

Fellowship für Innovationen in der Digitalen Hochschullehre

# Feedback und Lehrpfade in Jupyter Notebooks

Antrag von

Prof. Dr. Christina B. Claß

Ernst-Abbe-Hochschule Jena

Datum: 3. Mai 2025

## **Zusammenfassung**

An der EAH Jena werden Jupyter Notebooks, die in verschiedene Elemente der Sprache Python einführen, über einen Server zur Verfügung gestellt. Diese Notebooks werden im Selbststudium als Vorbereitung auf das Programmierpraktikum bearbeitet. Sie enthalten zahlreiche Beispiele und viele kleine Aufgaben, mit denen die Studierenden Schritt für Schritt Kompetenzen aufbauen. Zur Zeit gibt es noch kein automatisches formatives Feedback zu Lösungen dieser Aufgaben, auch werden Studierende nicht darauf hingewiesen, wenn sie Aufgaben überspringen.

Im Rahmen dieser Lehrinnovation wird ein Konzept für Feedback unter Nutzung verschiedener existierender Ansätze in Jupyter Notebooks entwickelt und exemplarisch umgesetzt. Studierende werden zu verschiedenen Zeitpunkten eingebunden, um sicherzustellen, dass das Feedback hilfreich ist. Die entwickelten Best Practice Beispiele sollen als Ergebnis der Lehrinnovation veröffentlicht werden.

Basierend auf den Projektergebnissen wird nach Projektende weiteres Feedback in existierenden Notebooks implementiert. Die Ergebnisse werden auch in die Gestaltung eines neuen Pflichtmoduls einfließen. In diesem Modul ist Feedback in Notebooks besonders wichtig, da sehr wenig Präsenzübungen eingeplant sind.

# 1 Einleitung und Lehrinnovation

Im Rahmen der Online Lehre während der Corona Pandemie habe ich für meine Informatik-Module, die Programmierkenntnisse in Python vermitteln, Jupyter Notebooks erstellt. Diese werden als Webpage dargestellt, beinhalten erläuternden Text und haben gleichzeitig Zellen mit interaktivem Python Code, der durch Studierende ausgeführt und geändert werden kann. Auch können hier Aufgaben gestellt werden, welche direkt im Notebook gelöst werden.

Zu den in den Notebooks vorhandenen Aufgaben erhalten die Studierenden aktuell kein **automatisches formatives Feedback**. Unter formativem Feedback wird im Folgenden Feedback verstanden, das Studierende während des Lernprozesses unterstützt. Eine Bewertung, im Sinne von Punkten, Prozentsätzen oder Noten schließt das Feedback im Sinne dieses Antrags nicht ein.

Im Falle von Syntaxfehlern erhalten Studierende die regulären Fehlermeldungen des Python Interpreters. Weiteres Feedback erhalten sie nicht. Dies betrifft unter anderem die folgenden Punkte:

**Feedback 1:** wurde die Fragestellung richtig verstanden und das richtige Problem bearbeitet?

**Feedback 2:** wurden die geforderten Sprachelemente von Python verwendet?

**Feedback 3:** ist die Lösung korrekt?

**Feedback 4:** wurde das Problem vollständig gelöst?

**Feedback 5:** wurde die Aufgabe bearbeitet?

Dieses formative Feedback wäre für die Studierenden wichtig. Feedback 2 bezieht sich z. B. auf die Sprachelemente (z.B. die `while`-Schleife, nicht die `for`-Schleife), die in der entsprechenden Aufgabe geübt werden sollen. Punkt 4 bezieht sich auch darauf, ob alle Fälle, einschl. von Spezialfällen, behandelt werden. Erfahrungsgemäß werden bei der Bearbeitung von Notebooks vereinzelt Aufgaben übersehen (Feedback 5). Diese Feedbacks wären für den Lernerfolg der Studierenden sehr hilfreich.

Automatisiertes Feedback für Programme zu geben ist komplex und nur eingeschränkt möglich. Es gibt verschiedene existierende Ansätze, die im Zusammenhang mit Feedback genannt werden. Es ist jedoch unklar, bei welchen Aufgaben und Fragestellungen diese genutzt werden können, um ein formatives Feedback wie oben genannt zu geben.

**Ziel** der beantragten Lehrinnovation ist, bestehende Ansätze für Feedback zu evaluieren und eine Abbildung zwischen Aufgaben, Anforderungen an Feedback und Ansätzen zu erstellen. Mit ausgewählten Ansätzen soll für geeignete konkrete Aufgaben in Notebooks Feedback implementiert und beim Einsatz in der Lehre konkret evaluiert werden. Damit möchte ich meine Lehre verbessern und Studierende besser dabei unterstützen, relevante Kompetenzen zu erwerben.

Fragen nach automatisiertem, qualitativ hochwertigem, formativem Feedback betreffen viele digitale Lernangebote und sind nicht auf Programmieraufgaben beschränkt. Daher werden Ergebnisse der Lehrinnovation, wenn sie im konkreten auch spezifisch sind, auch allgemein hilfreiche Erkenntnisse beinhalten.

Dieser Antrag ist wie folgt aufgebaut. Abschnitt 2 behandelt die Bedeutung von Informatik-Kompetenzen und den Programmierunterricht, eine Grundlage meines Engagements in der Lehre. In Abschnitt 3 erläutere ich das Konzept der Notebooks etwas genauer und stelle den aktuellen Einsatz und spezifische Herausforderungen vor. In Abschnitt 4 werden das Projekt und die geplanten Aktivitäten genauer vorgestellt. Dieser Abschnitt erläutert auch, wie die Ergebnisse genutzt werden. Abschnitt 5 erläutert meine Motivation für die Lehrinnovation und die Fellowship.

## 2 Die Bedeutung von Informatik-Kompetenzen

Ich arbeite seit Wintersemester 1999/2000 als Dozentin und später Professorin an verschiedenen Hochschulen in der Schweiz, Jordanien und seit 2017 an der EAH Jena. Bereits zu Beginn meiner Lehrtätigkeit habe ich Programmierung, damals in C, später Java, C# und Python, unterrichtet.

Programmieren zu lernen ist nicht trivial und umfasst mehr als das reine Kennenlernen einer Programmiersprache: Studierende müssen die Fragestellung (Anforderungen) präzise erfassen, notwendige Daten und ein Lösungsverfahren (den Algorithmus) definieren, diese in Strukturen des Programmierparadigmas (z.B. strukturierte, funktionale oder objektorientierte Programmierung) übertragen und dann in einer Programmiersprache ausdrücken. Die dabei notwendigen Kompetenzen müssen erarbeitet werden – Programmieren lernt man nur durch Praxis.

Die mit obigen Schritten zusammenhängenden Kompetenzen sind vielfältig und in unserer heutigen Welt allgemein relevant. So plädieren 2021 Andera Gadeib und Stephan Noller: "Die drei wesentlichen Kulturtechniken Lesen, Schreiben und Rechnen sollten daher um eine informatische Perspektive ergänzt werden." [GN21] 2006 wurde das Konzept des **Computational Thinking** / Informatischen Denkens als "eine grundlegende Fähigkeit für alle, nicht nur für Informatiker" definiert. [Win06] (Übersetzung von H. Hellwagner, G. Kappel und R. Grosu, <https://www.cs.columbia.edu/~wing/ct-german.pdf>) Die Bedeutung digitaler Kompetenzen wird auch von anderer Seite festgestellt: z.B. als "Digital Intelligence" definiert vom DQ Institute in Verbindung mit WEF, OECD, IEEE SA [Par19] sowie im Zusammenhang mit dem "Europäischen Referenzrahmen für digitale Kompetenzen der Bürgerinnen und Bürger" [CVP18].

2015 erlangte das Konzept von **Data Literacy**, der "Fähigkeit, Daten auf kritische Art und Weise zu sammeln, zu managen, zu bewerten und anzuwenden" ([RRS<sup>+</sup>15] nach [HBK18]) Aufmerksamkeit.

2020 stellte der Wissenschaftsrat fest, dass "die Informatik in eine Rolle an den Hochschulen hineinwächst, die derjenigen der Mathematik ähnlich ist. Informatisches Denken und informatische Konzepte sind für immer mehr Disziplinen von Relevanz und müssen aus der Informatik heraus gelehrt werden." ([Wis20])

Trotz dieser drängenden Herausforderungen ändert sich strukturell nur langsam etwas. Die Gesellschaft für Informatik hat 2008 die ersten Bildungsstandards Informatik für die Sekundarstufe 1 veröffentlicht, in diesem Jahr wurden sie überarbeitet [GI25]. Standards für die Sekundarstufe 2 und die Primarstufe folgten. Doch bis heute gibt

es kein Deutschland-weites Informatik-Pflichtfach.<sup>1</sup> An den Hochschulen können wir immer noch keine Informatik-Kenntnisse voraussetzen. Die Stundenkontingente für entsprechende Module können nur vereinzelt erhöht werden, neue Angebote wie z.B. in Data Literacy können nur vereinzelt eingeführt werden.

### **Konkrete Lehrer Erfahrungen und persönliche Motivation:**

Als wir in der Corona-Pandemie wieder vor Ort aber unter Beachtung von Abstandsregeln unterrichten konnten, wurde ein Tool angeschafft, das es Dozierenden ermöglicht, die Monitore der Studierenden einzusehen. Ich benutze dieses Tool weiterhin und mache den Nutzen selbstverständlich transparent. Ich schätze dieses Programm sehr: Wenn ich sehe, dass jemand mit einer Aufgabenstellung ringt oder einen Fehler gemacht hat, kann ich zuerst mal abwarten und beobachten, ob Studierende selber eine Lösung finden. Auch habe ich die Möglichkeit, Probleme und Fehler in Ruhe zu analysieren, um dann gezielt zu unterstützen.

Die folgenden Darstellungen basieren auf persönlichen Beobachtungen, sie sind nicht formal erhoben worden.

Die heutigen Studierenden zählen mehrheitlich zur Generation "Z", den "Digital Natives". Dies bezieht sich auf den selbstverständlichen Umgang mit der Technologie. Informatisches Denken entsteht nicht automatisch durch diesen Umgang, sondern muss erlernt werden. Basierend auf meiner eigenen Lehrer Erfahrung erleichtert der selbstverständliche Umgang vieles: Hemmschwellen sind nicht mehr vorhanden und der fast spielerische Umgang ermöglicht Lernen durch Ausprobieren und Experimentieren. Auf der anderen Seite scheint man oft sehr schnell einfach weiterzugehen ("swipen"), die nächste Aufgabe kurz anzusehen, den nächsten Code auszuprobieren ohne innezuhalten, um zu reflektieren, über Code, Ergebnisse oder auch Fehlermeldungen nachzudenken. So fehlt häufig eine notwendige "Reflexionsschleife", die das Lernen erst ermöglicht.

Ich passe meine Lehre beständig an, um sie besser an die neuen Lerngewohnheiten und die veränderte Interaktion mit Computern und Programmierwerkzeugen anzupassen, um die Grundlagen des informatischen und kritischen Denkens vermitteln zu können. Für mich persönlich geht es dabei nicht nur um die Vermittlung einer Programmiersprache und relevanter fachlicher Grundlagen für das weitere Studium.

Schüller und Busch begründen die Bedeutung von Data Literacy mit dem Erwerb von Fähigkeiten, "die ein mündiger Bürger [...] benötigt, um sich in einem Überangebot von Daten und Informationen zurechtzufinden und fundierte Entscheidungen zu treffen". ([SB19]) Programmiersprachen wie Python, im Netz vorhandene Codebeispiele und zunehmend auch generative KI machen es leicht, Daten auf vielfältige Art zu visualisieren und eine Abbildung so lange zu verändern, bis sie die gewünschte Aussage möglichst gut unterstützt. Meiner Meinung müssen wir Studierenden an Beispielen ermöglichen, hier selber Erfahrungen zu machen. So werden sie darauf sensibilisiert, Abbildungen genau anzusehen und kritisch zu hinterfragen – wir leben in einer Zeit, in der nicht nur viele "Fake News" kursieren, sondern wir zunehmend mit irreführenden Visualisierungen von Daten zu tun haben. Dies ist ein zunehmendes Problem für Wissenschaft sowie den gesellschaftlichen Diskurs.

---

<sup>1</sup> <https://informatik-monitor.de/2024-25>

Studierende fangen zunehmend damit an, auch einfache Aufgaben in Praktika offen z.B. mit ChatGPT<sup>2</sup> zu lösen. Ich möchte das nicht verbieten, sondern versuchen, dies auch in die Lehre einzubinden. Auf der Buchrückseite eines aktuellen Buchs steht z.B. "AI won't replace programmers - but programmers with AI are going to replace programmers that don't use AI." [KOS25] Es wird normal werden, sofern es das nicht schon ist, Codesegmente mit Hilfe von KI zu entwickeln. Der von generativen Modellen erstellte Code kann nicht besser sein, als die Prompts, mit denen man die Anforderungen an den Code spezifiziert. Und hierzu muss man sie kennen und präzise formulieren können, ebenso wie konkrete Hinweise im Blick auf die Lösung. Informatisches Denken und grundlegende Kenntnisse im Programmieren sind hierfür unerlässlich.

Die Bedeutung von Computational Thinking, Data Literacy, Grundkenntnissen der Programmierung und kritischem Denken nimmt daher zu. Ich erhoffe mir, durch bessere Möglichkeiten des Feedbacks diese Kompetenzen meiner Studierenden im Rahmen meiner Lehre zu stärken.

### 3 Jupyter Notebooks

#### Einsatz an der EAH Jena:

Im Zuge der Online Lehre während der Corona Pandemie habe ich zu allen Programmierinhalten Jupyter Notebooks entwickelt. Diese werden in einem Browser ausgeführt und enthalten reguläre Textelemente. Gleichzeitig gibt es Zellen mit interaktivem Python Code, der durch die Studierenden ausgeführt und geändert werden kann. In diesen Notebooks erläutere ich Konzepte und Sprachelemente. Auch erhalten Studierende kleine Aufgaben, in denen sie die erläuterten Sprachelemente anwenden (siehe Abb. 1).

```
: # wir definieren die Variable ergebnis und initialisieren sie
ergebnis=0
# wir lesen die erste Zahl ein
zahl=int(input('zahl: '))
# die while Schleife: solange die eingelesene Zahl ungleich 0 ist
while zahl!=0:
    # wir addieren die eingelesene Zahl zum Ergebnis
    ergebnis=ergebnis+zahl
    # erneutes Einlesen der Zahl
    zahl=int(input('zahl: '))
# und nun geben wir das Endergebnis aus:
print('Das Ergebnis ist',ergebnis)
print('Ende des Codes')
```

Testen Sie obigen Code.

Auch hier gilt natürlich, dass die Schleife nicht ausgeführt wird, wenn Sie bereits beim ersten Mal eine 0 eingeben.

**Aufgabe:** Schreiben Sie Python Code, der positive Zahlen einliest und diese ausgibt. Sobald der Benutzer eine Zahl  $\leq 0$  einliest, wird die Eingabe abgebrochen.

```
: # Ihr Code
```

**Aufgabe:** Schreiben Sie Python Code, der positive Zahlen einliest, deren Summe berechnet und diese ausgibt. Sobald der Benutzer eine Zahl  $\leq 0$  einliest, wird die Eingabe abgebrochen.

```
: # Ihr Code
```

**Frage:** Was müssen Sie in obigem Code ändern, um nicht nur das Endergebnis, sondern auch alle Zwischenergebnisse auszugeben? (Hinweis: Denken Sie an die Einrückungstiefe.)

#### Abbildung 1: Ausschnitt eines Notebooks zur `while`-Schleife

Nachdem die Python-Sprachelemente in einem Notebook eingeführt wurden, werden diese in Praktikumsaufgaben in einem normalen Entwicklungswerkzeug genutzt. Aktuell

<sup>2</sup>Ich verwende hier ChatGPT als Vertreter der Klasse der vielen Modelle der generativen KI, da dieses sehr bekannt ist und teilweise einen Modellcharakter hat, wie Google für Suchmaschinen.

nutze ich die Notebooks in drei Informatik-Pflichtmodulen in verschiedenen Studiengängen. Zur Übersicht befindet sich am Ende dieses Dokuments ein vollständiges Inhaltsverzeichnis meiner aktuell verwendeten Notebooks.

Im Jahr 2020 wurde unter gemeinsamer Leitung mit meinem Kollegen Prof. Dr. C. Schneider ein Python Tutorial zum Selbststudium zur Wiederholung mit Jupyter Notebooks entwickelt. Das Tutorial stellt am Ende jedes Kapitels Selbsttests mit Multiple Choice Fragen zur Verfügung. Ich habe das Tutorial 2022 komplett überarbeitet und auf Englisch übersetzt. Aktuell wird es in deutscher und englischer Version Studierenden verschiedener Module zur Verfügung gestellt.

In den ersten Jahren mussten die Studierenden die Notebooks lokal auf ihren Rechnern laufen lassen und entsprechende Software installieren. Im Frühjahr 2023 haben wir auf einem Server des Fachbereichs einen kleinen Jupyter Hub<sup>3</sup> installiert, der für ca. 100 User ausgelegt ist. Seit Herbst 2023 kann auf den Hub auch von außerhalb des EAH Netzes zugegriffen werden. Aktuell verwalte ich drei Instanzen des Hubs, um ca. 300 Studierenden die Möglichkeit zu geben, Notebooks nutzen.

### **Erfahrungen:**

Jupyter Notebooks wurden für interaktive Projekte im Bereich des Data Science entwickelt. Sie eignen sich zwar gut, um Programmelemente zu erläutern und auszuprobieren, aber sie sind per se nicht als Tool zum Programmieren-Lernen gedacht. Der Schritt in eine "normale" Programmierumgebung bedeutet eine Umstellung. Ich versuche daher die ersten Aufgaben, die Studierende ohne Notebooks lösen, immer sehr nah die Notebooks anzulehnen, um diese Umstellung zu erleichtern. Dies funktioniert aber nur dann, wenn die Studierenden die Aufgaben in den Notebooks bearbeitet und hierbei die entsprechenden Sprachelemente genutzt haben.

Im Lehralltag beobachte ich teilweise anderes: Studierende klicken sehr schnell und unaufmerksam durch die Notebooks hindurch, dies hat auch mit dem bereits erwähnten Lernverhalten zu tun. Teilweise bearbeiten sie mehrere Aufgaben nicht, die auf Aufgaben vorbereiten, die sie dann im Praktikum machen sollen. Wenn ich mit Studierenden zusammensitze, um sie zu unterstützen, bitte ich sie häufiger, ein Notebook zu öffnen, um mit ihnen als Hinweise bereits gelöste Aufgaben anzusehen. Teilweise stelle ich dabei fest, dass diese gar nicht gelöst wurden. Hier bitte ich die Studierenden, zuerst die Aufgabe(n) im Notebook zu bearbeiten. Falls Aufgaben unvollständig oder falsch gelöst wurden, diskutieren wir gemeinsam über die Lösung und korrigieren diese, bevor die Studierenden dann mit der Bearbeitung der Praktikumsaufgaben selbstständig fortfahren können.

Feedback, wie in Abschnitt 1 genannt, wäre für die Unterstützung der Studierenden sehr wichtig. Aus diesem Grund möchte ich im Rahmen der geplanten Lehrinnovation Möglichkeiten des Feedbacks eruieren, beispielhaft umsetzen und die Umsetzungen evaluieren.

---

<sup>3</sup><https://tljh.jupyter.org/en/latest/>

## 4 Die Lehrinnovation: Umsetzung

### Eine technische Nebenbemerkung zu existierenden Ansätzen:

Es existieren verschiedene Module, die im Zusammenhang mit Notebooks und Feedback genannt werden (z.B. jupytercheck, ipwidgets, nbgrader, testbook). jupytercheck<sup>4</sup> wurde für Aufgaben der Linearen Algebra, also einem ganz anderen Anwendungsgebiet, entwickelt. ipwidgets<sup>5</sup> stellen verschiedene Kontrollelemente zur Verfügung und eignen sich gut für Multiple Choice Fragen. testbook<sup>6</sup> und nbgrader<sup>7</sup> basieren auf beobachtbaren Antworten und überprüfen z.B. nicht, welche Code-Elemente verwendet werden. Die Möglichkeiten u.a. dieser Ansätze für Feedback sollen im Rahmen dieser Lehrinnovation untersucht werden.

Für die Umsetzung der Lehrinnovation sind die folgenden **Ressourcen** geplant (siehe Finanzplan):

- Vergabe von Lehraufträgen zur teilweisen Freistellung der Antragstellerin (Reduktion des Lehrdeputats) zur Bearbeitung des Projekts
- Beschäftigung von wissenschaftlichen Assistent:innen während der Lehrinnovation. Diese unterstützen bei der Konzeptentwicklung, Implementation und Evaluation. Sie spielen auch eine wichtige Rolle bei der gemeinsamen Analyse von Aufgaben und den Anforderungen an Feedback. Als gleichzeitig Studierende können sie hier sehr wichtigen Input für ein zielgruppengerechtes Feedback geben. Die Mitarbeit am Projekt gibt ihnen gleichzeitig die Möglichkeit zum Kompetenzerwerb im Bereich Jupyter Notebooks, Programmierung und der Gestaltung von Evaluationen.

In Abb. 2 auf S. 11 ist der Arbeitsplan skizziert. Wissenschaftliche Assistent:innen sind in allen Arbeitspaketen beteiligt.

### AP 1: Konzept

Anforderungen an Feedback werden basierend auf Aufgaben in bestehenden Notebooks und einer Umfrage bei aktuellen und ehemaligen Studierenden formuliert. Bestehende Ansätze (siehe technische Nebenbemerkung) werden anhand kleinerer Beispiele evaluiert. Hierauf basierend wird ein Konzept für die Implementation von Feedback in ausgewählten Notebooks erstellt.

### AP 2: Entwicklung Teil 1

Ein erster Teil des in AP 1 spezifizierten Konzepts wird umgesetzt. Ausgewählt werden verschiedene Arten von Feedbacks die zu Aufgaben in Notebooks passen, die zu Beginn des Sommersemesters zum Einsatz kommen. Anfang Mai 2026 ist die erste Version implementiert (Meilenstein 1). Die Notebooks kommen dann im Unterricht zum Einsatz und werden von den Studierenden evaluiert. Da das Feedback nur in einigen Notebooks umgesetzt wird, werden in der Evaluation Notebooks mit und ohne Feedback verglichen. Neben generellen Fragen haben die Studierenden auch die Gelegenheit, Rückmeldung zu spezifischen

<sup>4</sup><https://github.com/colbrydi/jupytercheck>

<sup>5</sup><https://ipywidgets.readthedocs.io/en/latest/>

<sup>6</sup><https://github.com/ninteract/testbook>

<sup>7</sup><https://nbgrader.readthedocs.io/en/stable/>

Feedbacks zu geben. Basierend auf den Hinweisen von Studierenden wird die Umsetzung der Feedbacks überarbeitet sowie einzelne Formulierungen verbessert.

### **AP 3: Entwicklung Teil 2**

Während die in Teil 1 entwickelten Notebooks zum Einsatz kommen und evaluiert werden, wird bereits mit der Entwicklung des zweiten Teils begonnen. Hierbei wird nach Möglichkeit auch automatisches Feedback implementiert, das so nicht direkt durch die genannten Ansätze unterstützt wird. Auch werden Möglichkeiten evaluiert, den Lehrpfad deutlicher zu machen, so dass keine Aufgaben übersehen werden. Die so erweiterten Notebooks stehen dann Ende September 2026 (Meilenstein 2) zur Verfügung, so dass sie zu Beginn des Wintersemesters 2026/27 genutzt werden können. Auch hier ist eine Evaluation durch Studierende sowie die Überarbeitung zur Verbesserung von unklarem Feedback vorgesehen.

### **AP 4: Abschluss**

Am Ende des Projekts werden die Ergebnisse zusammengefasst. Hierzu gehört eine Zusammenstellung der betrachteten Ansätze einschl. einer Bewertung der Eignung für automatische formative Feedbacks. Ein oder mehrere Notebooks mit Best Practice Beispielen für Feedback werden erstellt und die Implementation dokumentiert. Die Notebooks und die Dokumentation werden als Ergebnis mit einer offenen Lizenz zur Verfügung gestellt.

### **Verwendung der Lehrinnovation/Projektergebnisse:**

Die Ergebnisse der Lehrinnovation werden an der EAH Jena nachhaltig genutzt: Basierend auf den entwickelten Empfehlungen wird formatives Feedback in den existierenden Notebooks ergänzt. Diese Notebooks werden in Informatik Grundlagenmodulen verschiedener Studiengänge genutzt. Es handelt sich um Pflichtmodule. Die Notebooks sollen des Weiteren in einem neuen Modul, das ab Sommersemester 2027 angeboten wird, genutzt werden. Dieses Modul zeichnet sich durch einen besonders hohen Anteil an Selbststudium aus. Das Selbststudium mit Notebooks und die in dieser Lehrinnovation erarbeiteten Möglichkeiten des Feedbacks spielen hierbei eine wichtige Rolle.

### **Risiken:**

Wie in anderen Projekten auch, ist auch die Umsetzung dieser Lehrinnovation davon abhängig, dass geeignetes Lehrpersonal und wissenschaftliche Assistent:innen gefunden werden. Des Weiteren wird im Rahmen des Projekts auch eruiert, inwiefern es möglich ist, formatives Feedback wie in Abschnitt 1 beschrieben mit bestehenden Ansätzen und Modulen zu erstellen. Evtl. wird das Ergebnis sein, dass die Möglichkeiten eingeschränkt sind und die Ziele nur teilweise erreicht werden können. In diesem Fall würden wir andere Wege evaluieren, um Studierende, z.B. durch detailliertere Hinweise, zu unterstützen. Hierbei müsste dann evaluiert werden, wie diese Hinweise aussehen dürfen ohne den Erwerb relevanter Kompetenzen zu mindern. Erste Hinweise auf notwendige Anpassungen der Zielstellung würden sich schon früh im AP 1 ergeben, so dass darauf reagiert werden kann.

## 5 Motivation

Eine wichtige Motivation, mich um diese Fellowship zu bewerben, liegt im Wunsch begründet, meine Lehre und den Einsatz der Notebooks durch Möglichkeiten des automatischen formativen Feedbacks zu verbessern und meine Studierenden besser dabei zu unterstützen, relevante Kompetenzen zu erwerben.

Die Fellowship würde mir durch eine Lehrentlastung Zeit und Raum geben, mich mit dem Thema von formativem Feedback beim Programmieren-Lernen auseinanderzusetzen. Die geplanten drei Evaluationen / Umfragen bei den Studierenden geben mir die Gelegenheit, besser zu verstehen, wie Studierende mit den Notebooks und den kleinschrittigen Aufgaben umgehen und diese zum Lernen nutzen. Ich erhoffe mir Erkenntnisse über die Rolle von formativem Feedback in diesen konkreten Lernprozessen.

Die geplante Zusammenarbeit mit zwei wissenschaftlichen Assistent:innen wäre ein großer Gewinn für mich: Im Rahmen dieser Lehrinnovation werden wir uns immer wieder über Lernen, Aufgaben und Feedback austauschen und gemeinsam Feedback formulieren. So hätte ich Gelegenheit, mehr über die (Lern)Welt der aktuellen Studierenden zu erfahren und meine Lehre besser auf sie einzustellen.

Persönlich interessiere ich mich sehr für die Lehre und setze mich regelmäßig mit unterschiedlichen Themen der Didaktik auseinander. Leider lässt einem der Alltag, insbesondere an einer HAW mit einem Deputat von 18 SWS, oft wenig Zeit, sich intensiver mit Hintergründen auseinanderzusetzen. Und im Alltag kommt leider auch oft der gegenseitige Austausch, das Zuhören, Erzählen, voneinander Lernen zu kurz. Natürlich kann ich nicht wissen, welche Themen und Lehrinnovationen andere Fellows bearbeiten möchten. Ich würde mich aber sehr freuen, mich mit anderen über ihre Projekte auszutauschen und voneinander zu lernen, gemeinsam Ideen zu entwickeln und vielleicht auch gemeinsam Lösungsansätze für Fragen zu finden, die uns bewegen.

Wie bereits zuvor erwähnt, ist mir wichtig, Studierenden nicht nur fachliche sondern auch generelle Kompetenzen zu vermitteln, die sie in unterschiedlichen Kontexten benötigen. Grundlegende Kompetenzen des Computational Thinking, der Data Literacy, des kritischen Denkens und zunehmend auch des verantwortungsbewussten Umgangs mit KI-Tools sind für mich keine leeren Worte. Ich versuche diese zumindest mit kleinen Aufgaben und Impulsen zu stärken, auch wenn ich nicht viel Zeit zur Verfügung habe. Ich würde mich daher insbesondere freuen, wenn sich die Gelegenheit ergibt, über Kompetenzen, die im Blick auf generative KI wie ChatGPT etc. notwendig sind, zu sprechen: Welche Kompetenzen brauchen unsere Studierenden in Zeiten der KI? Wie können wir den Erwerb solcher Kompetenzen in der digitalen Lehre unterstützen? Wie können wir unsere Studierenden bestmöglich auf die Welt von morgen vorbereiten?

Der Austausch mit den anderen Fellows und die Weiterentwicklung meiner Lehrexpertise sind daher für mich eine zentrale Motivation, mich um die Fellowship zu bewerben.

### Antrag:

Für die hier vorgestellte Lehrinnovation **Feedback und Lehrpfade in Jupyter Notebooks** im Rahmen einer Fellowship für Innovationen in der Digitalen Hochschullehre beantrage ich die im Budget in der Anlage aufgeführten Mittel.

## Literatur

- [CVP18] S. Carretero Gomez, R. Vuorikari, and Y. Punie. DigComp 2.1 : the digital competence framework for citizens with eight proficiency levels and examples of use. Publications Office of the European Union, 2018.
- [GI25] GI. Bildungsstandards Informatik für die Sekundarstufe I – Empfehlungen der Gesellschaft für Informatik e. V. (GI). 2025.
- [GN21] Andera Gadeib and Stephan Noller. Wichtige Kompetenzen für eine lebenswerte Zukunft. In INFOS 2021–19. GI-Fachtagung Informatik und Schule, pages 21–24. Gesellschaft für Informatik, Bonn, 2021.
- [HBK18] Jens Heidrich, Pascal Bauer, and Daniel Krupka. Future Skills: Ansätze zur Vermittlung von Data Literacy in der Hochschulbildung. Number 37. Hochschulforum Digitalisierung, August 2018.
- [KOS25] Michael Kofler, Bernd Öggl, and Sebastian Springer. Coding mit KI Das Praxisbuch für die Softwareentwicklung. Rheinwerk Computing, Bonn, 2025.
- [Par19] Yuhyun Park. DQ Global Standards Report 2019: Common Framework for Digital Literacy, Skills and Readiness. DQ Institute, 2019.
- [RRS<sup>+</sup>15] Chantel Ridsdale, James Rothwell, Michael Smit, Hossam Ali-Hassan, Michael Bliemel, Dean Irvine, Daniel Kelley, Stan Matwin, and Bradley Wuetherick. Strategies and best practices for data literacy education: Knowledge synthesis report. 2015.
- [SB19] Katharina Schüller and Paulina Busch. Data Literacy: Ein Systematic Review. Number 46. Hochschulforum Digitalisierung, August 2019.
- [Win06] Jeannette M Wing. Computational thinking. Communications of the ACM, 49(3):33–35, 2006.
- [Wis20] Wissenschaftsrat. Perspektiven der Informatik in Deutschland. [https://www.wissenschaftsrat.de/download/2020/8675-20.pdf?\\_\\_blob=publicationFile&v=9](https://www.wissenschaftsrat.de/download/2020/8675-20.pdf?__blob=publicationFile&v=9), October 2020. [last accessed April 28 2025].

## A Arbeitsplan

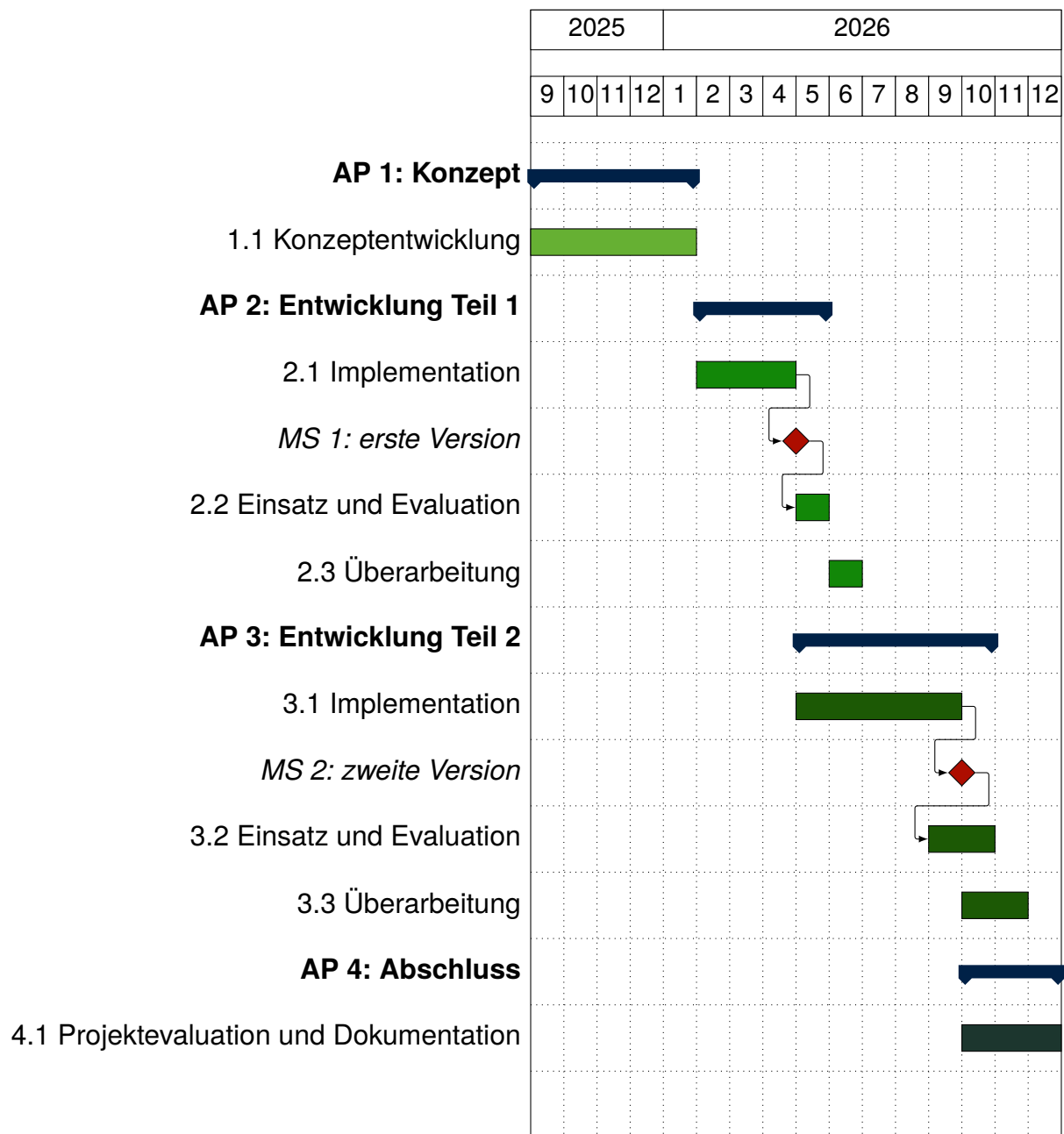


Abbildung 2: Arbeitsplan