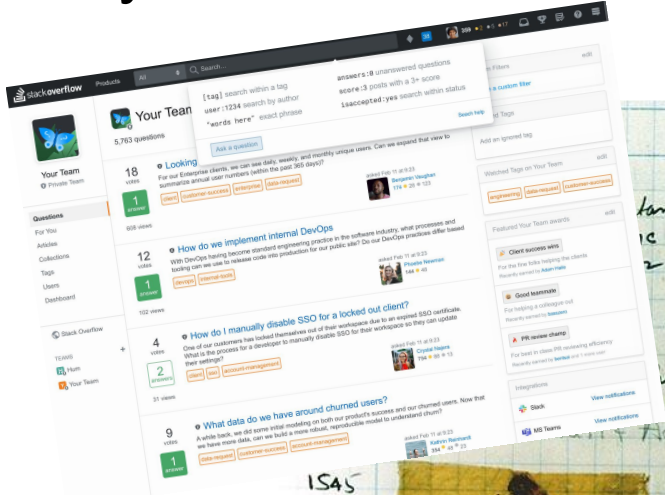
As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs.

Maurice Wilkes, 1949



Von Unknown (Smithsonian Institution) - Flickr: Grace Hopper and UNIVAC, CC BY 2.0, <https://commons.wikimedia.org/w/index.php?curid=197635>

Mythos "Fehler in der Informatik"?



Von Phrd/Stahlkocher, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=6114217>



Debugging?



Debugging is the process of determining **why** a given set of inputs causes an unacceptable behavior in a program and **what must be changed** to cause the behavior to be acceptable.

Testing is the process of determining **whether** a given set of inputs causes an unacceptable behavior in a program.

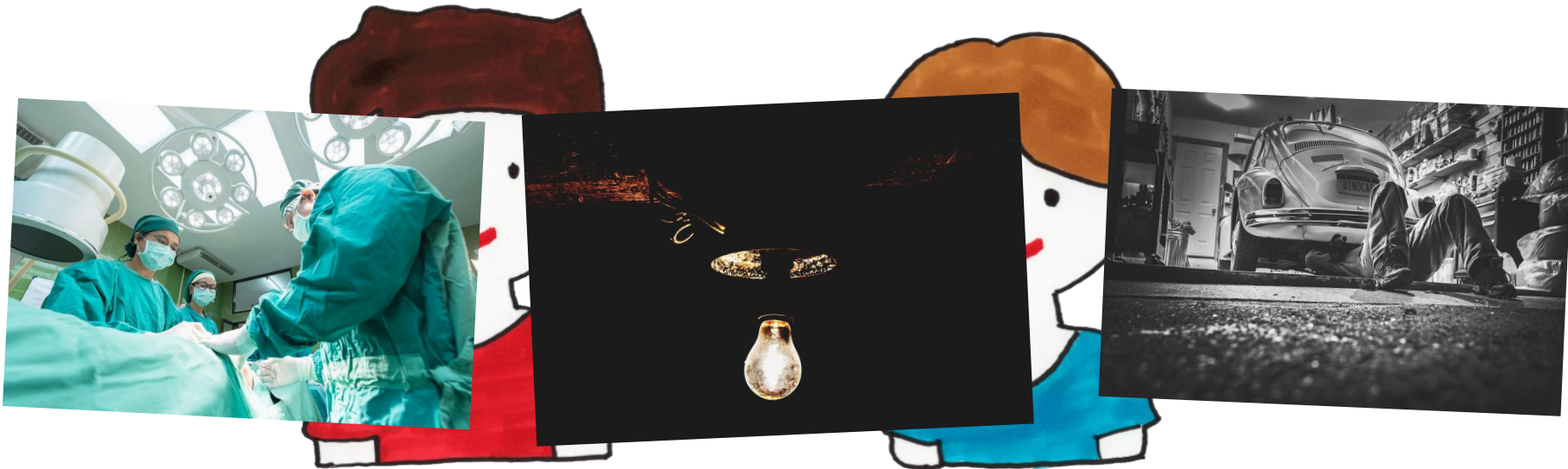
(Metzger, 2004)

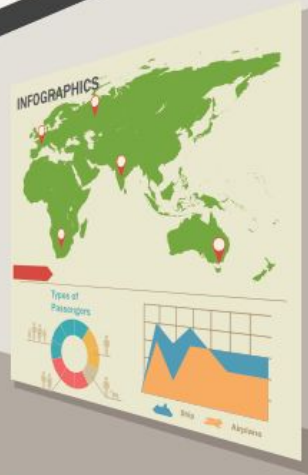
Computational Thinking & Troubleshooting

Das Internet ist kaputt!!!

Hm...

Ist mein LAN-Kabel verbunden?
Sind auch andere Geräte im Netzwerk betroffen?





Programmierfähigkeiten unterscheiden sich von Debuggingfähigkeiten

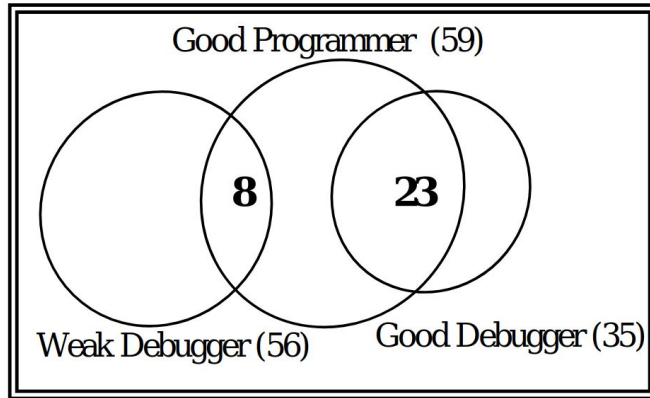


Figure 1: The relationship between Students' Programming and Debugging Ability

Ahmadzadeh et al., 2005

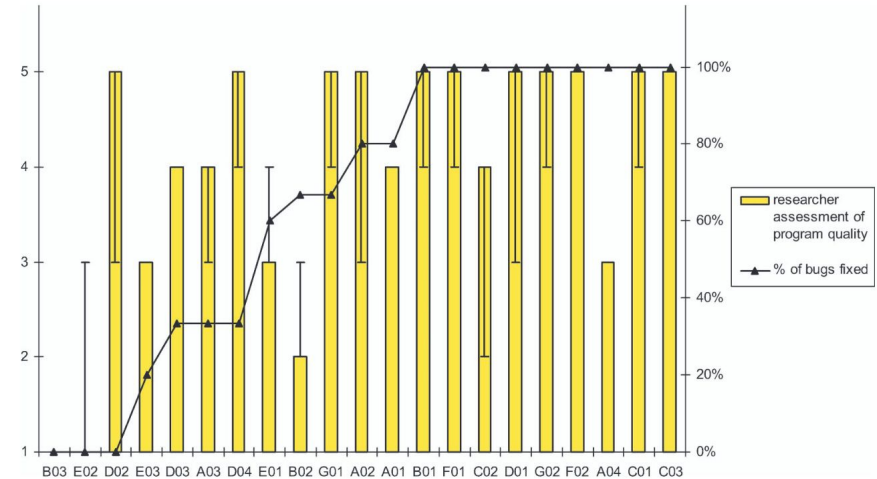


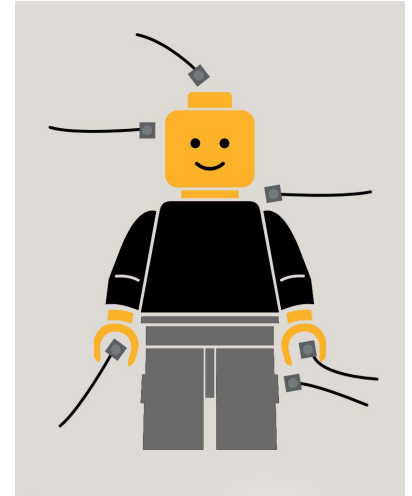
Figure 4. Programming ability versus bugs fixed. Student self-assessment of programming ability shown in error bars.

Fitzgerald et al., 2009



Programmieranfänger:innen

- erleben Debugging als großes Hindernis beim Programmierenlernen
- verbinden Debugging mit Frustration und nehmen es als persönliches Versagen wahr



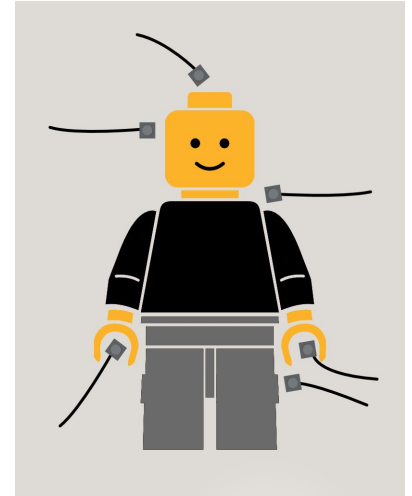


What we see as a good program with a small bug, the child sees as “wrong, ” “bad,” “a mistake.” School teaches that errors are bad; the last thing one wants to do is pore over them, dwell on them, or think about them. The child is glad to take advantage of the computer’s ability to erase it all without any trace for anyone to see. The debugging philosophy suggests an opposite attitude. Errors benefit us because they lead us to study what happened, to understand what went wrong, and, through understanding, to fix it.

(Papert, 1980)

Programmieranfänger:innen

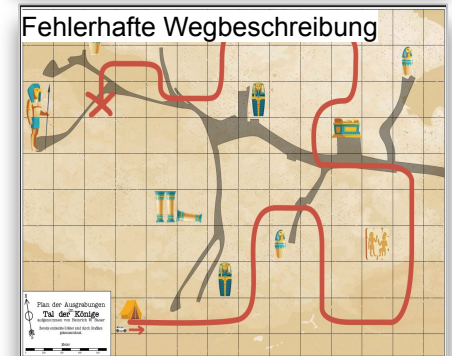
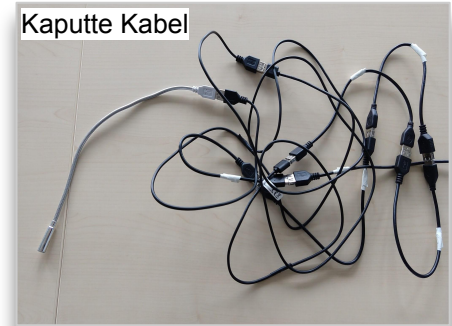
- erleben Debugging als großes Hindernis beim Programmierenlernen
- verbinden Debugging mit Frustration und nehmen es als persönliches Versagen wahr
- Viel trial-and-error
- Angewohnheit, neue Fehler einzubauen
- Probleme im Umgang mit Fehlermeldungen



Lernvoraussetzungen zum Debugging

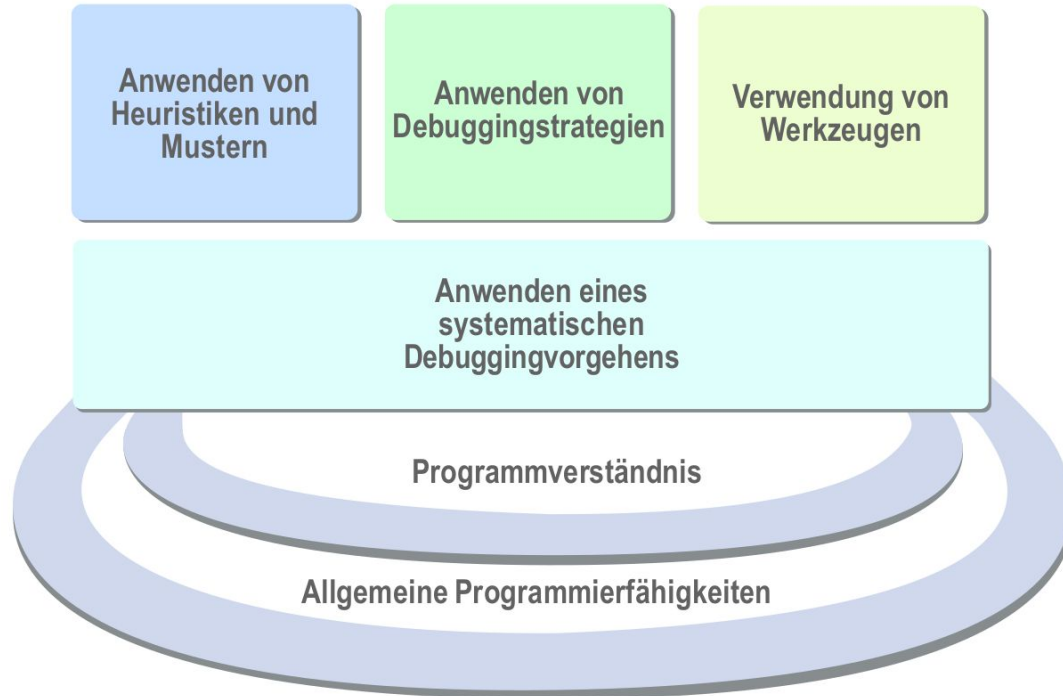
- Probleme mit der Generierung von **Hypothesen**
- Keine Strategien um ein System / einzelne Komponenten effektiv zu **testen**
- Probleme mit dem **Tracing** des Systems (cognitive load / externe Hilfsmittel)
- Probleme mit dem **Rückgängigmachen** erfolgloser Änderungen

(Michaeli et al., 2020)





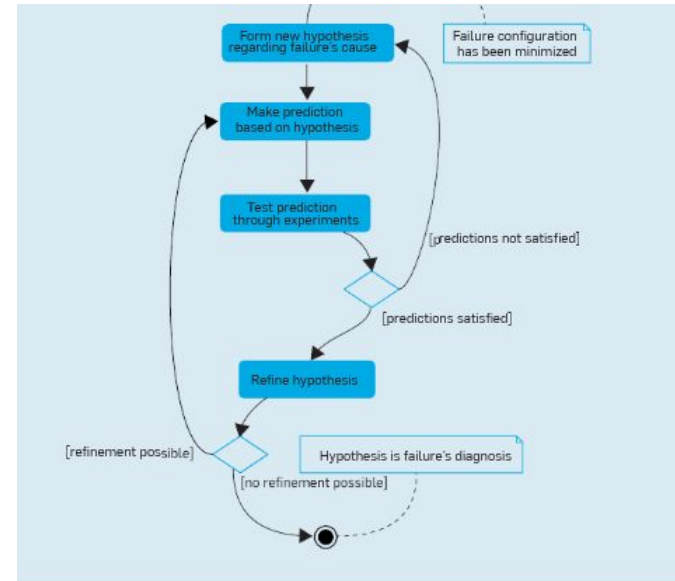
Debuggingfähigkeiten für Programmieranfänger:innen



Anwenden eines systematischen Debuggingvorgehens

“Wissenschaftliche Methode”:

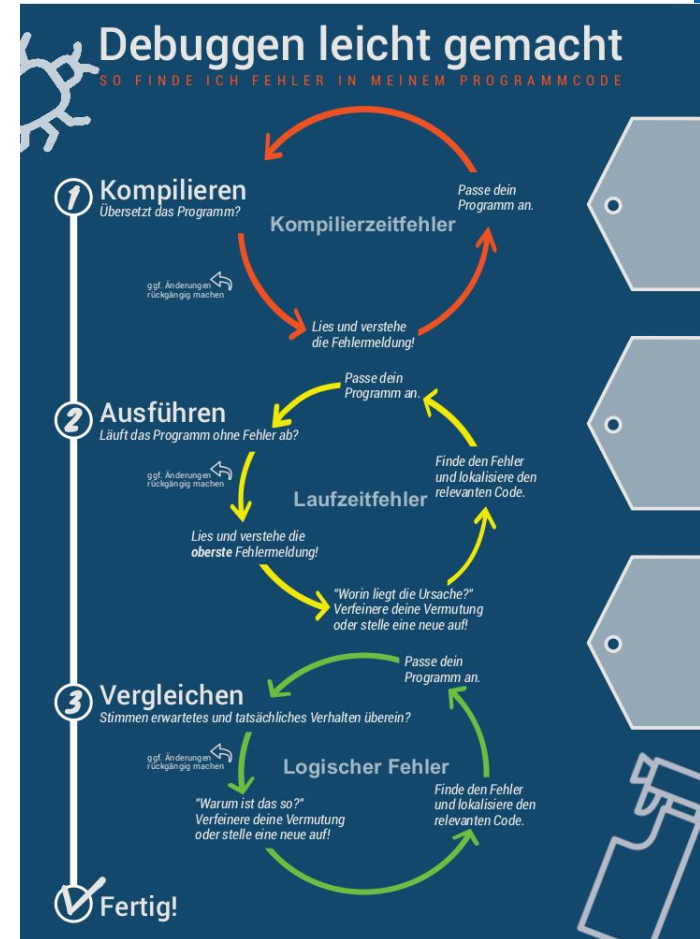
1. Testen und Fehler reproduzieren.
2. Falls nötig, Überblick über das Programm gewinnen.
3. Eine Hypothese formulieren.
4. Hypothese experimentell überprüfen.
5. Falls nötig, Hypothese verfeinern/anpassen.
6. Den Fehler beheben.
7. Das Programm erneut testen.





... für Schüler:innen

- Didaktisch adaptiertes systematisches Debuggingvorgehen
- Unterscheidung nach Fehlertypen
- Betonen von Undo
- Alternative zu Trial-and-Error-Vorgehen





Anwenden von Debuggingstrategien

- Auskommentieren/Slicing
- Logging und Tracing (z.B. print-debugging, Tracetabelle)
- Assertions
- Testen
- Informationen sammeln (z.B. API)
- Hilfe suchen (z.B. StackOverflow, “Rubberduck”)



Verwendung von Werkzeugen

- Debugger
- IDE-Feedback

Anwenden von Heuristiken und Mustern

- Debugging-Logs/Diaries



Debugging im Unterricht?

Unterrichtspraxis:

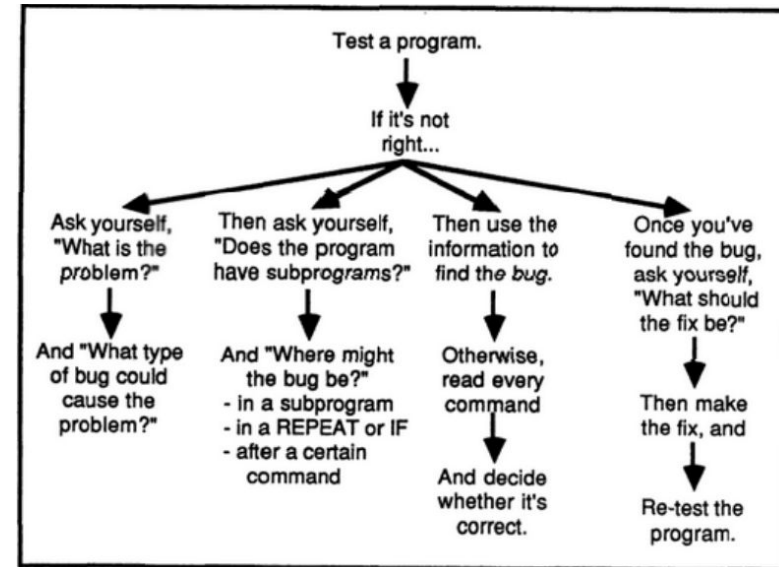
- Kein expliziter Inhalt von Curricula
- Wie haben Lehrkräfte debuggen “gelernt”? :)
- Typischer Ansatz: (Unsystematisches) Üben mit Hilfe von Debuggingaufgaben

(Michaeli, 2019a)

Debuggingfähigkeiten entwickeln sich nur eingeschränkt “von selbst”.

- Carver (1986) & Carver und Risinger (1987):
Vorgehen für LOGO

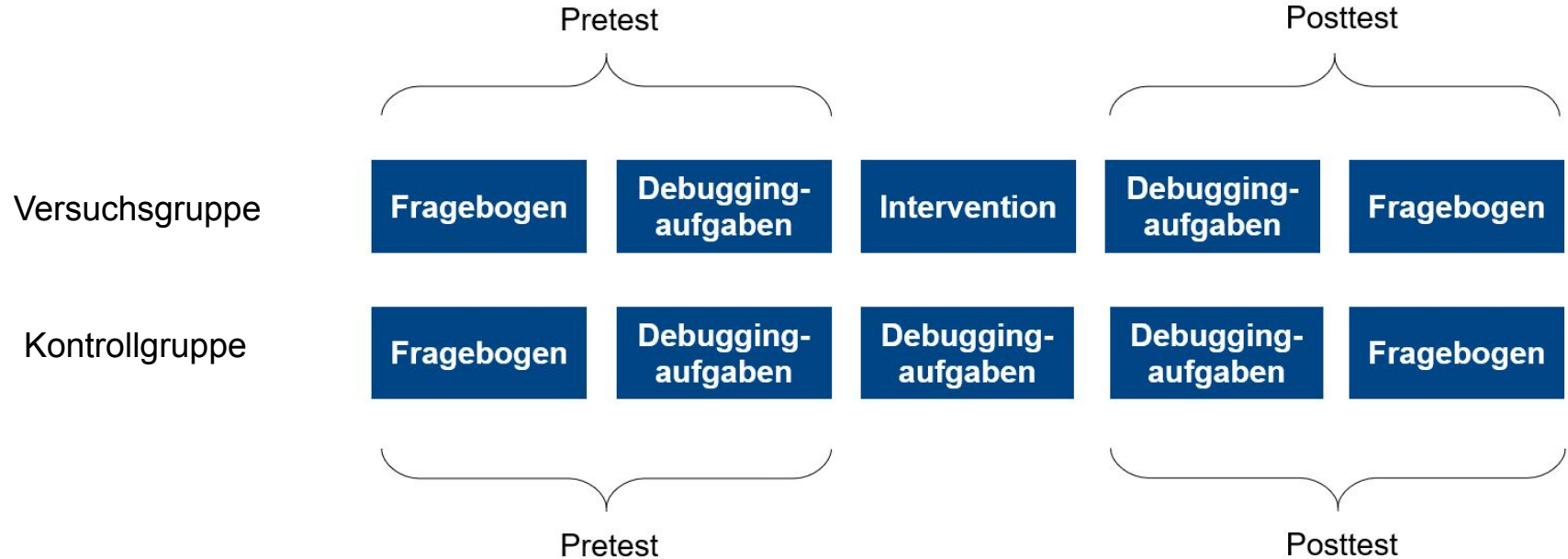
Welchen Effekt hat die explizite
Vermittlung von Debuggingfähigkeiten?



McCoy Sharon Carver and Sally Clarke Risinger. 1987. Improving children's debugging skills. In Empirical studies of programmers: Second workshop. Ablex Publishing Corp., Norwood, NJ, USA, 147-171



Studiendesign



RQ1: Selbstwirksamkeitserwartungen

Pre-Post-Vergleich innerhalb der Gruppen; Wilcoxon-Vorzeichen- Rang-Tests

	Median pre	Median post	Wilcoxon-Test
Pilotgruppe	2,75	3,25	$p = 0,044^*$
Kontrollgruppe	2,25	2,50	$p = 0,083$
Versuchsgruppe	2,25	2,75	$p = 0,001^*$

Tab. 1: Einfluss auf Selbstwirksamkeitserwartungen

- Signifikanter Anstieg in Pilot- und Versuchsgruppe (auf $\alpha = 0.05$)
- Mittlere Effektstärke nach Cohen ($d = 0,54$ und $d = 0,56$)

(Michaeli, 2019b)

RQ2: Debuggingleistung

Vergleich von Versuchsgruppe und Kontrollgruppe Pre und Post; Mann-Whitney-U-Tests

	Mann-Whitney-U-Test
Versuch- vs. Kontrollgruppe Pre	$p = 0,191$
Versuch- vs. Kontrollgruppe Post	$p = 0,049^*$

Tab. 2: Einfluss auf Debuggingleistung

- Pre: SuS kommen aus derselben Grundgesamtheit
- Post: Signifikant höhere Debuggingleistung in Versuchsgruppe (Median 4 vs. 2)
- Mittlere Effektstärke nach Cohen ($d = 0.69$)

(Michaeli, 2019b)





FOAM

FIRE EXTINGUISHER

100% AFFF

FIRE GUARD

USE UPRIGHT
PULL OUT
SAFETY PIN / CLIP

AIM NOZZLE AT BASE
OF FIRE FROM A MINIMUM
DISTANCE OF 1 METRE

SQUEEZE HANDLES

1 A B

APPROX. 10 LITRES

CE

FIRE SAFETY STORE

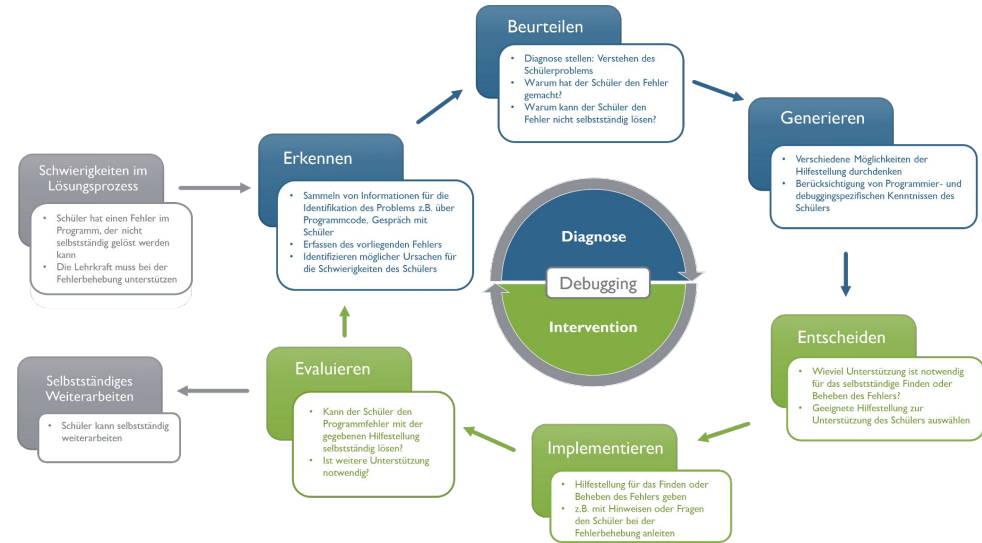
Herausforderung für Lehrkräfte

1. Problem diagnostizieren

- Warum hat die Schüler:in den Fehler gemacht?
- Warum kann die Schüler:in den Fehler nicht selbstständig lösen?

2. Auswahl einer geeigneten Intervention

- auf Wissensstand der Schüler:innen zugeschnitten
- den Fehler beheben
- Förderung der Selbstständigkeit bei der Fehlersuche



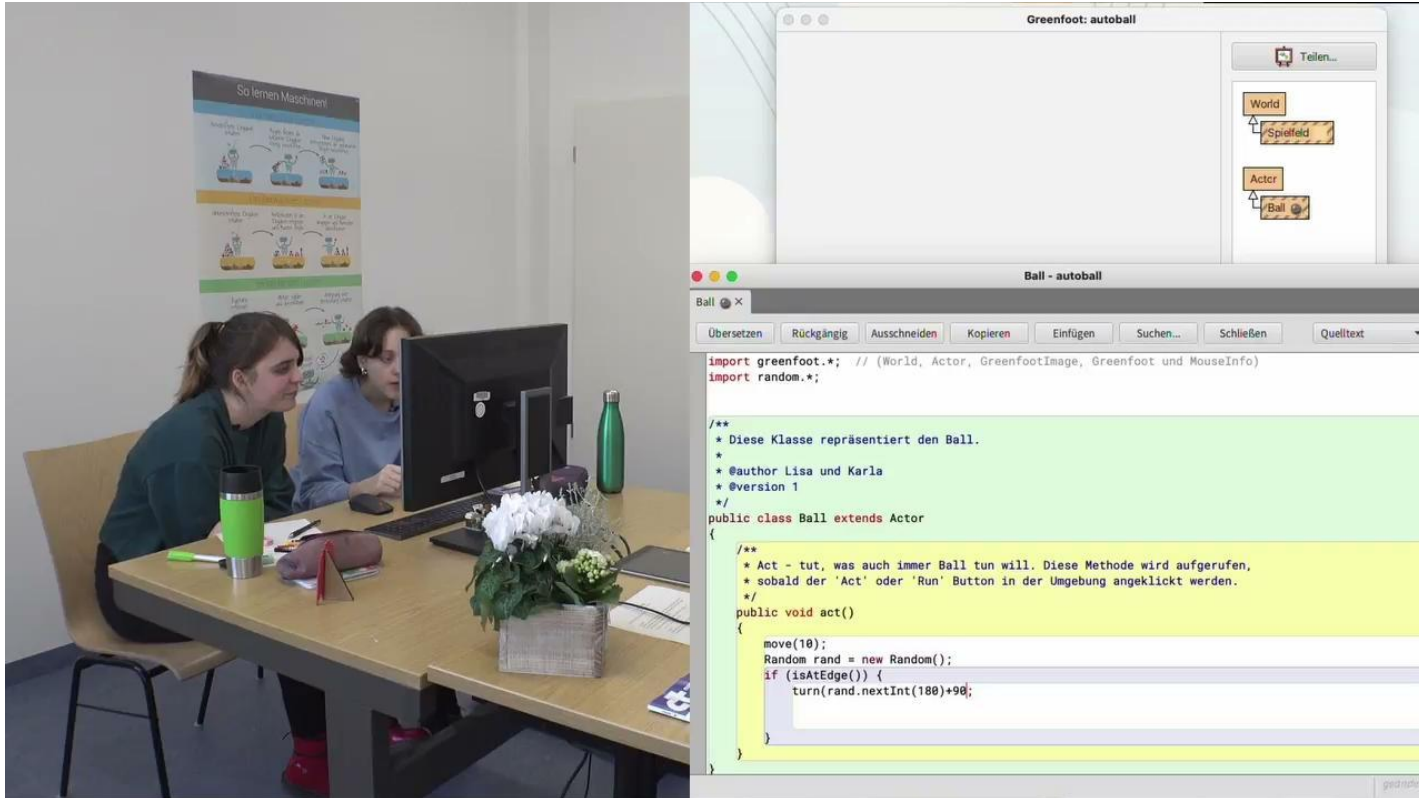
(Hennig & Michaeli, 2023)



Was wir wissen

- Vor allem angehende Lehrkräfte haben häufig in den ersten Jahren ihrer Lehrtätigkeit Schwierigkeiten bei Diagnoseprozessen [Codreanu et al., 2020]
- Verwendung von geskripteten Videovignetten ermöglicht repräsentationales Scaffolding: Vereinfachte Darstellung der komplexen Situation im Klassenzimmer [Sherin and Van Es, 2009; Seidel et al. 2022]
- In anderen Fachdidaktiken erfolgreich in der Lehrkräfteausbildung eingesetzt [Chernikova et. al, 2020]

Vignetten





Erste Ergebnisse

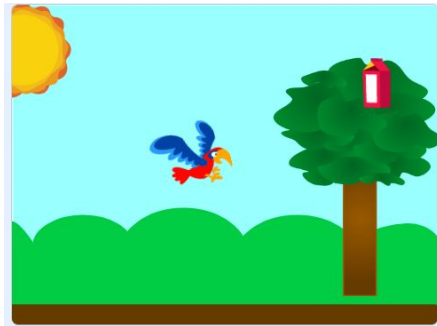
Einsatz mit (angehenden) Lehrkräften:

- Affektive Aspekte werden wenig beachtet
- deutliche Varianz in den Aspekten die als Grundlage für die Diagnose herangezogen wurden, als auch Interventionen
- Interventionen mit teilweise wenig Bezug zur vorher gestellten Diagnose

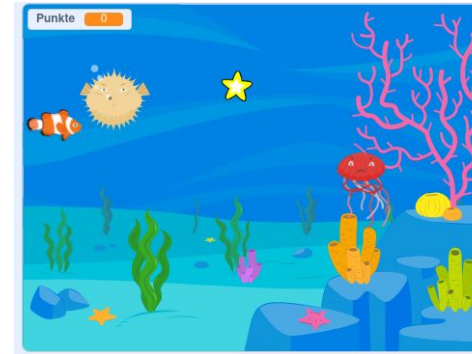


Analyse von Debuggingprozessen

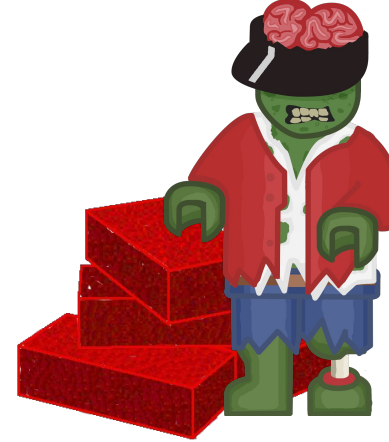
Mangel an Erkenntnissen zu den Debuggingprozessen und debuggingbezogenen Lernprozessen von Schüler:innen



```
when green flag clicked
  think Am Baum wächst Milch? for 1 seconds
  turn 15 degrees
  forever loop
    if touching Milch? then
      say Lecker! for 2 seconds
      stop all
    else
      forever loop
        next costume
        move 10 steps
        wait 0.1 seconds
```



```
when green flag clicked
  set Punkte to 0
  if touching Fisch? then
    go to random position
    change Punkte by 1
```





Fazit

- Debuggingfähigkeiten unterscheiden sich von Programmierfähigkeiten und entwickeln sich kaum von selbst
- Aber: Debugging kann und sollte (explizit) vermittelt werden!
- Wichtig dabei:
 - Fordern und Fördern von Selbstständigkeit
 - Variabel bei Bedarf im Unterricht einsetzbar sein (kein Wissen auf Vorrat)
 - Fokus auf das Formulieren von Hypothesen

Konkrete Ansätze für den Unterricht: Siehe Workshop ;)



Ich bin tatsächlich der Meinung, dass es ja nicht ganz fair ist, dem Schüler immer vorzuwerfen, er findet die Fehler nicht, wenn ich ihm doch nicht mal gezeigt habe, was ich normalerweise nutze. Das empfinde ich jetzt als unfair. Deshalb ist eben das Minimum, was ich ihm zeigen muss, das, was ich üblicherweise selber benutzte.

Unterrichtsmaterial unter [Computingeducation.de](https://www.computingeducation.de)



ComputingEducation
Computer science explained!

HOME ÜBER UNS PUBLIKATIONEN THEMEN

OBTS NEWS

21. JAN 2028

QUANTENINFORMATIK

Quanteninformatik

Quanteninformatik gilt als eine der vielversprechendsten neuen technologischen Entwicklungen - Forscherinnen und Forscher...

1 MIN. LESEZEIT

Von DATEN und BÄUMEN

KI
IT2School - Module zu künstlicher Intelligenz

Künstliche Intelligenz In Kooperation mit der Wissensfabrik haben wir im Rahmen des IT2School-Projekts vier Unterrichtsmodule zum Thema...

Agile Schule

AGILE METHODEN
Agile Schule

Agile Projektarbeit macht Schule! Die aus der Softwareentwicklung bekannte Vorgehensweise lässt sich gewinnbringend auf den...

Schlag den Roboter – Maschinelles Lernen erfahren

KI
Schlag den Roboter – Maschinelles Lernen erfahren

Schlag den Computer im Mini-Schach (online). Hinter den jüngsten Entwicklungen im Bereich künstliche Intelligenz stecken vor allem...



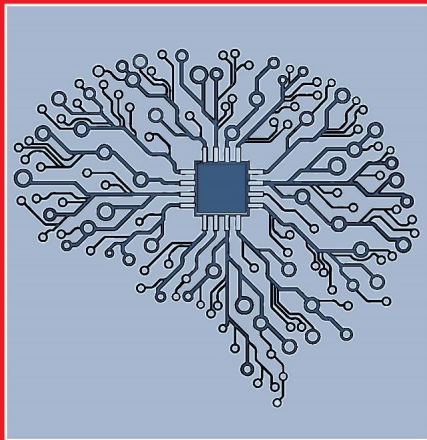


Vielen Dank für die Aufmerksamkeit

<http://www.log-in-verlag.de/>

LOG IN

Informatische Bildung und Computer in der Schule



Künstliche Intelligenz
und Unterricht.
Menschheitsträume.
So lernen Maschinen!
Nachdenken über KI.

Maschinelles Lernen.
KI im Unterricht.
IT-Ethik in der Schule.
Robby lernt –
aber nicht alles!

Nr. 193/194

2020

LOG IN Verlag

Informatische Bildung in Schulen

Zeitgemäße, gute informatische Bildung ist entscheidend, um jeden und jede zum Verstehen und Gestalten der digitalen Welt im Sinne mündiger Bürgerinnen und Bürger zu befähigen.

Im Jahr 2022 ist die letzte Ausgabe der LOG IN, der einzigen deutschsprachigen Zeitschrift für Lehrkräfte im Bereich der informatischen Bildung, erschienen. In der Zeitschrift *Informatische Bildung in Schulen (IBIS)* wird diese wichtige Arbeit unter dem Dach des Fachausschusses Informatische Bildung an Schulen und der Fachgruppe Didaktik der Informatik der Gesellschaft für Informatik e. V. fortgeführt.

ARTIKEL SENDEN

