

# LuCID – Die Lernumgebung fürs Programmieren basierend auf 4C-ID

## 1 Persönliche Motivation

Ich unterrichte seit 2,5 Jahren an der FH Aachen im Fachbereich Wirtschaftswissenschaften unterschiedliche Module im Bereich der Wirtschaftsinformatik. In dieser Zeit habe ich mehrere neue Module konzipiert und unterrichtet, in denen eine Einführung in die Programmierung gegeben wird. Diese Veranstaltungen richten sich hauptsächlich an Studierende aus Studiengängen (z.B. Betriebswirtschaftslehre oder Industrial Engineering) in welchen Programmieren üblicherweise kein Bestandteil ist. Mein Angebot gründet in der Erkenntnis, dass Programmierkenntnisse eine elementare Fähigkeit darstellen, um die Möglichkeiten der stetig voranschreitenden Digitalisierung einschätzen und nutzen zu können. Gleichzeitig hat die Anzahl der Studierenden, die sich für das Thema Programmierung interessieren, stetig zugenommen. Aktuell lernen und vertiefen jährlich 80 Studierende (Tendenz stark steigend) in den verschiedenen Veranstaltungen die Programmiersprachen Python.

Im Rahmen dieser Veranstaltungen habe ich in den letzten Semestern die Erfahrung gemacht, dass Studierende das Programmieren im Vergleich zu anderen Fächern als schwierig und anstrengend empfinden. (Diese Erfahrung teilen insbesondere auch die Kollegen aus dem Studiengang Informatik. Auch hier ist Programmierung neben der Mathematik das Fach, bei dem die Studierenden die meisten Probleme haben.) Diese Sicht der Studierenden auf das Programmieren führt dazu, dass trotz steigender Tendenz vergleichsweise wenige Studierende aus dem Fachbereich Wirtschaftswissenschaften sich die Mühe machen, Programmieren zu lernen. Zusätzlich brechen viele Studierende, die eine entsprechende Veranstaltung belegen, diese nach wenigen Wochen ab. Auf der anderen Seite geben Studierende, die eine Veranstaltung bis zum Ende absolvieren, ein sehr positives Feedback. Dieses positive Feedback wird auch dadurch unterstrichen, dass mich jedes Semester einige Studierende im Rahmen ihres Abschlussprojekts mit anspruchsvollen Programmen zu selbst definierten Problemen überraschen.

Auf der Suche nach einer Lösung für dieses Problem ist mir aufgefallen, dass einerseits viele Ansätze zum Lernen von Programmieren nicht evidenzbasiert sind. Teilweise widersprechen die eingesetzten Ansätze wie Lernvideos, in denen eine Person zu sehen ist, sogar dem aktuellen Stand der Forschung. Andererseits existiert mit dem Vier-Komponenten Instruktionsdesign Modell (4C-ID-Modell) ein auf jahrzehntelanger Grundlagenforschung basierendes Modell zum Erlernen von komplexen Fähigkeiten wie dem Programmieren. Bisher existiert allerdings keine Lernumgebung für das Programmieren, die dieses Modell vollständig umsetzt. Im Rahmen des Fellowships möchte ich eine solche Lernumgebung und zugehörige Lerninhalte entwickeln und zusammen mit Erziehungswissenschaften evaluieren. Hierbei werde ich von Prof. Dr. Stephan Jacobs, FH Aachen, unterstützt. Mit Prof. Jacobs unterrichte ich gemeinsam in verschiedenen Modulen das Programmieren mit Python. Außerdem wird mich assoc. Prof. Dr. Halszka Jarodzka vom Research Centre for Learning, Teaching and Technology an der Open Universiteit Nederland unterstützen. Frau Jarodzka wird im Projekt insbesondere die Evaluierung mit Hilfe empirischer Studien durchführen. Von diesem interdisziplinären Vorgehen verspreche ich mir:

- einen höheren Lernerfolg bei den Studierenden,
- bessere Lehrveranstaltungen für das Lernen von Programmieren
- Interessante Forschungsergebnisse in Bezug auf das Lehren und Lernen von Programmieren.

Ich möchte möglichst viele Studierende für das Programmieren begeistern!

## 2 Problemstellung

Unsere Erfahrung in der Lehre führt zu folgenden Beobachtungen:

- (1) Programmieraufgaben mit Bezug zur realen Welt führen zur Einübung von Fertigkeiten, die später im Beruf ganz konkret angewendet werden können (im Gegensatz zu abstrakten Übungsaufgaben). Schwierig bei der Erstellung solcher Aufgaben ist aber, diese so zu gestalten, dass die Studierenden nicht überfordert werden.
- (2) Programmieren – egal ob Excel oder Python – erlernt man nur durch viel praktisches Üben.
- (3) Die Unterschiede bzgl. der Vorkenntnisse und Lernfortschritte der Studierenden sind im Vergleich zu anderen Fachgebieten sehr groß.
- (4) Studierende durchlaufen an unterschiedlichen Stellen Frustrationen und geben auf, wenn keine Hilfe kommt.
- (5) Gewisse Teilaspekte des Programmierens benötigen ausgiebige Übung, sodass die Fertigkeiten ohne Nachdenken (d.h. automatisiert) ausgeführt werden können.

Aus Beobachtung (1) ergibt sich, dass Lernaufgaben reale, für die Studierende interessante Problemstellungen beinhalten müssen. Beobachtung (2) führt zu der Schlussfolgerung, dass viele unterschiedliche Aufgaben benötigt werden, die die Studierenden lösen können. Aus Beobachtungen (3) und (5) ergeben sich zwei Anforderungen. Erstens werden Aufgaben auf unterschiedlichen Schwierigkeitsniveaus benötigt. Zweitens gibt es individuelle Lernwege und Lerngeschwindigkeiten. Für jeden dieser Lernwege müssen angepasste Aufgaben verfügbar sein. Die unterschiedlichen individuellen Lernwege führen zu großen Unterschieden im Lernfortschritt. In Kombination mit der Notwendigkeit des *praktischen* Ausprobierens ergibt sich, dass eine klassische Vorlesung als Lehrform ungeeignet ist. Praktika in kleinen Gruppen stellen jedoch auch keine Lösung des Problems dar. Beobachtung (4) führt dazu, dass Studierenden in Frustrationssituationen schnell auf die Problemstellung angepasste Hilfen und Feedback benötigen. Der wöchentliche Rhythmus eines Praktikums passt nicht zu den kurzen Zyklen, in denen Hilfen und Feedback benötigt werden.

Zusammenfassend wird eine digitale Lernumgebung benötigt, die viele unterschiedliche Programmieraufgaben zur Verfügung stellt, den Studierenden schnell und einfach Hilfen auf individuelle Fragen gibt und unterschiedliche Lernwege und einen individuellen Lernfortschritt ermöglicht.

## 3 Welche Ziele sollen erreicht werden?

Ziel dieses Projektes ist, eine Lernumgebung zu konzeptionieren, zu implementieren und zu evaluieren, die die obigen Anforderungen erfüllt. Als theoretische Grundlage dient dabei das Lernmodell *Four Components Instructional Design (4C-ID)* [MCCr 2002]. Zur Erreichung dieses Ziels müssen im Rahmen des Projektes drei Arbeitspakete bearbeitet werden:

- (1) Konzeption und Implementierung einer Lernumgebung auf Basis des 4C-ID-Modells (siehe Abschnitt 3.2)
- (2) Erstellung von Lerninhalten für die Lernumgebung bestehend aus auf einander abgestimmten Übungen und Erklärungen (siehe Abschnitt 3.3)
- (3) Einsatz und Evaluation der Lernumgebung im Rahmen von Lehrveranstaltungen (siehe Abschnitt 3.4).

Im Folgenden werden die theoretischen Grundlagen sowie die einzelnen Arbeitspakete näher beschrieben.

### 3.1 Four Components/Instructional Design(4C-ID)

In der empirischen Bildungsforschung wurden, basierend auf jahrzehntelanger Grundlagenforschung, Modelle entwickelt, die es erlauben Lernumgebungen und Curricula so zu konzipieren, dass diese nachweislich zu signifikant höherem Lernerfolg führen. Eines dieser Modelle ist das Vier-Komponenten Instruktionsdesign Modell (4C-ID-Modell) [MCCr 2002]. Dieses Modell geht von einem holistischen Ansatz aus, welcher Lernen innerhalb der Komplexität der eigentlichen Aufgabenausführung (z.B. im Beruf) ansiedelt. Der Vorteil ist, dass es nicht zu sogenanntem trägen Wissen kommt, welches nach dem Studium nicht auf Aufgaben im echten Berufsleben übertragen werden kann. Der Nachteil eines holistischen Ansatzes des komplexen Lernens ist, dass der Student von der Komplexität schnell überfordert wird. Dazu bietet das 4C-ID-Modell konkrete Ansätze, um das Lernen innerhalb dieser Komplexität dennoch zu ermöglichen, zu unterstützen und zu fördern. Dazu müssen vier Komponenten entwickelt werden:

- (1) **Lernaufgaben:** eine Datenbank von authentischen (im Sinne von: im Berufsleben anwendbaren) Aufgaben, die in Schwierigkeit, Unterstützung und Oberflächenmerkmalen variieren. Diese dienen dem Schemaerwerb und der Abstraktion.
- (2) **Unterstützende Information:** Lehrelemente (z.B. Lehrvideos), die das notwendige Vorwissen verständlich darbieten und dabei an bereits vorhandenes Wissen anknüpfen. Werden den Studierenden vor der Aufgabenbearbeitung dargeboten, stehen ihnen aber auch während der Aufgabenbearbeitung weiterhin zur Verfügung.
- (3) **Prozedurale Information:** Information für Routine Aspekte einer Aufgabe, die direkt bei Bedarf (just-in-time) angeboten werden.
- (4) **Übung von Teilaufgaben:** intensive Übung einzelner Teilaspekte einer Aufgabe, die automatisiert werden sollen.

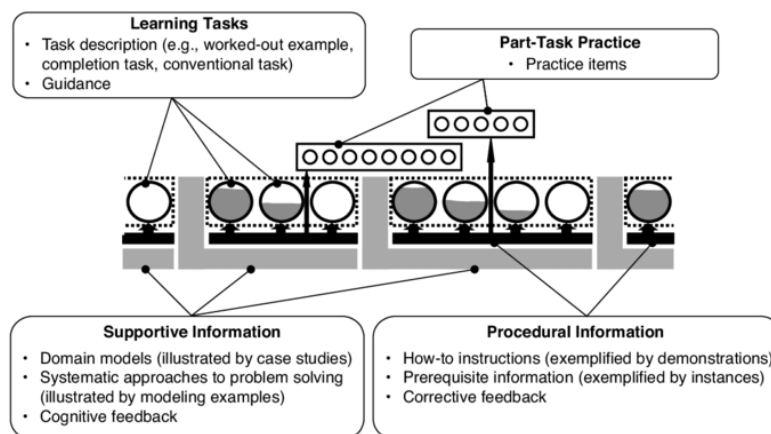


Abbildung 1: 4C-ID-Modell nach [MCCr 2002]

Diese vier Komponenten werden in enger Zusammenarbeit mit Fachdidaktikern (hier: Prof. Drumm, Prof. Jacobs) und Erziehungswissenschaftlern bzw. Kognitionspsychologen (hier: Assoc. Prof. Jarodzka) entwickelt. Zusammenfassend adressiert das 4C-ID-Modell die in der Problemstellung (Abschnitt 2) genannten Beobachtungen.

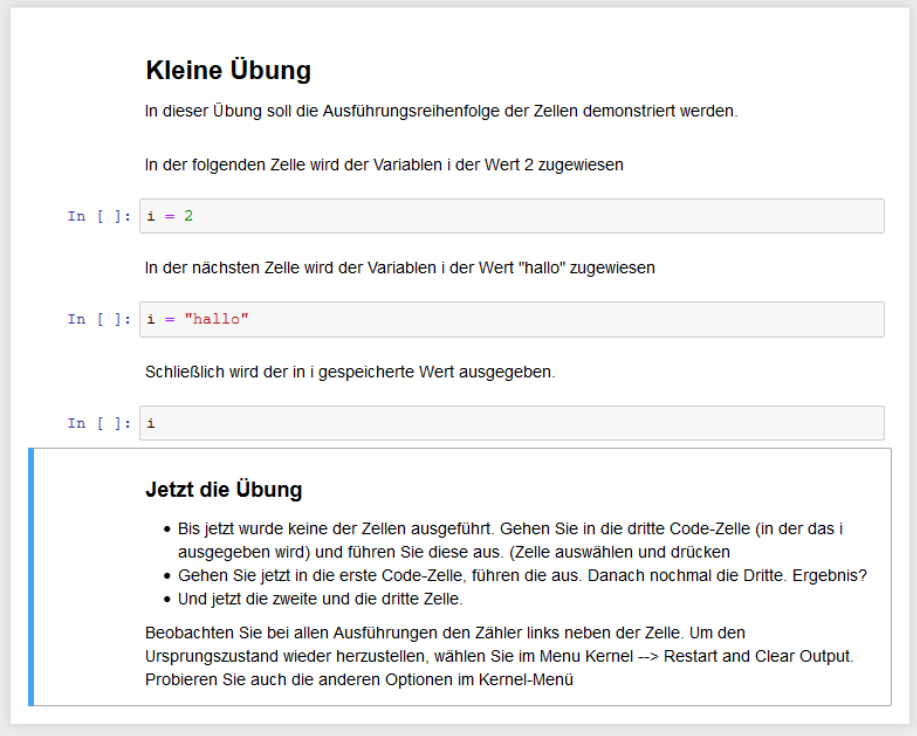
- Holistischer Ansatz entspricht der Beobachtung (1)
- Lernaufgaben entsprechen der Beobachtung (2)
- Unterstützende Informationen entsprechen der Beobachtung (3)
- Prozedurale Informationen entsprechen der Beobachtung (4)
- Übung von Teilaufgaben entsprechen der Beobachtung (5).

## 3.2 Lernumgebung

Derzeit existiert keine Lernumgebung, die die einzelnen Komponenten des 4C-ID-Modell in *einem* System integriert. Im Projekt wird deshalb eine Lernumgebung entwickelt werden, die die technische Basis zur Anwendungen des 4C-ID-Modells in der Lehre bereitstellt. Die im Rahmen des Projektes zu entwickelnde Lernumgebung muss folgende Anforderungen erfüllen:

- (1) Die Lernumgebung soll eine strukturierte Umgebung bieten, in der den Studierenden Lernaufgaben, unterstützende Information, prozedurale Informationen und Übungen von Teilaufgaben je nach Kenntnisstand und Lernfortschritt zur Verfügung gestellt wird.
- (2) Die Lernumgebung soll den Studierenden auf Basis identifizierter Probleme und Fehlern in den Lösungen bearbeiteter Aufgaben, geeignete Aufgabe zur weiterführenden Übung vorschlagen. Verfahren zur Auswahl geeigneter Aufgaben sind aus der 4C-ID Methode bekannt.
- (3) Die Lernumgebung soll den Dozenten die Möglichkeit geben den Lernfortschritt der einzelnen Studierenden zu überprüfen.
- (4) Die Lernumgebung soll den Studierenden ein automatisiertes Feedback zu den gelösten Aufgaben geben. Im einfachsten Fall ist das eine Antwort "Richtig" bzw. "Falsch". Darüber hinaus sollen genauere Hinweise zu erwarteten bzw. häufig vorkommenden Fehlern bzw. Problemen gegeben werden. Im Bereich der automatisierten Tests in der Software-Entwicklung werden solche Verfahren seit langem erfolgreich eingesetzt.

Die Lernumgebung wird dabei nicht von Grund auf neu entwickelt. Stattdessen sollen existierende Funktionen und Software zu einer kompletten Lernumgebung integriert werden. Als Basis der Lernumgebung werden Jupyter Notebooks verwendet. Jupyter Notebooks sind *Notizbücher*, die in jedem Browser dargestellt werden können. Ein wesentliches Element der Jupyter Notebooks ist es, dass Erklärung und ausführbarer Programmcode im gleichen Dokument dargestellt werden (siehe *Abbildung 2*).



**Kleine Übung**

In dieser Übung soll die Ausführungsreihenfolge der Zellen demonstriert werden.

In der folgenden Zelle wird der Variablen `i` der Wert 2 zugewiesen

```
In [ ]: i = 2
```

In der nächsten Zelle wird der Variablen `i` der Wert "hallo" zugewiesen

```
In [ ]: i = "hallo"
```

Schließlich wird der in `i` gespeicherte Wert ausgegeben.

```
In [ ]: i
```

**Jetzt die Übung**

- Bis jetzt wurde keine der Zellen ausgeführt. Gehen Sie in die dritte Code-Zelle (in der das `i` ausgegeben wird) und führen Sie diese aus. (Zelle auswählen und drücken)
- Gehen Sie jetzt in die erste Code-Zelle, führen die aus. Danach nochmal die Dritte. Ergebnis?
- Und jetzt die zweite und die dritte Zelle.

Beobachten Sie bei allen Ausführungen den Zähler links neben der Zelle. Um den Ursprungszustand wieder herzustellen, wählen Sie im Menü Kernel --> Restart and Clear Output. Probieren Sie auch die anderen Optionen im Kernel-Menü

Abbildung 2: Beispiel eines Jupyter Notebooks

Jupyter Notebooks werden seit einigen Jahren erfolgreich in der Programmierung vor allem im Bereich Data Science eingesetzt ([Cour 2019], [Vand 2016], [Petr 2019]) indem Beispielprogramme weitergegeben oder Übungsaufgaben formuliert werden. Der Nutzer\*in eines Jupyter Notebooks hat die Möglichkeit, die Zellen zu verändern und so mit dem Programm zu experimentieren. Zusätzlich bietet die Software JupyterHub die Möglichkeit, Jupyter Notebooks auf einem zentralen Server auszuführen. So kann dann z.B. der Lernfortschritt der Studierenden überprüft werden. In der Lehre bieten Jupyter Notebooks verschiedene Vorteile:

- Wie schon oben dargestellt lassen sich Dokumentation und ausführbare Programme sehr eng in *einem* Dokument verzahnen.
- Es werden aus Sicht des Studierenden keine weiteren Werkzeuge benötigt. Der Studierende kann sich ganz auf die Programmierung konzentrieren.
- Die einzelnen Zellen sind miteinander verbunden. Auf diese Weise kann ein größeres Programm in mehrere Zellen zerlegt werden. Diese einzelnen Zellen können unabhängig voneinander ausgeführt werden. Die Zwischenergebnisse können durch entsprechende Ausgaben einfach dargestellt und kontrolliert werden. Auf diese Weise können komplexe Programme in überschaubare Teile zerlegt werden. Der Studierende lernt, wie ein größeres Problem in kleinere Unteraufgaben zerlegt wird - eine der wesentlichen Herausforderungen bei der Programmierung.
- Für Programme können "Gerüste" (Snippets) vorbereitet werden. Ein Teil eines Programms kann schon vor-implementiert sein. Der Studierende kann sich auf das aktuell besprochene Problem konzentrieren.
- Aus erziehungswissenschaftlicher Sicht, ermöglicht das Jupyter Notebook die Erstellung von Lernaufgaben mit prozeduraler ‚just-in-time‘ Informationen für die Studierenden.

In Bezug zum 4C-ID Modell bieten Jupyter Notebooks folgende Möglichkeiten:

- Sämtliche Lernaufgaben können mit Hilfe von Notebooks dargestellt werden. Die Lernaufgaben werden in der Regel in verschiedenen Versionen zur Verfügung gestellt: ungelöst, zum Teil gelöst, fast vollständig gelöst, vollständig gelöst („level of instructional support“ das sich an das Niveau des Lernenden anpassen lässt). Durch die Integration von Texten und ausführbarem Programmcode, drängen sich Jupyter-Notebooks als Medium für die Aufgaben geradezu auf.
- Unterstützende Informationen werden in der Lernumgebung textuell oder als Screencast (Video) zur Verfügung gestellt. In der Textform lässt sich die unterstützende Information wieder mit Programmbeispielen innerhalb von Notebooks kombinieren. Dies wird z.B. in mehreren "Lehrbüchern" zum Thema Data Science bereits durchgeführt (z.B. [McKi 2019], [Vand 2016]). Videos können einfach aus den Notebooks heraus verlinkt werden.
- Für prozedurale Information gilt das gleiche wie für unterstützende Informationen. Die Art der Information ist lediglich auf einem niedrigeren Level.
- Übungen von Teilaufgaben können direkt innerhalb eines Notebooks gestellt, gelöst und überprüft werden. Die Lernumgebung gibt ein automatisiertes Feedback.

### 3.3 Lerninhalte

Neben der Lernumgebung sind die Lerninhalte ein Hauptbestandteil der im Projekt zu erarbeitenden Ergebnisse. Basierend auf dem 4C-ID-Modell werden im Rahmen des Projektes eine vollständige Bibliothek von Lerninhalten für die Einführung in die Programmierung mit Python entwickelt. Die Lerninhalte bestehen aus vier Teilen:

- (1) Mehrere Datenbanken mit **Lernaufgaben**. Für jeden Teilbereich der Programmierung in Python wird eine Datenbank von Aufgaben entwickelt, die in Schwierigkeit, Unterstützung

und Oberflächenmerkmalen variieren. Beispiele für diese Teilbereiche sind bedingte Anweisung, Schleifen, Funktionen oder die Verwendung von Programmbibliotheken.

- (2) **Unterstützende Informationen**, die zu jedem Teilbereich einleitende Informationen geben, welchen Zweck dieser Bereich hat, welche Python-Konstrukte es gibt, wie diese im entsprechenden Bereiche verwendet werden.
- (3) **Prozedurale Information** zu den Lernaufgaben. Hierzu werden Lehrvideos und Jupyter Notebooks erstellt, die das notwendige Vorwissen wie Syntax der Programmiersprache oder Grundlagenwissen zu den Lernaufgaben darstellen.
- (4) **Übung von Teilaufgaben** der Programmierung. Auch hierzu werden Lehrvideos und Jupyter Notebooks zur intensive Übung einzelner Teilaspekte wie z.B. der Fehlersuche in Programmen entwickelt.

### 3.4 Evaluierung

Die gesamte Lernplattform wird mit Unterstützung des Research Centre for Learning, Teaching and Technology der Open Universiteit Nederland evaluiert. Die Evaluierung folgt dem „design-based-research“ Ansatz. Das heißt, dass die Evaluierungen den gesamten Entwicklungsprozess begleiten und steuern. In einem ersten Schritt, werden die neue Lernumgebung und neue Lerninhalte zunächst mit kleinen *Pilotgruppen* im Detail evaluiert. Dies kann bereits mit 5 Studierenden geschehen, wie Usability Studien belegen. Dabei werden die Lernprozesse der Lernenden möglichst detailliert und zeitnah aufgezeichnet. Anschließend werden die Aufzeichnungen gemeinsam mit den Studierenden besprochen (sogenannte *qualitative* Evaluierung). Auf diese Art und Weise werden die größten Designprobleme entdeckt, bevor die Lernumgebung und die Lerninhalte in einem größeren Umfang umgesetzt werden. In einem zweiten Schritt, werden die Lernumgebung und Lerninhalte basierend auf dieser Pilotierung entsprechend angepasst und verbessert. Im dritten Schritt, werden die Lernumgebung und Lerninhalte in den Unterricht implementiert werden. Diese Implementierung wird mittels einer *quantitativen* Evaluierung begleitet. Dazu werden regelmäßig kurze Zwischentests aber auch Studentenevaluierungen über Fragebögen erhoben. Darüber hinaus, werden am Ende des Semesters die Lernergebnisse, sowie die Studentenevaluierungen mit denen des vorherigen Semesters statistisch verglichen, um konkrete Effekte der Implementierung zu erfassen.

## 4 In welchen Studiengängen, -abschnitten soll die Innovation eingesetzt werden

Die Lernumgebung soll zuerst in allen Modulen, in denen die Programmiersprache Python verwendet wird, eingesetzt werden. Dazu gehören:

- Einführung in die Programmierung (Bachelor Betriebswirtschaft, Wahlmodul in den ersten Semestern)
- Digital Lab (Bachelor Betriebswirtschaft, Vertiefungsmodul)
- Informationstechnik 1 (Bachelor Wirtschaftsingenieurwesen, Pflichtmodul)
- Analyse großer Datenmengen (Master Industrial Engineering, Pflichtmodul)
- Industrielle Produktion (Master Industrial Engineering, Pflichtmodul)

Die Lernumgebung soll in diesen Modulen zur Einführung in Programmiersprache Python dienen. Auf diese Weise sollen erste Erfahrungen mit der Lernumgebung gesammelt werden. Im zweiten Schritt ist der Einsatz für vertiefende Themenbereiche wie z.B. "Python und Maschinelles Lernen", "Python auf dem Raspberry Pi" oder "Python und Alexa" geplant. Der dritte Schritt zielt auf eine Ausweitung der Lernumgebung auf weitere Themenfelder wie z.B. die Programmiersprache Java (Studiengang Wirtschaftsinformatik) oder testgetriebene Entwicklung von Software.

Die Lernumgebung soll also zuerst mit einem eingeschränkten Fokus auf die Einführung in die Programmiersprache Python eingesetzt werden. Danach werden angrenzende Themenfelder hinzugenommen und die Studierendenzahlen auf der Lernumgebung vergrößert. Wenn die Lernumgebung stabil läuft, wird sie auf weitere Themenfelder (z.B. Java) mit größeren Studierendenzahlen weiterentwickelt.

## 5 Zukünftige Lehrformen

Wie in der Problemstellung beschrieben, eignen sich klassische Lehrformate wie Vorlesungen mit anschließenden Übungen nur bedingt zum Erlernen der Programmierung. Die im Rahmen des Projektes entwickelte Lernumgebung ermöglicht es, durch die erstellten Lerninhalte zukünftig deutlich individueller auf die einzelnen Studierenden einzugehen.

Die Studierenden erarbeiten sich dabei sukzessive die einzelnen Themenfelder der Programmierung mit Hilfe der in der Lernumgebung verfügbaren Lerninhalten. Dabei dient die **unterstützende Information** in Form von Videos und Jupyter Notebooks der Einführung in ein Themenfeld. Mit Hilfe der Datenbank der **Lernaufgaben** können die Studierenden dann das neu gelernte Wissen zum Lösen von authentischen Probleme anwenden. **Prozedurale Informationen** und Übungen zu Teilaufgaben helfen dabei bestimmte Teilbereich der Programmierung zu automatisieren.

Die Präsenzzeit wird zukünftig vom Dozenten genutzt werden um

- (1) Wichtige Inhalte weiter zu vertiefen und detaillierter zu erläutern
- (2) Auf individuelle Verständnisprobleme einzugehen und diese auszuräumen. Diese werden durch die Analyse der Ergebnisse bei der Bearbeitung der Lernaufgaben ermöglicht.

Zusätzlich zum Flipped Classroom Konzept erlaubt die Lernumgebung und die zugehörigen Lerninhalte die Einführung in die Programmierung für größere Gruppen von Studierenden zu öffnen. Aus der Erfahrung skalieren Übungsgruppen bei der Programmierung bis maximal 20 Teilnehmer. Ich erwarte, dass mit der Lernumgebung größere Gruppen von Studierenden deutlich individueller betreut werden können. Perspektivisch könnte die Lernumgebung soweit ausgebaut werden, dass ein vollständig selbstständiges Lernen möglich ist. Die Lernumgebung könnte dann sogar eingesetzt werden um Veranstaltungen in Form von Massive Open Online Course (MOOC) anzubieten.

## 6 Wie lassen sich Erfolg und Risiken beurteilen

Im Folgenden werden die Erfolgsfaktoren und Risiken für das Projekt eingeschätzt.

### 6.1 Beurteilung des Erfolgs der Maßnahme

In den wirtschaftswissenschaftlichen Studiengängen ist Programmieren ein freiwilliges Wahlfach. Die Abbrecherquote in den ersten Wochen ist relativ hoch, da die Fächer aus Sicht der Studierenden aufwändig und anspruchsvoll sind. Ich messe den Erfolg der Lernplattform daran, dass mehr Studierende die Programmierung wählen, dass die Abbrecherquote sinkt, dass mehr Studierende mich mit ihren Abschlussarbeiten begeistern.

In den Informatikstudiengängen (Informatik, Wirtschaftsinformatik, Media and Communication for Digital Business (MCD)) ist Programmieren ein Pflichtfach im ersten Semester. Neben Mathematik ist Programmieren das Fach, welches für die Studierenden die größte Herausforderung darstellt. Entsprechend hoch sind Abbrecherquoten (d.h. Studierende, die das Fach am Anfang des Semesters beginnen, dann aber nicht zu Ende führen) und Durchfallquoten. Ich messe den Erfolg der Lernplattform daran, dass sich diese Quoten verbessern.

Neben diesem Vergleich der typischen Kennzahlen aus dem Hochschulcontrolling, sind Evaluierungen geplant, um die Effizienz des Lernens der Studierenden besser beurteilen zu können. Durch die auf dem zentralen Server ausgeführten Aufgaben ist es möglich z.B. folgende Zahlen zu ermitteln:

- Wie viele Studierende erledigen welche Aufgabe?
- Wie viel Zeit benötigen Studierende zwischen dem Herunterladen der Aufgabe und dem Hochladen einer ersten Lösung bzw. zum Hochladen einer korrekten Lösung.
- An welcher Stelle brechen Studierende ab.

Zu diesen Fragen gibt es bislang keine Daten.

Neben den objektiven Kennzahlen ist es mir wichtig, die Freude und Begeisterung beim Programmieren zu wecken. Um dies beurteilen zu können, sind Evaluierungen durch Umfragen geplant.

## 6.2 Risiken

Es gibt ein methodisches Risiko, dass der Ansatz des 4C-ID nicht passend für die Lehre von Programmieren ist. Die Wahrscheinlichkeit des Risikos erachte ich als sehr gering. Einerseits besteht ein enger (zum Teil persönlicher) Kontakt zu den Entwicklern der Methode (u.a. van Merriënboer, Kirschner, siehe Literatur). Bei Bedarf besteht können methodische Problem somit leicht geklärt werden. Andererseits sind mehrere Projekte (außerhalb der Programmierung) bekannt, in denen die Methode erfolgreich eingesetzt wurde. Schließlich wurden Teile der Komponenten (z.B. unterstützende Information in Form einzelner Notebooks) schon mit Erfolg umgesetzt.

Es gibt das technische Risiko, dass die zu entwickelnde Plattform nicht funktioniert bzw. zu anspruchsvoll in der Umsetzung ist. Dieses Risiko erachte ich als gering. Die einzelnen Komponenten der Plattform (Notebooks, Ausführung von Notebooks auf Servern, automatisierte Tests, ...) sind alle schon ausprobiert worden bzw. sind Industriestandard. Alle Komponenten wurden als Prototyp im kleinen Bereich bei uns schon eingesetzt. Es fehlt die Zusammenführung in eine integrierte Lernumgebung. Der Entwicklungsprozess für die Lernumgebung ist so angelegt, dass die einzelnen Komponenten unabhängig voneinander funktionieren. Das heißt, dass auch Teile einer Lernumgebung nutzbar sind, wenn wider Erwarten die Implementierung der kompletten Lernumgebung nicht gelingt.

Ein drittes Risiko liegt darin, dass nicht genügend Lerninhalte (Lernaufgaben, Unterstützende Informationen, Prozedurale Informationen, Übungen von Teilaufgaben), erstellt werden. Das Risiko ist gering, da die Lerninhalte auch verwendet werden können, wenn nicht für ein komplettes Gebiet wie z.B. die Einführung in die Programmiersprache Python Lerninhalte verfügbar sind. Erste Erfahrungen bei der Erstellung von Lerninhalten aus Basis von Jupyter Notebooks werden derzeit in Rahmen von Bachelorarbeiten, Studierenden-Arbeiten und eigenen Arbeiten gemacht. Daher kann der Aufwand der Erstellung der Lerninhalte relativ genau abgeschätzt werden.

Ein letztes Risiko besteht darin, Studierenden mit Hilfe der Lernumgebung zum eigenständigen Bearbeiten der Lerninhalte zu motivieren. Dieses Risiko kann nur schwer eingeschätzt werden. Allerdings gibt es nach dem Einsatz erster Jupyter Notebooks in der Lehre sehr positives Feedback. Eine mit Hilfe eines Jupyter Notebooks erstellte Realwelt-Aufgabe (Analyse der Kommunikation einer eigenen WhatsApp-Gruppe) wurde von einigen Studierenden am Ende des Semesters als "Durchbruch" bezeichnet. Durch das Jupyter Notebook erkannten Sie, welche Möglichkeiten die Programmierung bietet und wie sich die einzelnen Komponenten des bis dahin Erlernten zusammensetzen lassen.

## 7 Wie soll die Lehrinnovation verstetigt werden?

Ziel des Innovationsprojektes ist die Entwicklung der Lernumgebung, die Erstellung von Lerninhalten sowie eine begleitende Evaluierung. Eine nachhaltige Nutzung und Weiterentwicklung wird durch mehrere Komponenten gesichert:



Bei der Entwicklung sind zwei Professoren – Prof. Drumm und Prof. Jacobs – beteiligt, die beide in den oben genannten Vorlesungen unterrichten und dadurch ein Interesse an der langfristigen Nutzung der Plattform haben. Der Fachbereich Wirtschaftswissenschaften der FH Aachen stellt derzeit wissenschaftliche Mitarbeiter zur Unterstützung der Lehre ein. Die für den Bereich Wirtschaftsinformatik zur Verfügung stehende Mitarbeiterin wird in den oben genannten Vorlesungen eingesetzt und wird nach der Fertigstellung der Lernumgebung zur Wartung, Administration und Weiterentwicklung zur Verfügung stehen.

Die Professoren im Studiengang Wirtschaftsinformatik (Fachbereiche Informationstechnik) sind bereits jetzt am Ergebnis des Projekts interessiert. Aktuell wird in diesem Studiengang die "Einführung der Programmierung" auf die Programmiersprache Java umgestellt. Eine entsprechende Lernumgebung ist auch für die Einführung in die Programmiersprache Java geplant. Prof. Drumm entwickelt gerade Ideen, wie die Lernumgebung im Bereich SAP-Programmierung eingesetzt werden kann. Sollte die Lernumgebung im Bereich Programmierung die erhofften Erfolge erzielen ist außerdem geplant die Ansätze auf Angrenzende Themenbereich wie z.B. Softwarearchitektur, Softwarequalität und Softwaretests anzuwenden.

Das Innovationsprojekt wird vom *Research Centre for Learning, Teaching and Technology* an der *Open Universiteit Nederland* in Heerlen vertreten durch assoc. Prof. Dr. Halszka Jarodzka unterstützt. Das Research Centre stellt insbesondere die Kompetenz im Bereich 4C-ID und im Bereich empirischer Evaluierung zur Verfügung. Neben der wissenschaftlichen Begleitung des Innovationsprojekts besteht von Seiten des Research Centre for Learning, Teaching and Technology außerdem das Interesse die entwickelte Lernumgebung in der Lehre und Forschung einzusetzen.

Die FH Aachen hat zwei zentrale Einrichtungen, mit denen das Innovationsprojekt kooperiert. Die Datenverarbeitungszentrale (DVZ) stellt die notwendige Hardware (Server, PC-Labore) zur Verfügung. Darüber hinaus werden Dienste wie Bereitstellung der Software, An- und Abmeldung der Studierenden, Verwaltung der Nutzerdaten von der DVZ auch langfristig übernommen. Innerhalb der DVZ betreut die Fachgruppe eLearning das bisherige Angebot in eben diesen Bereich. Sollte die Lernplattform in Zukunft von mehreren Studiengängen genutzt werden, steht eine Organisationseinheit zur Betreuung zur Verfügung. Die zweite zentrale Einrichtung der FH Aachen, das Zentrum für Hochschuldidaktik und Qualitätsentwicklung (ZHQ) und hier insbesondere das Projekt "Systematische und nachhaltige Qualitätsentwicklung in Studium und Lehre (SQSL)" unterstützt die Entwicklung und den späteren Einsatz der Lehrplattform aus methodischer Perspektive.

## 8 Auf welche Lehr- / Lernsituation kann die Innovation übertragen werden (auch andere Disziplinen)

Der in der Lernumgebung geplante Ansatz lässt sich sofort auf alle Vorlesungen übertragen, in denen Programmieren unterrichtet wird. Die Anwendung der Umgebung in einer anderen Programmiersprache sollte insbesondere dann unproblematisch sein, wenn es bereits einen Kernel in eben dieser Sprache für Jupyter-Notebooks existiert. Dies ist z.B. für Java, C, verschiedene Shell-Skriptsprachen, Ruby (vgl. [Jupy 2019]) der Fall.

Einen Überblick darüber, wo Jupyter Notebooks derzeit in der Lehre eingesetzt werden gibt das Online-Buch *jupyter4edu* [Bar\* 2019]. Das Buch ist eine Einführung in Jupyter Notebooks und erläutert die Vorteile des Systems auch aus pädagogischer Perspektive. Insbesondere werden folgende Anwendungsgebiete genannt, in den Notebooks bereits erfolgreich eingesetzt wurden: Physik, Geowissenschaften, Biologie, Data Science, Statistik. Somit besteht auch die Möglichkeit die entwickelte Lernumgebung in anderen Disziplinen einzusetzen. Möglicherweise kann dann jedoch

nicht das gesamte Lernkonzept übertragen werden. Allerdings wurden Teile offensichtlich schon in anderen Disziplinen erfolgreich angewendet.

## 9 Was versprechen Sie sich vom Austausch mit anderen Fellows?

Von dem Austausch mit anderen Fellows erhoffen ich mir insbesondere in zwei Bereichen zusätzliche Informationen und Erfahrungen.

In den vergangenen Jahren gab es fachlich vergleichbare Anträge. Im Jahr 2018 z.B. Markus Dahm "Integriertes interaktives Lernen und Lehren – insbesondere der Grundlagen der Programmierung" oder Andreas Maurischat "Entwicklung eines adaptiven Aufgabensystems zur Höheren Mathematik I". Beide Vorhaben enthalten Elemente ("adaptiv", "Ausprobieren von Codierung"), die auch in diesem Projekt eine Rolle spielen. Andererseits gibt es bei diesen Anträgen durchaus Unterschiede bei den verwendeten Methoden und bei den anvisierten Zielen. Es ist fruchtbar bei fachlich ähnlichen Vorhaben, von den gegenseitigen Erfahrungen zu lernen und sich wechselseitig zu inspirieren.

Unabhängig von fachlich verwandten Themen gibt es generelle Herausforderungen in der Hochschullehre. Beispielsweise wie Studierende motiviert oder wie die Nachhaltigkeit der unterrichteten Inhalte überprüft werden können. In der Vergangenheit (z.B. bei hdw-nrw Workshops) war dieser Austausch über Fachgrenzen hinweg immer fruchtbar.

Ich freue mich auf einen institutionalisierten Austausch, da für einen selbst initiierten Austausch im Hochschulalltag häufig keine Zeit bleibt.

## 10 Literatur

- [MCCr 2002] van Merrinboer J., Clark R., de Croock M., "Blueprints for Complex Learning: The 4C-ID-Model", Educational Technology Research and Development, Vol. 50, No. 2, pp. 39-64, 2002
- [McKi 2019] McKinney W., "Datenanalyse mit Python", O'Reilly, 2. Auflage 2019
- [MJPa 1992] van Merrinboer J., Jelsma O., Paas F., "Training for reflective expertise: A four-component instructional design model for training complex cognitive skills", Educational Technology Research and Development, Vol. 40, No. 2, pp. 23-43, 1992
- [MeKi 2018] van Merrienboer J., Kirschner P., "Ten Steps to Complex Learning", Third Edition, Routledge, New York, 2017
- [Vand 2016] VanderPlas J., "Python Data Science Handbook: Essential Tools for working with Data", O'Reilly, 2016

## 11 Internetquellen

- [Bar\* 2019] Barba L., Barker L., Blank D., Brown J., Downey A., George T., Heagy L., Mandli K., Moore J., Lippert D., Niemeyer K., Watkins R., West R., Wickes E., Willing C., Zingale M.: Teaching and Learning with Jupyter, <https://jupyter4edu.github.io/jupyter-edu-book/index.html>
- [Cour 2019] Coursera, Introduction to Data Science with Python, <https://jupyter4edu.github.io/jupyter-edu-book/index.html>,
- [Jupy 2019] Jupyter Documentation, Kernels (Programming Languages), <https://jupyter.readthedocs.io/en/latest/projects/kernels.html>

[Petr 2019]

Petrou T., "Master Data Analysis with Python – Volume – Parts 1 & 2",  
<https://online.dunderdata.com/courses/master-data-analysis-with-python-intro-to-pandas>