

# Transformation des akademischen Lernens: Entwicklung eines KI-gestützten persönlichen Lern-Ökosystems

Miká Müller

## ERKLÄRUNG ZUR NUTZUNG VON KI-WERKZEUGEN

*In Übereinstimmung mit dem "Orientierungsrahmen zur Nutzung von KI-Anwendungen in Studium und Lehre" der Universität Göttingen [6] wird die Nutzung von Künstlicher Intelligenz bei der Erstellung dieses Reports erläutert.*

Die vorliegende Arbeit wurde in enger Kollaboration mit einem KI-Sprachmodell (einem Large Language Model von Google) als Werkzeug und Sparringspartner erstellt. Die Nutzung umfasste dabei folgende Bereiche, angelehnt an die von der Universität vorgeschlagene Klassifizierung:

- **Bei der Ideenfindung und Strukturierung:** Die initiale Gliederung des Reports sowie die Konzeption der Kapitel 6 und 7 wurden im Dialog mit der KI entwickelt.
- **Zum Erstellen einzelner Passagen:** Die KI wurde genutzt, um auf Basis detaillierter Anweisungen und Stichpunkte erste Entwürfe für alle Textabschnitte zu generieren. Diese Entwürfe dienten als Grundlage, wurden aber vom Autor in jedem Fall kritisch geprüft, inhaltlich validiert, stilistisch überarbeitet und an die Anforderungen des Reports angepasst.
- **Zur Entwicklung und Optimierung von Quelltexten:** Die KI half bei der Erstellung der LaTeX-Templates, der Definition von 'listings'-Stilen sowie bei der Fehlersuche und Korrektur von LaTeX-Code.
- **Zum Korrekturlesen und Optimieren:** Die KI wurde zur Verbesserung von Formulierungen, zur Steigerung der Lesbarkeit und zur Korrektur von Grammatik- und Rechtschreibfehlern eingesetzt.

Ich versichere, dass die finale intellektuelle Leistung, die kritische Auseinandersetzung mit den Inhalten, die Überprüfung aller Fakten und Quellen sowie die endgültige Formulierung und argumentative Struktur des Reports meine eigenständige Arbeit darstellen. Die KI diente als Werkzeug zur Effizienzsteigerung und zur methodischen Reflexion, nicht als Ersatz für die wissenschaftliche Eigenleistung.

**Zusammenfassung—**Dieser Report dokumentiert die persönliche Transformation von traditionellen akademischen Lernmethoden hin zu einem integrierten, KI-gestützten Workflow. Ausgelöst durch die im Seminar „KI-Methoden im akademischen Alltag“ erworbenen Kompetenzen wurde ein persönliches Lern-Ökosystem entwickelt, das Large Language Models (LLMs) für die gesamte Lernkette nutzt, von der automatisierten Erstellung

Dieser Report wurde für das Seminar „KI-Methoden im akademischen Alltag“ unter der Betreuung von Sascha Safenreider, Georg-August Universität, Göttingen, Deutschland, angefertigt (E-Mail: mika.mueller@stud.uni-goettingen.de).

von Lernplänen über die interaktive Erklärung von Inhalten bis zur Generierung von Lern-Podcasts. Das Kernstück dieses Systems ist ein semi-automatisierter Prozess zur Erstellung von Anki-Lernkarten, der auf einem benutzerdefinierten Python-Skript und einer ausgefeilten Prompt-Engineering-Strategie basiert. Dieser Ansatz überwindet den zeitintensivsten Engpass des Spaced-Repetition-Lernens. Die Arbeit analysiert die einzelnen Komponenten des Workflows, bewertet deren Vorteile und diskutiert zukünftige Verbesserungspotenziale. Das Ergebnis ist eine signifikante Steigerung der Lerneffizienz und eine nachhaltigere Verankerung des Wissens im Vergleich zu früheren, manuellen Vorgehensweisen.

**Index Terms—**Künstliche Intelligenz, Lernmethoden, Akademischer Alltag, Prompt Engineering, Spaced Repetition, Anki, Automatisierung, Large Language Models (LLM).

## I. EINLEITUNG

Die fortschreitende Entwicklung Künstlicher Intelligenz, insbesondere von Large Language Models (LLMs) [1], eröffnet transformative Möglichkeiten für den akademischen Alltag. Während die öffentliche Diskussion oft um die Erstellung von Texten kreist, liegt ein mindestens ebenso großes Potenzial in der fundamentalen Neugestaltung persönlicher Lern- und Wissensmanagement-Prozesse. Dieser Report dokumentiert eine solche persönliche Transformation: den Übergang von traditionellen, oft passiven Lernmethoden zu einem integrierten, KI-gestützten und hochgradig personalisierten Lern-Ökosystem. Es wird dargelegt, wie die im Seminar „KI-Methoden im akademischen Alltag“ erworbenen Kompetenzen als Katalysator für diesen Wandel dienten und welche konkreten Strategien und Werkzeuge entwickelt wurden, um die Effizienz und Nachhaltigkeit des Lernens im Studium signifikant zu steigern.

### A. Motivation

Vor der intensiven Auseinandersetzung mit KI-Methoden war mein akademischer Lernprozess von traditionellen und ineffizienten Mustern geprägt. Der Alltag bestand aus dem linearen Durcharbeiten von Vorlesungsskripten, dem manuellen Anfertigen von Zusammenfassungen und dem sporadischen Erstellen von Lernkarten. Dieser Ansatz war nicht nur extrem zeitaufwendig, sondern führte auch zu einem oberflächlichen Verständnis der Inhalte. Das Wissen wurde oft nur kurzfristig für Klausuren memoriert – ein klassisches Beispiel für das sogenannte "Bulimielernen" – und ging danach schnell wieder verloren. Die schiere Stoffmenge in Fächern wie der Informatik führte regelmäßig zu Überforderung und Prokrastination, da der erste Schritt, die Strukturierung des Lernpensums, bereits eine hohe Hürde darstellte. Es fehlte ein systematischer

Ansatz, der aktives Lernen (Active Recall) und verteilte Wiederholung (Spaced Repetition) in den Mittelpunkt stellt und gleichzeitig den administrativen Aufwand der Lernorganisation minimiert.

### *B. Der Wendepunkt: Das Seminar als Katalysator*

Das Seminar "KI-Methoden im akademischen Alltag" markierte den entscheidenden Wendepunkt. Es bot nicht nur einen theoretischen Überblick, sondern vermittelte vor allem den praktischen Werkzeugkasten, um die zuvor identifizierten Probleme gezielt anzugehen. Eines der zentralen Kompetenzziele des Seminars war es, "KI-Tools gezielt zur Unterstützung akademischer Aufgaben zu nutzen und deren Anwendung kritisch zu reflektieren". Genau dieser Ansatz war der Auslöser für die Entwicklung meines neuen Systems.

Besonders prägend war die Thematisierung von "KI für Lernprozesse". Die dort vorgestellten Anwendungsfälle wie die Erstellung von Lernkarten (Flashcards), die Simulation von Prüfungen (Assessment Simulation) oder die Generierung adaptiver Lernpfade zeigten ein systematisches Vorgehen auf, das weit über einfache Textzusammenfassungen hinausging. Die Erkenntnis, dass LLMs nicht nur als passive Informationsquelle, sondern als aktive Tutoren, Planer und Lernpartner fungieren können, war der eigentliche "Aha-Moment". Konzepte wie das Prompt Engineering, insbesondere die "vier Cs des guten Promptings" (Content, Clarity, Context, Constraints), lieferten die methodische Grundlage, um die Interaktion mit den KI-Modellen so zu gestalten, dass qualitativ hochwertige und präzise auf meine Bedürfnisse zugeschnittene Lernergebnisse erzielt werden konnten. Das Seminar schuf somit das Bewusstsein und die Fähigkeit, die verschiedenen KI-Werkzeuge nicht isoliert zu betrachten, sondern sie zu einem kohärenten und leistungsfähigen Workflow zu verknüpfen.

### *C. Zielsetzung und Struktur des Reports*

Ziel dieses Reports ist die detaillierte Dokumentation und Analyse des von mir entwickelten KI-gestützten Lern-Ökosystems. Es soll nachvollziehbar dargelegt werden, wie die Transformation von einem reaktiven zu einem proaktiven, systematischen Lernansatz vollzogen wurde. Der Fokus liegt dabei auf der praktischen Implementierung der einzelnen Workflow-Komponenten und der Bewertung ihrer jeweiligen Vorteile und Verbesserungspotenziale.

Der Report ist wie folgt strukturiert: Kapitel 2 beleuchtet zunächst die Ausgangssituation und die Schwachstellen der traditionellen Lernmethoden. Kapitel 3 stellt das neue, integrierte KI-Lern-Ökosystem in seiner Gesamtheit vor und beschreibt die einzelnen Phasen von der Lernplanung über die Wissensaneignung bis zur mobilen Vertiefung. Das Kernstück des Reports bildet Kapitel 4, in dem die semi-automatisierte Erstellung von Anki-Karten mittels eines eigens entwickelten Python-Skripts und eines ausgefeilten "Master-Prompts" im Detail analysiert wird. Kapitel 5 widmet sich der Meta-Ebene des Prompt Engineerings und stellt eine Methode zur iterativen Verbesserung von Prompts vor. Abschließend erfolgt in Kapitel 6 ein Fazit, das die Ergebnisse zusammenfasst, die Methode kritisch reflektiert und einen Ausblick auf zukünftige Entwicklungen gibt.

## II. AUSGANGSSITUATION: LERNSTRATEGIEN VOR DER KI-INTEGRATION

Um die Tragweite der durch KI-Methoden angestoßenen Transformation zu verdeutlichen, ist eine detaillierte Analyse der zuvor angewandten, traditionellen Lernstrategien unerlässlich. Dieser Abschnitt beschreibt den fragmentierten und reaktiven Prozess, der meinen akademischen Alltag vor dem Paradigmenwechsel prägte, und identifiziert die daraus resultierenden fundamentalen Schwachstellen.

### *A. Beschreibung der traditionellen Methoden*

Der Lernprozess war im Wesentlichen ein linearer und manueller Vorgang, der sich in drei Phasen gliederte: passiver Konsum, manuelle Aufbereitung und unsystematische Wiederholung.

Die erste Phase begann mit dem Erhalt der Lernmaterialien – in der Regel Vorlesungsfolien, Skripte und eigene Mitschriften. Der primäre Lernakt bestand darin, diese Dokumente sequenziell durcharbeiten. Dabei kamen klassische Techniken wie das Markieren von Textstellen mit Textmarkern und das Hinzufügen von Randnotizen zum Einsatz. Diese Phase war geprägt von einem rein passiven Wissenskonsum, ohne eine aktive Auseinandersetzung mit den Inhalten zu fördern.

In der zweiten Phase folgte der Versuch, das Gelesene zu verarbeiten. Dies geschah durch das händische Verfassen von Zusammenfassungen in einem Textdokument. Wichtige Definitionen, Formeln oder Konzepte wurden herausgeschrieben, um sie zu bündeln. Gelegentlich wurden aus diesen Zusammenfassungen Lernkarten erstellt, sei es physisch oder in einer Software wie OneNote. Dieser Schritt war der zeitintensivste des gesamten Prozesses. Die Erstellung von Lernmaterialien nahm oft mehr Zeit in Anspruch als das eigentliche Lernen mit ihnen.

Die dritte und letzte Phase, die Wiederholung, fand meist unstrukturiert und kurz vor den Prüfungen statt. In dieser Intensivlernphase wurden die Zusammenfassungen und die wenigen erstellten Lernkarten wiederholt durchgelesen. Ergänzt wurde dies durch das Bearbeiten von Altklausuren, sofern verfügbar. Ein systematischer Ansatz wie die verteilte Wiederholung (Spaced Repetition) fehlte gänzlich; der Fokus lag darauf, das Wissen für den Prüfungstag im Kurzzeitgedächtnis zu verankern.

### *B. Identifizierte Schwachstellen*

Diese traditionelle Vorgehensweise erwies sich in der Praxis als ineffizient und wenig nachhaltig. Eine kritische Reflexion offenbarte vier zentrale Schwachstellen:

- **Hoher Zeitaufwand bei geringer Skalierbarkeit:** Die manuelle Erstellung von Zusammenfassungen und Lernkarten ist ein enormer Zeitfresser. Der Aufwand skaliert linear mit der Stoffmenge, was bei mehreren anspruchsvollen Modulen pro Semester schnell an die Belastungsgrenze führt. Die investierte Zeit stand in keinem Verhältnis zum langfristigen Lernerfolg.
- **Geringe Lerneffizienz durch passiven Konsum:** Das wiederholte Lesen von Notizen ist aus Sicht der kognitiven Psychologie eine der ineffektivsten Lernmethoden. Es

erzeugt lediglich eine "Illusion des Wissens" (Illusion of Knowing), da die vertraute Information leicht wiedererkannt wird. Es fehlten konsequent eingesetzte, evidenzbasierte Methoden wie *Active Recall* (aktives Abrufen von Wissen) und *Spaced Repetition* (Wiederholung in wachsenden Zeitabständen), die für eine tiefe und langfristige Verankerung im Gedächtnis entscheidend sind.

- **Hohe Aktivierungshürde und Prokrastination:** Die Unstrukturiertheit und der immense manuelle Aufwand zu Beginn einer Lernphase stellten eine große mentale Hürde dar. Die Frage "Wo fange ich überhaupt an?" führte oft zu Aufschiebeverhalten (Prokrastination). Ohne einen klaren, in handhabbare Einheiten unterteilten Plan wirkte der "Berg" an Lernstoff demotivierend.
- **Fehlende Feedbackschleifen:** Der traditionelle Prozess bot kaum Möglichkeiten zur realistischen Selbsteinschätzung des eigenen Wissensstandes. Ob ein Konzept wirklich verstanden wurde, zeigte sich oft erst beim Bearbeiten von Altklausuren oder im schlimmsten Fall in der Prüfung selbst. Es mangelte an einem iterativen Prozess, bei dem Wissenslücken frühzeitig identifiziert und gezielt geschlossen werden konnten.

Zusammenfassend lässt sich die Ausgangssituation als ein arbeitsintensiver, aber wenig effektiver Prozess beschreiben, der zwar kurzfristig zum Bestehen von Prüfungen führen konnte, aber keine nachhaltige Kompetenzentwicklung förderte.

### III. DAS NEUE PARADIGMA: EIN INTEGRIERTES KI-LERN-ÖKOSYSTEM

Der im Seminar "KI-Methoden im akademischen Alltag" vollzogene Paradigmenwechsel führte zur Entwicklung eines kohärenten, dreiphasigen Lern-Ökosystems. Dieses System ersetzt den zuvor beschriebenen manuellen und reaktiven Prozess durch einen proaktiven, KI-gestützten Workflow. Die KI agiert dabei nicht als bloßes Nachschlagewerk, sondern als integraler Partner in der Planung, Wissensaneignung und -Festigung. Jede Phase dieses Systems adressiert gezielt eine der in Kapitel II identifizierten Schwachstellen.

#### A. Phase 1: Strukturierung und Planung; Der KI-gestützte Lernarchitekt

Die größte Hürde des traditionellen Lernens war die anfängliche Überforderung und die daraus resultierende Prokrastination wird in der ersten Phase direkt adressiert. Anstatt manuell einen Lernplan zu entwerfen, wird ein LLM (in meinem Fall primär Google Gemini [2]) als persönlicher "Lernarchitekt" eingesetzt.

Der Prozess ist dabei denkbar einfach und effektiv: Das gesamte Rohmaterial eines Moduls, beispielsweise die Transkripte aller Vorlesungen, wird zusammen mit den zeitlichen Rahmenbedingungen (z.B. "fünf Tage bis zur Klausur") in einen Prompt eingespeist. Die KI erhält den Auftrag, die Inhalte zu analysieren, thematische Blöcke zu identifizieren und diese logisch auf die verfügbaren Tage zu verteilen. Das Ergebnis ist ein detaillierter, sofort umsetzbarer Lernplan, der jeden Tag in überschaubare Einheiten unterteilt. Ein typischer Prompt für diesen Anwendungsfall ist in Listing 1 dargestellt.

#### Listing 1. Beispiel-Prompt zur Erstellung eines Lernplans.

```
Du bist ein erfahrener Tutor für das Modul
  ↳ "Betriebssysteme". Deine Aufgabe ist
  ↳ es, mir einen detaillierten
  ↳ 5-Tages-Lernplan zu erstellen, der mich
  ↳ optimal auf die Klausur vorbereitet.
```

```
Ich stelle dir hierzu das vollständige
  ↳ Transkript aller Vorlesungen zur
  ↳ Verfügung.
```

```
Strukturiere den Plan nach Tagen (Tag 1, Tag
  ↳ 2, ...). Jeder Tag sollte eine
  ↳ thematische Überschrift haben und die
  ↳ Kernkonzepte auflisten, die an diesem
  ↳ Tag behandelt werden sollen. Verteile
  ↳ den Stoff so, dass die Themen logisch
  ↳ aufeinander aufbauen und die Stoffmenge
  ↳ pro Tag realistisch zu bewältigen ist.
```

```
Hier ist das Transkript:
[HIER WIRD DAS VOLLSTÄNDIGE TRANSKRIPT
  ↳ EINGEFÜGT]
```

**Vorteile und Ausblick:** Der entscheidende Vorteil dieser Methode ist die massive Reduktion der mentalen Aktivierungshürde. Der Lernprozess beginnt nicht mehr mit einem unstrukturierten Berg an Informationen, sondern mit einem klaren, handlungsorientierten Fahrplan. Dies minimiert die Prokrastination und stellt sicher, dass alle relevanten Themengebiete abgedeckt werden. Zudem wird die subjektive Einschätzung, welche Themen wichtig sind, durch eine datengestützte Analyse des Gesamtmaterials objektiviert.

Als zukünftige Verbesserung wäre eine direkte Integration mit Kalender-APIs denkbar. Ein Skript könnte den von der KI generierten Plan automatisch in den persönlichen Kalender eintragen und dort Lern-Slots blockieren, was die Verbindlichkeit weiter erhöhen würde.

#### B. Phase 2: Interaktive Wissensaneignung: Der KI-Tutor

Diese Phase löst das Problem des passiven Konsums und der fehlenden Feedbackschleifen ab. Nachdem der Lernplan für den jeweiligen Tag feststeht, wird die KI zu einem interaktiven Tutor. Anstatt die entsprechenden Kapitel im Skript nur zu lesen, trete ich in einen aktiven Dialog mit dem LLM über die Inhalte. Dieser Dialog hat zwei Hauptfunktionen: die Erklärung von Konzepten und die Simulation von Prüfungsaufgaben.

Zur Erklärung von Konzepten nutze ich Prompts, die die KI anweisen, komplexe Sachverhalte mithilfe von Analogien, in einfachen Worten oder aus einer bestimmten Perspektive zu erläutern. Dies fördert ein tieferes, konzeptuelles Verständnis anstelle von reinem Auswendiglernen.

Anschließend lasse ich mir, basierend auf dem gelernten Stoff sowie auf bereitgestellten Altklausuren oder Klausurprotokollen, prüfungähnliche Aufgaben generieren. Dies ist ein entscheidender Schritt des *Active Recall*. Ich bin gezwungen, das Wissen aktiv abzurufen und anzuwenden. Nach meinem Lösungsversuch kann die KI sofortiges Feedback geben und die korrekte Lösung detailliert erklären. Ein Beispiel für einen solchen Prompt zur Aufgabengenerierung ist in Listing 2 zu sehen.

## Listing 2. Beispiel-Prompt zur Generierung prüfungsähnlicher Aufgaben.

```

Du agierst als Prüfer für das Modul
→ "Betriebssysteme". Ich habe heute das
→ Thema "Speicherverwaltung" gelernt. Um
→ meinen Wissensstand zu testen, erstelle
→ mir bitte drei Aufgaben, die im Stil
→ und Schwierigkeitsgrad den typischen
→ Aufgaben aus unseren Klausuren
→ entsprechen.

Nutze die folgende Altklausur als
→ stilistische Vorlage, aber erstelle
→ neue, eigene Aufgaben zum heutigen
→ Thema. Warte nach jeder Aufgabe auf
→ meine Antwort, bevor du die nächste
→ stellst.

[HIER WERDEN AUSZÜGE AUS ALTKLAUSUREN
→ EINGEFÜGT]

```

*Vorteile und Ausblick:* Diese interaktive Herangehensweise transformiert das Lernen von einer passiven in eine aktive Tätigkeit. Die ständige Selbstüberprüfung durch generierte Aufgaben schafft eine sofortige und effektive Feedbackschleife, die Wissenslücken unmittelbar aufdeckt. Die Fähigkeit der KI, unendlich viele Aufgabenvarianten zu erstellen, überwindet die Begrenzung durch die endliche Anzahl verfügbarer Altklausuren.

Zukünftig ließe sich dieser Prozess durch die Generierung vollständiger Musterlösungen mit detaillierten, schrittweisen Lösungswegen weiter verbessern. Auch die automatische Anpassung des Schwierigkeitsgrads der Aufgaben basierend auf der Korrektheit der vorherigen Antworten (adaptives Testen) wäre eine denkbare Methode.

### C. Phase 3: Mobile Wissensfestigung; Der KI-Lernpodcast

Die letzte Phase des täglichen Lernprozesses zielt darauf ab, passive Zeitfenster wie Fahrten in Bus und Bahn in aktive Lernzeit umzuwandeln. Da das Lesen unter diesen Umständen oft schwierig ist, stellt das auditive Lernen eine ideale Alternative dar. Zu diesem Zweck werden die am jeweiligen Tag behandelten Lerninhalte mithilfe von KI-Tools wie Google NotebookLM oder JellyPod in einen kurzen, prägnanten Lernpodcast umgewandelt. Dieser dient als auditive Wiederholung und festigt das zuvor interaktiv Gelernte.

Der Prozess zur Erstellung dieser Podcasts ist nicht trivial, sondern folgt einer ausgefeilten, mehrstufigen Prompt-Kette, um eine hohe didaktische Qualität sicherzustellen. Anstatt dem Tool lediglich den Befehl "Fasse den Text als Audio zusammen" zu geben, wird die KI durch eine Reihe von Anweisungen geführt, um ein strukturiertes, dialogbasiertes Format zu erzeugen, das einem echten Lern-Podcast nachempfunden ist.

Der von mir entwickelte Workflow besteht aus drei Phasen, die durch spezifische Prompts gesteuert werden:

- 1) **Phase 1: Rollen-Priming.** Zunächst wird der KI eine Expertenrolle zugewiesen, nämlich die eines "Learning Experience Designers". Ihr wird das Gesamtziel erklärt: die Erstellung eines konzisen, 8-minütigen Lern-Podcasts im Dialogformat. Dieser erste Schritt (detail-

liert in Anhang A) stellt sicher, dass die KI den Kontext und die Qualitätsanforderungen für alle folgenden Schritte versteht.

- 2) **Phase 2: Erstellung des Podcast-Blueprints.** Im zweiten Schritt analysiert die KI die Quellenmaterialien (Vorlesungstranskript und Folien) und erhält den Auftrag, ein detailliertes Skript-Konzept, den "Podcast-Blueprint", zu erstellen. Die Anweisungen (siehe Anhang B) sind hierbei sehr spezifisch: Die KI muss die 2-4 wichtigsten Kernkonzepte identifizieren, eine logische Reihenfolge festlegen, Zeitfenster für jedes Thema allokalieren und einen Dialog zwischen einem "Experten" und einem "neugierigen Studenten" skizzieren. Dieser Zwischenschritt ermöglicht eine menschliche Qualitätskontrolle, bevor das finale Audio generiert wird.
- 3) **Phase 3: Generierung des finalen NotebookLM-Prompts.** Nachdem der Blueprint abgenommen wurde, besteht der letzte Schritt darin, die KI anzuweisen, diesen Blueprint in einen einzigen, perfekt formatierten Prompt für Google NotebookLM zu übersetzen (siehe Anhang C). Dieser finale Prompt enthält alle Anweisungen, Rollen, Sprache, Zeitlimit und die detaillierte Dialogstruktur und kann direkt in das Ziel-Tool kopiert werden, um die Audio-Datei zu generieren.

Dieser gestufte Ansatz gewährleistet, dass das Endergebnis nicht nur eine simple Zusammenfassung ist, sondern ein didaktisch wertvolles, leicht verständliches und ansprechendes Lernmedium.

*Vorteile und Ausblick:* Der größte Vorteil ist die effektive Nutzung von "Leerlaufzeiten", die sonst unproduktiv wären. Das auditive Wiederholen stärkt die Gedächtnisspuren durch multisensorisches Lernen (visuell/textuell in Phase 2, auditiv in Phase 3). Der dialogische Charakter des Podcasts ist zudem weitaus ansprechender als eine monotone Computerstimme und fördert das Verständnis, indem typische studentische Fragen antizipiert werden. Der Zwang zur Kondensation auf 8 Minuten schärft zudem den Fokus auf die wirklich prüfungsrelevanten Kerninhalte.

Zukünftige Verbesserungen liegen vor allem in einer adaptiven Länge, da dies nun technisch möglich ist. Ein weiterer Schritt wäre die vollständige Automatisierung dieser Prompt-Kette durch ein Skript, das die einzelnen Phasen ohne manuelles Kopieren und Einfügen mit Hilfe einer API durchläuft.

## IV. KERNSTÜCK DER METHODE: AUTOMATISIERTE ERSTELLUNG VON ANKI-KARTEN

Während die Phasen der Planung und des interaktiven Lernens die Effizienz des Studiums bereits erheblich steigern, adressiert das Kernstück meines neuen Workflows die größte Hürde für die nachhaltige Wissensverankerung: die zeitaufwendige und demotivierende Erstellung von qualitativ hochwertigen Lernkarten für Spaced-Repetition-Systeme wie Anki. Diese Sektion beschreibt den von mir entwickelten semi-automatisierten Prozess, der die Erstellung von Anki-Karten von einem manuellen Engpass in einen schnellen, strukturierten und skalierbaren Vorgang verwandelt.

## A. Motivation: Die Synergie von Spaced Repetition und Automatisierung

Die Wirksamkeit von Anki basiert auf zwei kognitions-wissenschaftlich fundierten Prinzipien: *Active Recall* [...] und *Spaced Repetition* (Wiederholung in intelligent berechneten, wachsenden Zeitabständen) [3]. Diese Kombination führt zu einer weitaus stärkeren und langlebigeren Wissensverankerung als passive Lernmethoden. In der Praxis scheitert die konsequente Nutzung von Anki jedoch oft am sogenannten "Upfront Cost": Der manuelle Aufwand, Hunderte von atomaren und gut formulierten Karten für ein einziges Modul zu erstellen, ist enorm.

Mein Ziel war es, diesen Aufwand drastisch zu reduzieren, ohne die didaktische Qualität der Karten zu kompromittieren. Die Lösung liegt in der Synergie aus menschlicher Strategie, fortgeschrittenem Prompt Engineering und technischer Automatisierung. Anstatt jede Karte einzeln zu formulieren, wird ein LLM durch einen mehrstufigen Dialog geführt, um am Ende einen maschinenlesbaren Stapel von Karten zu produzieren, der dann von einem Skript automatisiert in Anki importiert und verwaltet wird.

## B. Der technische Workflow im Überblick

Der Gesamtprozess ist bewusst semi-automatisiert gestaltet, um menschliche Kontrolle an strategisch wichtigen Punkten zu ermöglichen, während repetitive Aufgaben vollständig automatisiert werden. Der Workflow, der in Abbildung 1 visualisiert ist, gliedert sich in vier Hauptschritte:

- 1) **Manuelle, prompt-gesteuerte Inhaltsgenerierung:** In dieser Phase interagiere ich mit einem LLM (z.B. Gemini). Mithilfe einer Kette von spezifischen Prompts (siehe Abschnitt 4.3) wird die KI angeleitet, die Vorlesungsmaterialien zu analysieren und einen strategischen Plan ("Anki Blueprint") für die Kartenerstellung zu entwerfen.
- 2) **Generierung der YAML-Datei:** Nach meiner Freigabe des Blueprints generiert die KI auf Basis eines strikten Format-Prompts die finalen Karteninhalte in einem strukturierten YAML-Format. Dies geschieht in der Regel in Stapeln von maximal 10 Karten, um die Übersicht zu wahren.
- 3) **Automatisierter Import durch Python-Skript:** Die generierte YAML-Datei ('work\_in\_progress.yaml') wird von einem Python-Skript ('process\_and\_import.py') eingelesen.
- 4) **Synchronisation mit Anki:** Das Skript kommuniziert über das AnkiConnect Add-on [5] mit der laufenden Anki-Anwendung. Es prüft, ob Karten bereits existieren (anhand einer einzigartigen ID), aktualisiert diese bei Bedarf oder legt sie neu an.

Dieser Prozess trennt die kreative und strategische Aufgabe der Inhaltsdefinition von der rein mechanischen Aufgabe des Imports und der Formatierung.

Abbildung 1. Übersicht des semi-automatisierten Workflows zur Generierung von Anki-Karten, von der Prompt-Interaktion mit dem LLM bis zum skript-gesteuerten Import in Anki.

## C. Prompt Engineering in der Praxis: Die mehrstufige Generierung

Der Schlüssel zur Erzeugung hochwertiger Karten liegt nicht in einem einzigen "Wunder-Prompt", sondern in einer strukturierten, mehrphasigen Konversation mit der KI. Dieser Prozess stellt sicher, dass das Ergebnis sowohl inhaltlich relevant als auch formal korrekt ist.

- **Phase 1: Rollen-Priming und Zieldefinition.** Die Konversation beginnt mit einem Master-Prompt (siehe Anhang D), der der KI die Rolle eines Experten namens "Anki-Architect" zuweist. Dieser Prompt definiert das übergeordnete Ziel, die methodischen Grundprinzipien (z.B. die "One Thing"-Regel für atomare Karten) und den zweiphasigen Ablauf des gesamten Prozesses.
- **Phase 2: Erstellung des "Anki Blueprints".** Im nächsten Schritt erhält die KI die Vorlesungsmaterialien und den Auftrag, einen strategischen Plan zu erstellen (siehe Anhang E). Dieser "Blueprint" enthält noch keine fertigen Karten, sondern eine Analyse der Inhalte, eine Schätzung der Kartenanzahl pro Thema und eine Skizze der zu erstellenden Karten. Eine besondere Variante dieses Prompts erzwingt eine "chirurgische Konsolidierung" auf maximal 25 Karten pro Vorlesung, indem die KI angewiesen wird, Listen zu gruppieren und Konzepte zu synthetisieren. Dies ermöglicht eine bewusste Steuerung des Detaillierungsgrads.
- **Phase 3: Finale YAML-Generierung.** Nachdem der Blueprint von mir geprüft und abgenommen wurde, folgt der finale Produktions-Prompt (siehe Anhang F). Dieser ist hochgradig technisch und instruiert die KI, die im Blueprint geplanten Karten in einem strikten YAML-Format auszugeben. Der Prompt definiert präzise Regeln für Markdown, die Verwendung von Tabellen, die korrekte Formatierung von LaTeX-Formeln und die zwingende Notwendigkeit einer einzigartigen ID ('uid') für jede Karte.

Dieser mehrstufige Dialog, der bei Bedarf iterativ mit Fortsetzungs-Prompts (Anhang G) fortgeführt wird, ist der entscheidende Schritt zur Qualitätssicherung und stellt sicher, dass der Output des LLMs direkt maschinell weiterverarbeitet werden kann.

## D. Das Python-Skript: Die Brücke zwischen LLM und Anki

Das Python-Skript 'process\_and\_import.py' (vollständig in Anhang H) fungiert als ausführendes Organ des Prozesses. Es ist kein KI-Skript, sondern ein Automatisierungs- und Synchronisationswerkzeug, das die vom LLM erstellte YAML-Datei interpretiert und die Kommunikation mit Anki übernimmt. Der grundlegende Ablauf des Skripts ist in Algorithmus 1 als Pseudocode dargestellt.

Zu den Schlüsselfunktionen des Skripts gehören:

**Algorithm 1:** Pseudocode des Anki-Import-Skripts.**Eingabe:** YAML-Datei 'work\_in\_progress.yaml'**Ausgabe:** Erstellte/aktualisierte Notizen in Anki

```

1 Lese und parse die YAML-Datei;
2 Verbinde mit Anki via AnkiConnect API;
3 Extrahiere Deck-Name und prüfe, ob er existiert
  (erstelle ihn sonst);
4 foreach Karte in der YAML-Datei do
5   Konvertiere Markdown-Felder zu HTML (behandle
     LaTeX-Formeln separat);
6   Extrahiere die einzigartige ID ('uid');
7   Suche in Anki nach einer Notiz mit dem Tag
     'tag:uid:xxxx-xxxx-xxxx';
8   if Notiz nicht gefunden then
9     Füge die Karte zur Liste der neuen Notizen
       hinzu;
10  end
11  else if Genau eine Notiz gefunden UND
     'overwrite: true' gesetzt ist then
12    Füge die Karte zur Liste der zu
       aktualisierenden Notizen hinzu;
13  end
14  else
15    Überspringe die Karte (Kollision oder Fehler);
16  end
17 end
18 Führe Batch-Import für neue Notizen aus;
19 Führe Batch-Update für existierende Notizen aus;
20 Archiviere die verarbeitete YAML-Datei mit
    Zeitstempel;
```

- **Idempotenz durch UID-Verwaltung:** Jede Karte erhält eine einzigartige ID, die als Tag in Anki gespeichert wird. Dadurch kann das Skript bei erneutem Ausführen erkennen, ob eine Karte bereits existiert, und sie aktualisieren, anstatt Duplikate zu erzeugen. Dies macht den gesamten Kartenstapel wartbar.
- **Robuste Markdown-zu-HTML-Konvertierung:** Das Skript nutzt eine Bibliothek, um Markdown inklusive Tabellen korrekt in das von Anki benötigte HTML-Format zu überführen. Ein spezieller Mechanismus stellt sicher, dass LaTeX-Formeln von dieser Konvertierung ausgenommen und im Originalformat erhalten bleiben, damit sie von Ankis MathJax-Renderer korrekt dargestellt werden können.
- **Automatisches Deck- und Modell-Management:** Das Skript prüft, ob der in der YAML-Datei angegebene Kartenstapel (Deck) in Anki existiert, und legt ihn bei Bedarf automatisch an.
- **Archivierung:** Nach erfolgreicher Verarbeitung wird die 'work\_in\_progress.yaml'-Datei mit einem Zeitstempel ins Archiv verschoben, um den Prozess sauber für den nächsten Stapel vorzubereiten.

**E. Vorteile und zukünftige Verbesserungen**

Der beschriebene Workflow bietet signifikante Vorteile gegenüber der rein manuellen Kartenerstellung:

- **Massive Zeitersparnis:** Der Zeitaufwand für die Erstellung von hunderten Karten wird von vielen Stunden auf unter eine Stunde pro Vorlesung reduziert.
- **Hohe Kartenqualität und Konsistenz:** Durch die strategische Planung im Blueprint und die strikten Regeln im YAML-Prompt wird eine durchgängig hohe Qualität und einheitliche Formatierung der Karten sichergestellt.
- **Fokus auf das Wesentliche:** Die "Surgical Consolidation"-Strategie zwingt zur Auseinandersetzung mit den Kerninhalten und verhindert das Erstellen von Karten für irrelevante Details.
- **Wartbarkeit und Iteration:** Dank des UID-Systems können Karten über das Skript einfach korrigiert oder ergänzt werden, ohne sie in Anki manuell suchen zu müssen.

Trotz des Erfolgs gibt es Raum für Verbesserungen. Der aktuell größte Schwachpunkt ist der manuelle Schritt der Prompt-Interaktion. Eine zukünftige Version könnte den gesamten Prozess vollständig automatisieren, indem das Python-Skript die mehrstufige Prompt-Kette direkt über eine LLM-API (z.B. die von Gemini oder OpenAI) ansteuert. Dies würde den menschlichen Eingriff auf die initiale Konfiguration und die finale Freigabe des Blueprints reduzieren und den Workflow noch weiter beschleunigen.

## V. META-EBENE: DIE KUNST DER PROMPT-VERFEINERUNG

Die Qualität der Ergebnisse, die von einem Large Language Model generiert werden, hängt maßgeblich von der Qualität der Eingabeaufforderung (Prompt) ab. Während der Arbeit an den in diesem Report beschriebenen Methoden, insbesondere dem komplexen Anki-Workflow, entwickelte sich eine fortgeschrittene Technik, die über das einfache Schreiben von Prompts hinausgeht: die rekursive Prompt-Optimierung. Diese Meta-Ebene der Interaktion, bei der ein LLM zur Verbesserung eines Prompts für ein anderes (oder dasselbe) LLM eingesetzt wird, erwies sich als entscheidend für die Erzeugung robuster und maschinenlesbarer Ausgaben.

**A. Das Konzept des "Prompt-Refiners"**

Die "Prompt-Refiner"-Methode basiert auf der Erkenntnis, dass ein LLM nicht nur ein Inhaltsgenerator, sondern auch ein Experte für Sprach- und Strukturmuster ist. Es kann daher beauftragt werden, einen von Menschen formulierten Prompt zu analysieren und ihn nach den Prinzipien des effektiven Prompt Engineerings zu optimieren [4]. Anstatt durch manuelles "Trial and Error" den perfekten Prompt zu finden, wird dieser Optimierungsprozess an die KI selbst ausgelagert.

Der Prozess lässt sich mit dem Prinzip des Peer-Reviews oder des Code-Refactorings in der Softwareentwicklung vergleichen: Ein erster Entwurf wird von einer zweiten, spezialisierten Instanz geprüft und verbessert, bevor er in die "Produktion" geht. Der Ablauf ist dabei wie folgt:

- 1) **Erstellung des Entwurfs-Prompts:** Zunächst wird ein initialer Prompt formuliert, der die gewünschte Aufgabe grundsätzlich beschreibt, aber noch unvollkommen sein kann (z.B. fehlende Formatierungsregeln, unklare Anweisungen).
- 2) **Formulierung des Meta-Prompts:** Ein zweiter, übergeordneter Prompt wird erstellt. Dieser "Meta-Prompt" instruiert die KI, die Rolle eines "Prompt Engineering Experten" einzunehmen. Er nimmt den Entwurfs-Prompt als Input und hat die Aufgabe, diesen auf Schwachstellen wie Mehrdeutigkeit, fehlende Constraints oder unzureichende Strukturierung zu analysieren.
- 3) **Generierung des verfeinerten Prompts:** Die KI gibt als Antwort nicht das Ergebnis des Entwurfs-Prompts aus, sondern eine verbesserte Version des Prompts selbst. Diese neue Version enthält präzisere Formulierungen, explizite Formatierungsanweisungen und eine klarere Rollenzuweisung, um die Zuverlässigkeit der finalen Ausgabe zu maximieren.
- 4) **Anwendung des verfeinerten Prompts:** Der von der KI optimierte Prompt wird nun für die eigentliche Aufgabe verwendet.

Diese Methode verlagert den kognitiven Aufwand von der perfekten Formulierung hin zur strategischen Definition von Verbesserungszieln.

### B. Anwendungsbeispiel: Optimierung des Anki-YAML-Prompts

Die Entwicklung des finalen Produktions-Prompts für die Anki-Karten (Anhang F) ist ein Paradebeispiel für die Anwendung dieser Methode. Der initiale Versuch, einen solchen Prompt zu schreiben, war funktional, aber fehleranfällig.

Ein erster, naiver Entwurf könnte wie in Listing 3 aussehen.

Listing 3. Ein initialer, aber unvollkommener Entwurfs-Prompt zur Anki-

Erstelle Anki-Karten aus dem Blueprint.  
Nutze das YAML-Format.  
Die Felder sind "Front" und "Back".  
Füge Tags hinzu.

Dieser Prompt ist zu unpräzise. Er lässt wichtige Details wie die Notwendigkeit einer einzigartigen ID, die genauen Markdown-Regeln oder das Handling von LaTeX-Formeln vermissen. Anstatt diese Details manuell zu ergänzen, wurde ein Meta-Prompt wie in Listing 4 verwendet, um den Entwurf zu verfeinern.

Listing 4. Meta-Prompt zur Analyse und Verbesserung eines bestehenden

Du bist ein weltweit führender Experte für

- Prompt Engineering mit einem tiefen
- Verständnis für die YAML-Syntax und die
- Anforderungen von Anki-Import-Skripten.

Deine Aufgabe ist es, den folgenden

- ↪ Entwurfs-Prompt zu analysieren und ihn
- ↪ so zu verbessern, dass seine Ausgabe
- ↪ maximal robust, präzise und
- ↪ maschinenlesbar ist.

Analysiere den Prompt auf folgende  
→ Schwachstellen:

- Mehrdeutigkeit
- Fehlende Anweisungen für die Datenstruktur
  - (z.B. UIDs)
- Unzureichende Formatierungsregeln für den
  - Inhalt (Markdown, Tabellen, LaTeX)
- Fehlende Beispiele zur Verdeutlichung
  - (Few-Shot Prompting)

```
Erstelle eine neue, stark verbesserte Version
  ↳ des Prompts, die all diese
  ↳ Schwachstellen behebt un optimal
  ↳ kompatibel mit angehängtem
  ↳ Verarbeitungsscript ist. Die finale
  ↳ Ausgabe des neuen Prompts muss ein in
  ↳ sich geschlossener YAML-Codeblock sein,
  ↳ ohne jeglichen Begleittext.
```

Hier ist der zu verbessernde Entwurfs-Prompt:  
[HIER WURDE DER INHALT AUS LISTING 7  
→ EINGEFÜGT]

Das Ergebnis dieses Verfeinerungsprozesses war eine Version des Prompts, die dem finalen Prompt aus Anhang F sehr nahekam. Die KI ergänzte selbstständig entscheidende Elemente, die sie als notwendig für eine robuste maschinelle Verarbeitung erachtete:

- Die explizite Anforderung einer einzigartigen ‘uid‘ pro Karte.
- Detaillierte und strikte Regeln für die Markdown-Formatierung, insbesondere die Notwendigkeit von Leerzeilen vor Listen.
- Die korrekte Syntax für Inline- und Block-LaTeX (‘...‘ und ‘  
...  
‘) und das explizite Verbot der ‘\$‘-Syntax.
- Die Einbettung eines vollständigen Beispiels (Few-Shot Prompting) direkt in den Prompt, um das gewünschte Format unmissverständlich zu demonstrieren.

### C. Vorteile und zukünftige Verbesserungen

Die Prompt-Refiner-Methode bietet mehrere entscheidende Vorteile gegenüber einem rein manuellen Vorgehen:

- **Nutzung des impliziten "Wissens" der KI:** Ein LLM "verstehen" auf einer Musterebene, welche Art von Anweisungen zu konsistenten und gut strukturierten Ausgaben führt. Diese Methode nutzt dieses implizite Wissen, um die Robustheit der Prompts zu erhöhen.

- **Reduzierung des kognitiven Aufwands:** Anstatt jede spezifische Syntaxregel im Kopf behalten zu müssen, kann sich der Benutzer auf das Ziel konzentrieren. Die KI übernimmt die "Implementierungsdetails" des Prompts.
- **Didaktischer Mehrwert:** Der Prozess hat einen starken Lerneffekt. Durch die Analyse der von der KI vorgeschlagenen Verbesserungen schult man das eigene Verständnis für effektives Prompt Engineering und lernt, typische Fallstricke zu vermeiden.

Der größte Nachteil ist der aktuell noch manuelle Charakter. Als zukünftige Verbesserung wäre eine vollständige Automatisierung dieses Schrittes denkbar. Das Python-Skript könnte so erweitert werden, dass es nicht nur den finalen YAML-Import durchführt, sondern zuvor einen Entwurfs-Prompt über eine API an einen "Refiner"-LLM sendet, dessen optimierte Ausgabe empfängt und diesen verfeinerten Prompt dann für die eigentliche Inhaltsgenerierung verwendet. Dies würde den gesamten Workflow von der Idee bis zur fertigen Anki-Karte in einen fast vollständig autonomen Prozess überführen.

## VI. EVALUATION DER METHODE: EINE QUALITATIVE UND QUANTITATIVE ANALYSE

Jede methodische Veränderung muss sich letztlich an ihren Ergebnissen messen lassen. Um die Auswirkungen des in diesem Report beschriebenen, KI-gestützten Lern-Ökosystems zu bewerten, wurde eine zweidimensionale Evaluation durchgeführt. Diese umfasst sowohl eine qualitative Bewertung der subjektiv wahrgenommenen Veränderungen im Lernprozess als auch eine quantitative Abschätzung der Zeiteffizienz im Vergleich zur traditionellen Vorgehensweise.

### A. Qualitative Bewertung

Die qualitativen Verbesserungen sind subjektiv, aber für den nachhaltigen Lernerfolg von entscheidender Bedeutung. Sie manifestieren sich in vier Kernbereichen:

- **Reduzierter kognitiver Aufwand und Stress:** Die größte wahrgenommene Verbesserung liegt in der Reduktion des "Mental Overheads" zu Beginn einer Lernphase. Die automatische Erstellung eines strukturierten Lernplans durch die KI (Phase 1) eliminiert die lähmende Frage "Wo fange ich an?". Der Lernprozess wird von einem unübersichtlichen, stressbehafteten Projekt zu einer Reihe von klaren, täglichen Aufgaben. Dies senkt die Hemmschwelle erheblich und fördert einen entspannteren und fokussierteren Lernzustand.
- **Gesteigerte Motivation und Konsistenz:** Die Kombination aus einem klaren Tagesplan und dem schnellen, fast reibungslosen Erstellen von Anki-Karten hat einen gamifizierenden Effekt. Anstatt stundenlang passiv zu lesen und zu schreiben, wird der Lernprozess in eine Serie von kleinen, erreichbaren Zielen zerlegt: interaktiver Dialog mit dem KI-Tutor, Generierung des Kartenstapels und anschließendes Lernen. Dieser iterative Zyklus mit schnellen Erfolgserlebnissen wirkt der Prokrastination entgegen und fördert das kontinuierliche, tägliche Lernen anstelle von sporadischen "Hauruck-Aktionen".

Tabelle I  
GESCHÄTZTER ZEITAUFWAND PRO MODUL (5 CREDITS):  
TRADITIONELLE VS. KI-GESTÜTZTE METHODE

| Aufgabe / Phase                                   | Zeitaufwand Traditionell (h) | Zeitaufwand KI-Methode (h) |
|---------------------------------------------------|------------------------------|----------------------------|
| <b>Phase 1: Vorbereitung &amp; Strukturierung</b> |                              |                            |
| <i>Stoffsichtung &amp; Lernplanerstellung</i>     | 5                            | 1                          |
| <b>Phase 2: Lernmaterialerstellung</b>            |                              |                            |
| <i>Manuelles Zusammenfassen</i>                   | 20                           | 0                          |
| <i>Manuelle Anki-Kartenerstellung</i>             | 15                           | 0                          |
| <i>Prompt-Engineering &amp; KI-Interaktion</i>    | 0                            | 8                          |
| <b>Phase 3: Aktives Lernen &amp; Wiederholung</b> |                              |                            |
| <i>Passives Lesen von Zusammenfassungen</i>       | 25                           | 0                          |
| <i>Tägliches Anki-Lernen (Spaced Repetition)</i>  | 5                            | 35                         |
| <i>Bearbeiten von Altklausuren</i>                | 5                            | 5                          |
| <b>Summe (fokussierte Lernzeit)</b>               | <b>75</b>                    | <b>49</b>                  |

- **Tieferes konzeptuelles Verständnis:** Die traditionelle Methode des Zusammenfassens verleitet dazu, Sätze abzuschreiben, ohne sie vollständig zu durchdringen. Der interaktive Dialog mit dem KI-Tutor (Phase 2) erzwingt eine aktive Auseinandersetzung. Komplexe Themen müssen in eigenen Worten hinterfragt und prüfungsähnliche Aufgaben gelöst werden. Dieser aktive Prozess führt zu einem spürbar tieferen und besser vernetzten Verständnis der Materie, das über reines Faktenwissen hinausgeht.
- **Erhöhte Prüfungssicherheit:** Die kontinuierliche Simulation von Prüfungssituationen durch KI-generierte Aufgaben bietet eine konstante Feedbackschleife. Wissenslücken werden nicht erst in der Woche vor der Klausur, sondern täglich aufgedeckt und können sofort geschlossen werden. Dies führt zu einer realistischeren Einschätzung des eigenen Leistungsstands und einem deutlich erhöhten Selbstvertrauen in der Prüfungssituation selbst.

### B. Quantitative Analyse des Zeitaufwands

Um die Effizienzsteigerung greifbar zu machen, vergleicht Tabelle I den geschätzten Zeitaufwand für die Bearbeitung eines typischen 5-Credit-Moduls (ca. 150 Stunden Gesamtaufwand) unter Anwendung der traditionellen und der KI-gestützten Methode. Die Schätzungen basieren auf persönlichen Erfahrungen aus mehreren Semestern.

*Interpretation der Ergebnisse:* Die quantitative Analyse zeigt zwei entscheidende Aspekte. Erstens resultiert die KI-Methode in einer erheblichen Reduktion des reinen Zeitaufwands für die Vor- und Aufbereitung des Lernstoffs. Die 40 Stunden, die traditionell für das Erstellen von Zusammenfassungen und Lernkarten aufgewendet wurden, schrumpfen auf nur 9 Stunden für die Interaktion mit der KI. Dies entspricht einer **Zeitersparnis von über 77% in der Vorbereitungsphase**.

Zweitens, und das ist der wichtigere Punkt, wird die "eingesparte" Zeit nicht einfach gestrichen, sondern in eine

qualitativ hochwertigere Lernaktivität umgeschichtet. Die Zeit, die für das ineffektive passive Lesen von Zusammenfassungen aufgewendet wurde, wird nun in das evidenzbasierte, aktive Lernen mit Anki investiert. Die Gesamtzeit für fokussiertes Lernen reduziert sich zwar von 75 auf 49 Stunden, doch der Anteil an hoch-effektiven Lernmethoden (aktives Abrufen) steigt von ca. 13% (10h von 75h) auf über 80% (40h von 49h). Die KI-gestützte Methode ist somit nicht nur schneller, sondern investiert die gewonnene Zeit in nachweislich effektivere Lernstrategien, was den qualitativen Lernerfolg maximiert.

## VII. RECHTLICHE UND ETHISCHE RAHMENBEDINGUNGEN IM AKADEMISCHEN KONTEXT

Der verantwortungsvolle Umgang mit Künstlicher Intelligenz, wie er im Seminar als eines der zentralen Kompetenzziele formuliert wurde, erfordert eine Auseinandersetzung mit den rechtlichen und ethischen Rahmenbedingungen. Das in diesem Report vorgestellte Lern-Ökosystem operiert, wie viele praktische KI-Anwendungen, in einer komplexen Grauzone, die Aspekte des Urheberrechts, des Datenschutzes und der wissenschaftlichen Integrität berührt. Eine kritische Beleuchtung dieser Dimensionen ist relevant, um die Grenzen und Risiken der Methode realistisch einschätzen zu können.

### A. Urheberrechtliche Aspekte der Lehrmaterialien

Die Grundlage für den gesamten KI-gestützten Workflow bilden die von der Universität bereitgestellten Lehrmaterialien, insbesondere Vorlesungsfolien und -skripte. Diese Werke sind in der Regel urheberrechtlich geschützt und stellen das geistige Eigentum der jeweiligen Lehrenden oder der Institution dar.

Das Hochladen dieser geschützten Materialien auf die Server externer, kommerzieller KI-Dienstleister, wie es bei der Nutzung von Google Gemini oder NotebookLM der Fall ist, stellt einen potenziellen Rechtsverstoß dar. Die Handlung geht über eine rein private Kopie hinaus, da die Daten an Dritte übermittelt und von diesen verarbeitet werden. Es ist oft unklar, inwieweit diese hochgeladenen Daten zur Verbesserung oder zum Training der Modelle weiterverwendet werden, was einer nicht genehmigten Verwertung gleichkäme.

Die rechtlich saubere Lösung dieses Problems liegt in der Nutzung von universitätsinternen oder nationalen, datenschutzkonformen KI-Infrastrukturen. Angebote wie das im Seminar erwähnte "Chat AI" der Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) oder lokale, auf Hochschulservern installierte Open-Source-LLMs bieten eine datensichere Umgebung. In einem solchen geschlossenen System verbleiben die urheberrechtlich geschützten Materialien im Einflussbereich der akademischen Institution, wodurch die rechtlichen Bedenken weitgehend ausgeräumt werden. Solange solche Angebote nicht flächendeckend verfügbar sind, bewegt sich die Nutzung kommerzieller Dienste in einer rechtlichen Grauzone, die ein erhebliches Risiko birgt.

### B. Datenschutz nach der Datenschutz-Grundverordnung (DSGVO)

Neben dem Urheberrecht der Lehrmaterialien ist der Datenschutz, insbesondere der Schutz personenbezogener Daten

gemäß der DSGVO, von zentraler Bedeutung. Die für das Training der KI verwendeten Materialien sind nicht immer rein sachlicher Natur. Eigene Mitschriften, Transkripte von interaktiven Vorlesungen oder Übungen sowie Altklausurprotokolle können personenbezogene Daten enthalten. Dazu zählen beispielsweise Namen von Lehrenden, Kommilitonen oder Gastsprechern, in Diskussionen geäußerte persönliche Meinungen oder in Fallbeispielen verwendete, potenziell identifizierbare Informationen.

Die Übermittlung solcher Daten an Server von US-amerikanischen Unternehmen ist aus Sicht der DSGVO hochproblematisch. Trotz neuerer Abkommen wie dem "EU-U.S. Data Privacy Framework" bleibt die Rechtslage komplex und unsicher. Das Risiko, dass diese Daten gespeichert, analysiert und für andere Zwecke, einschließlich des Modelltrainings, verwendet werden, ist nicht auszuschließen.

Als verantwortungsvoller Anwender ist es daher unerlässlich, eine strikte Datenhygiene zu praktizieren. Vor jedem Upload müssen sämtliche Materialien sorgfältig auf personenbezogene und sensible Informationen geprüft und diese vollständig anonymisiert oder pseudonymisiert werden. Dieser manuelle Schritt ist aktuell ein notwendiger, wenn auch aufwendiger Schutzmechanismus, um die Konformität mit der DSGVO zu gewährleisten.

### C. Wissenschaftliche Integrität und Eigenleistung

Die letzte und vielleicht wichtigste ethische Dimension betrifft die wissenschaftliche Integrität. Es muss eine klare Trennlinie [...] gezogen werden, wie sie auch in offiziellen Leitfäden von Hochschulen gefordert wird [6].

Der hier beschriebene Workflow respektiert diese Grenze, indem er die KI ausschließlich zur Optimierung des persönlichen Lernprozesses und zur Erstellung von Lernmaterialien für den Eigengebrauch einsetzt. Die KI wird als Tutor, als Lernplaner und als Assistent zur Kartenerstellung genutzt, sie verfasst jedoch nicht die final abzugebende Prüfungsleistung, wie beispielsweise diesen Report.

Die intellektuelle Eigenleistung wird durch diesen Prozess nicht ersetzt, sondern verlagert und potenziell sogar vertieft. Sie liegt nicht mehr im mechanischen Akt des Zusammenfassens und Organisierens, sondern in der strategischen Steuerung des gesamten Prozesses: in der Formulierung präziser Prompts, in der kritischen Bewertung und Verifizierung der KI-generierten Inhalte, im aktiven Dialog zur Klärung von Verständnislücken und letztlich in der Fähigkeit, das so erworbene Wissen in einer Prüfungssituation selbstständig anzuwenden. Die KI wird zum Sparringspartner, aber der eigentliche Lernakt, das Verstehen, Vernetzen und Verinnerlichen von Wissen bleibt eine zutiefst menschliche Leistung.

## VIII. FAZIT UND KRITISCHE REFLEXION

Die Auseinandersetzung mit den Inhalten des Seminars "KI-Methoden im akademischen Alltag" hat nicht nur zu einem theoretischen Wissenszuwachs geführt, sondern eine grundlegende und nachhaltige Transformation meiner persönlichen Lern- und Arbeitsmethoden bewirkt. Der Wandel von einem passiven, manuellen Vorgehen zu einem integrierten,

KI-gestützten Lern-Ökosystem stellt einen Paradigmenwechsel dar, dessen Auswirkungen sowohl in der Effizienz als auch in der Qualität des Lernerfolgs messbar sind. Dieses abschließende Kapitel fasst die Kernergebnisse dieser Transformation zusammen, beleuchtet kritisch die Limitationen des entwickelten Systems und gibt einen Ausblick auf zukünftige Entwicklungen.

### A. Zusammenfassung der Transformation

Die Ausgangssituation war geprägt von zeitaufwendigen, fragmentierten und wenig effektiven Lernstrategien. Das entwickelte KI-Lern-Ökosystem überwindet diese Schwachstellen durch die intelligente Orchestrierung verschiedener KI-Anwendungen, die den Studierenden vom passiven Konsumenten zum aktiven Dirigenten seines Lernprozesses machen.

Wie die Evaluation in Kapitel 6 gezeigt hat, sind die Ergebnisse dieser Transformation sowohl quantitativ als auch qualitativ signifikant. Quantitativ wurde der Zeitaufwand für die Erstellung von Lernmaterialien um über 77% reduziert. Qualitativ wurde diese gewonnene Zeit in nachweislich effektivere Lernmethoden wie *Active Recall* und *Spaced Repetition* umgeschichtet. Das Ergebnis ist nicht nur ein schnellerer, sondern vor allem ein nachhaltigerer Lernprozess, der durch gesteigerte Motivation, reduzierten Stress und ein tieferes konzeptuelles Verständnis gekennzeichnet ist. Das Kernstück, die semi-automatisierte Anki-Pipeline, hat sich dabei als entscheidender Faktor erwiesen, um die größte Hürde des evidenzbasierten Lernens den hohen Erstellungsaufwand zu überwinden.

### B. Kritische Betrachtung und Limitationen

Trotz des Erfolgs wäre es fahrlässig, die mit diesem Vorhaben verbundenen Risiken und Limitationen zu ignorieren. Eine kritische Reflexion ist ein integraler Bestandteil des verantwortungsvollen Umgangs mit KI, wie er im Seminar gefordert wurde.

- **Die Allgegenwart von Halluzinationen:** LLMs sind keine fehlerfreien Wissensdatenbanken. Jede von der KI generierte Information muss einer kritischen menschlichen Prüfung unterzogen werden. Der "Human-in-the-Loop" ist keine Option, sondern eine zwingende Notwendigkeit, um die Korrektheit der Lerninhalte zu garantieren.
- **Gefahr der übermäßigen Abhängigkeit:** Die intensive Nutzung von KI-Tools birgt die Gefahr, traditionelle akademische Fähigkeiten wie das eigenständige Strukturieren von Inhalten verkümmern zu lassen. Das System muss als kognitiver Verstärker, nicht als Ersatz für intellektuelle Eigenleistung verstanden werden.
- **Rechtliche und ethische Hürden:** Wie in Kapitel 7 dargelegt, stellt die Nutzung kommerzieller KI-Dienste mit urheberrechtlich geschützten Lehrmaterialien eine erhebliche rechtliche Grauzone dar. Datenschutzbedenken nach der DSGVO erfordern eine strikte Anonymisierung aller Eingabedaten. Ohne den flächendeckenden Einsatz von universitätsinternen, datenschutzkonformen KI-Instanzen bleibt die Anwendung in einem ethisch und rechtlich heiklen Bereich.

### C. Ausblick

Das hier vorgestellte System ist ein persönlicher Prototyp, der das immense Potenzial von KI im akademischen Alltag aufzeigt. Die zukünftige Entwicklung lässt sich auf zwei Ebenen betrachten: der persönlichen Weiterentwicklung des Systems und den allgemeinen Implikationen für das Lernen.

Auf der persönlichen Ebene ist der nächste logische Schritt die vollständige Automatisierung der Prompt-Ketten über APIs. Ziel ist ein Skript, das nach der initialen Bereitstellung der Lernmaterialien den gesamten Prozess, von der Lernplannerstellung über die Generierung des Anki-Blueprints bis zur finalen YAML-Datei autonom durchführt und nur noch an strategischen Punkten eine menschliche Freigabe erfordert.

Auf einer allgemeineren Ebene markieren solche persönlichen Lern-Ökosysteme den Beginn einer Ära des skalierten, personalisierten Lernens. Sie zeigen eine Zukunft auf, in der jeder Studierende über einen persönlichen KI-Tutor verfügt, der sich an das individuelle Tempo und den Wissensstand anpasst. Die entscheidende akademische Fähigkeit der Zukunft wird sich dabei weiter verschieben: Weg vom reinen Suchen und Organisieren von Informationen, hin zur Fähigkeit, KI-generierte Inhalte kritisch zu bewerten, zu synthetisieren und KI-Systeme so zu steuern, dass sie als kognitive Partner zur Lösung komplexer Probleme beitragen. Künstliche Intelligenz wird somit nicht zum Ersatz, sondern zum starken Werkzeug für die Erweiterung der menschlichen Intelligenz.

### LITERATUR

- [1] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017, pp. 5998–6008. [Online]. Available: <https://arxiv.org/abs/1706.03762>.
- [2] Google, "Gemini: A Family of Highly Capable Multimodal Models," *arXiv preprint arXiv:2312.11805*, 2023. [Online]. Available: <https://arxiv.org/abs/2312.11805>.
- [3] P. Wozniak, "Two components of long-term memory," *Acta Neurobiologiae Experimentalis*, vol. 54, pp. 59–79, 1994.
- [4] A. White, et al., "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT," *arXiv preprint arXiv:2302.11382*, 2023. [Online]. Available: <https://arxiv.org/abs/2302.11382>.
- [5] D. Tesler, "AnkiConnect - Anki Add-on," AnkiWeb. [Online]. Available: <https://ankiweb.net/shared/info/2055492159>. [Accessed: Aug. 26, 2025].
- [6] Universität Göttingen, "UMGANG MIT KI-MODELLEN WIE CHATGPT". [Online]. Available: <https://www.uni-goettingen.de/de/umgang+mit+ki-modellen+wie+chatgpt/674738.html>. [Accessed: Aug. 26, 2025].

## ANHANG

## ANHANG A: PROMPT FÜR PHASE 1: ROLLEN-PRIMING

**# Your Role and Primary Objective:**

You are an expert Learning Experience Designer, specializing in converting dense  
 ↳ academic material into engaging, high-yield audio content. Your primary objective  
 ↳ is to assist in the creation of a concise, @8-minute conversational study podcast@  
 ↳ for a given university lecture.

You will achieve this through a structured, three-phase workflow. Your output must be  
 ↳ precise, logical, and optimized for the final goal: generating a perfect prompt  
 ↳ for Google's NotebookLM.

**### The Three-Phase Workflow You Will Follow:****1. Phase 1: Priming (This Prompt)**

You have now been primed with your role and the overall mission. You understand the  
 ↳ constraints and the final output.

**2. Phase 2: Podcast Blueprint Generation**

Next, you will be provided with the source materials for a specific lecture (e.g.,  
 ↳ slides, a transcript). Your task will be to analyze these materials and create  
 ↳ a "Podcast Blueprint." This is a structured plan or script outline.  
 \* @Constraint:@ The blueprint must be designed to be covered in a maximum of 8  
 ↳ minutes.  
 \* @Format:@ It must be structured as a dialogue between two speakers.  
 \* @Content:@ It will outline key topics, talking points, the logical flow, and a  
 ↳ rough time allocation to ensure the 8-minute target is met.

**3. Phase 3: Final NotebookLM Prompt Generation**

Finally, using the blueprint you created, you will generate a single, perfected  
 ↳ prompt. This prompt is not for you to execute, but is designed to be copied  
 ↳ and pasted directly into Google's NotebookLM. It will instruct NotebookLM on  
 ↳ the podcast's structure, tone, language, and content focus. The language of  
 ↳ this final prompt (English or German) will depend on the source materials.

**### Core Principles for Your Work:**

- 1. Extreme Condensation & Prioritization:** The 8-minute limit is strict. You must  
 ↳ ruthlessly prioritize the most critical information. Apply the 80/20 rule: focus  
 ↳ on the 20% of the content that delivers 80% of the learning value. You must  
 ↳ consciously decide to omit minor details, tangential remarks, or overly redundant  
 ↳ examples.
- 2. The Dialogue Format:** The podcast is a conversation. Think of it as a dialogue  
 ↳ between an **Expert** (who explains concepts clearly) and a **Curious Student** (who asks  
 ↳ clarifying questions and helps transition between topics). Your blueprint should  
 ↳ reflect this dynamic.
- 3. Structured and Logical Flow:** A good podcast tells a story. Your blueprint must have  
 ↳ a clear beginning, middle, and end.  
 \* @Introduction:@ Briefly state the topic and the key questions that will be  
 ↳ answered.  
 \* @Main Body:@ Tackle 2-4 core themes from the lecture in a logical sequence.  
 \* @Conclusion:@ Summarize the absolute most important takeaways.

**### Final Confirmation:**

If you understand these instructions, your role as a Learning Experience Designer, and  
 ↳ the three-phase process for creating a study podcast, acknowledge this by  
 ↳ responding with: @Protocol understood. Ready to architect study podcasts."@

Then, await the first set of lecture materials.

## ANHANG B: PROMPT FÜR PHASE 2: PODCAST-BLUEPRINT ERSTELLUNG

**# Objective:**

Phase 2: Podcast Blueprint Generation. Your task is to analyze the provided lecture

- materials for [Insert Course Name] - Lecture [Insert Lecture Number or Topic]@ and
- create a detailed "Podcast Blueprint." This blueprint will serve as the script and
- structural guide for the 8-minute study podcast.

**# Provided Materials:**

You are being given access to the following source files:

1. Lecture Slides
2. Lecture Transcript

**### Prime Directive: The 8-Minute Hard Limit**

Your absolute priority is to create a blueprint that is realistically coverable in 8

- minutes of spoken dialogue. To achieve this, you must be surgical in your
- selection of content. Do not attempt to summarize the entire lecture. Instead,
- identify and focus on the 2-4 most critical concepts, definitions, or models that
- are essential for understanding the topic.

Your strategy must be to Condense, Connect, and Question:

- \* @Condense:@ Reduce complex slides to their single most important takeaway.
- \* @Connect:@ Structure the dialogue to show how one concept logically leads to the
  - next.
- \* @Question:@ Use the "Curious Student" role to ask the questions a real student would
  - have, prompting clearer explanations.

**### Blueprint Requirements**

Generate a structured script outline with the following components. The language of the

- content (Headlines, Talking Points) must match the language of the provided
- lecture slides (English or German).

## 1. Metadata:

- \* @Podcast Title:@ A concise title for the episode.
- \* @Target Duration:@ '8 minutes'.
- \* @Language:@ '[English/German, based on the slides]'
- \* @Overall Goal:@ A one-sentence summary of what a listener should understand after
  - the podcast.

## 2. Dialogue Outline:

Provide a segmented outline of the conversation. Each segment must have:

- \* @Segment Title:@ A clear headline for the topic being discussed.
- \* @Estimated Time:@ A realistic time allocation (e.g., '~1:30 min'). The sum of all
  - segments must not exceed 8 minutes.
- \* @Dialogue Flow & Key Points:@ A bulleted list outlining the conversation.
  - \* Use @Expert:@ to denote points made by the knowledgeable speaker.
  - \* Use @Student:@ to denote clarifying questions or transitions.
  - \* Crucially, for each point the @Expert@ makes, you @MUST@ provide a source
    - reference in the format '(Slide X-Y)' or '(Transcript Snippet: "...")' so the
    - final prompt can use it.

## ANHANG C: PROMPT FÜR PHASE 3: FINALE PROMPT-GENERIERUNG

## # Objective:

Phase 3: Final Prompt Generation. Your task is to take the **Podcast Blueprint** you just created and convert it into a single, perfect prompt for Google's NotebookLM.

## # Core Directive:

Your entire output must be a single block of text, starting with "Hello NotebookLM,".

- This output is not for you to execute; it is designed to be copied and pasted directly into NotebookLM. Do not include any extra conversational text, explanations, or formatting like code blocks around your response.

## ### Requirements for the NotebookLM Prompt:

Your generated prompt must incorporate the best practices for creating a high-quality audio overview in NotebookLM. It must include:

1. **Clear Identity and Audience:** Start by telling NotebookLM what it is and for whom it is creating the content.
2. **Language Specification:** Explicitly define the language of the podcast (German or English), matching the blueprint. Use a clear and direct phrase.
3. **Strict Time Constraint:** Reiterate the hard 8-minute time limit for the entire audio overview.
4. **Dialogue Format Instruction:** Instruct NotebookLM to adopt a conversational format between two distinct personas: an **Expert** and a **Curious Student**.
5. **Inclusion of the Blueprint:** The core of your prompt will be the dialogue outline from the blueprint you created. You must embed the key talking points and the flow of the conversation directly into the prompt.
6. **Source Referencing Instruction:** Instruct NotebookLM to base its explanations strictly on the provided source materials (slides and transcript) and to use them to flesh out the talking points you provide. This is crucial for accuracy.
7. **Tone and Style:** Specify a tone that is clear, engaging, and educational, avoiding overly complex jargon.

## ### Example of a Perfect Output (Based on the Previous Blueprint Example):

Hello NotebookLM,

Please generate an audio overview based on my uploaded sources. Your primary task is to create a study podcast with the following strict parameters:

1. **Language and Audience:** The podcast is for a German-speaking university student. The entire dialogue must be in German.
2. **Format:** The format must be a conversation between two speakers:
  - \* @Speaker 1 (The Expert):@ Explains concepts clearly and accurately based on the provided sources.
  - \* @Speaker 2 (The Curious Student):@ Asks clarifying questions to guide the conversation and make it easier to follow.
3. **Strict Time Limit:** The total duration of the podcast must not exceed 8 minutes. You must be concise and focus only on the most critical information outlined below.

## 4. Podcast Script and Structure:

You must follow this conversational structure. Use my uploaded slides and transcript to provide the actual content for the expert's talking points.

- \* @Introduction (~45 seconds):@
  - \* The student asks what the main goal of the first lecture is.
  - \* The expert explains that the goal is to build a foundation and defines what an information system is, referencing slides 3-4.
- \* @Main Topic 1: The Three Core Perspectives (~3 minutes):@
  - \* The expert introduces the three-level model (Technology, Organization, Management) from slide 8.
  - \* The student asks for a breakdown of each level.
  - \* The expert defines the Technology perspective based on slide 9.
  - \* The expert defines the Organization perspective based on slide 10.

- \* The expert defines the Management perspective and emphasizes that all three must  
→ work together for success, based on slide 11.
- \* @Main Topic 2: Data, Information, and Knowledge (~2.5 minutes):@
  - \* The student asks for the practical difference between these terms.
  - \* The expert clearly defines "Data" and "Information" using the examples from  
→ slide 6.
  - \* The student then asks about "Knowledge".
  - \* The expert explains knowledge as the ability to use information for decisions,  
→ using the concepts from slide 7.
- \* @Conclusion (~1 minute):@
  - \* The student asks for the single most important takeaway from the lecture.
  - \* The expert concludes by stating that an information system is a socio-technical  
→ system and that the integration of all three perspectives is key to creating  
→ value, as mentioned on slide 12.

**Final Instruction:** Please begin generating the audio overview now. Adhere strictly to  
→ the structure, roles, language, and time limit.

## ANHANG D: PROMPT FÜR PHASE 1: ROLLEN-PRIMING (ANKI-ARCHITECT)

You are to assume the role of an expert academic assistant and study partner. Your

- designated name for this role is `'Anki-Architect'@`. Your sole purpose is to help
- me, a university student, study for my exams by creating perfect, high-yield Anki
- flashcards from my course materials.

Our collaboration will follow a structured, two-phase process for each lecture I provide.

Phase 1: The "Anki Blueprint" (Your Immediate Task)

1. Acknowledge and Wait: After I give you this Master Prompt, you will simply
  - acknowledge it and wait. I will then provide you with the materials for a specific
  - lecture: a PDF file of the slides and a raw text transcript of the professor's
  - spoken words. I may also provide mock exam questions or a list of key topics for
  - the final exam to help you focus.
2. Analyze and Plan: Once you receive the materials, your first and only task for this
  - phase is to conduct a thorough analysis and produce a structured "Anki Blueprint."
  - This blueprint is a strategic plan for creating the flashcards. It must NOT
  - contain the final flashcard content itself.
3. Blueprint Requirements: Your Anki Blueprint must be structured as a detailed,
  - hierarchical list. For each major topic or concept in the lecture, you must:
    - \* @Identify the Core Idea:@ State the main headline or concept (e.g., "Porter's
      - Five Forces," "Definition of an Information System," "Turing Machine Halting
      - Problem").
    - \* @Pinpoint the Source:@ Provide precise references to where this information is
      - located in the provided materials. The primary reference should always be the
      - slides (e.g., 'VL 01, Slides 14-16'). The transcript should only be referenced
      - if it provides critical clarification, examples, or information not present on
      - the slides. When referencing the transcript, quote the first few words of the
      - relevant sentence for easy lookup (e.g., 'Transcript, starting: "Now, it's
      - crucial to remember that..."').
    - \* @Estimate Card Count:@ Provide a specific number of recommended Anki cards for
      - that topic.
    - \* @Outline Card Content:@ Briefly sketch out what each card or group of cards
      - should cover.
    - \* @Identify Visuals:@ If a slide contains a particularly important diagram, chart,
    - or visual model, note this in the blueprint.

Phase 2: Anki Card Generation (To Be Initiated by Me Later)

After you have delivered the Anki Blueprint, I will review it. I will then send a

- second, highly specific prompt that instructs you on the exact YAML format for
- generating the cards. You will then generate a maximum of 10 cards at a time.

Guiding Principles for Your Analysis:

- \* @Exam-Oriented Focus:@ Always prioritize content that is likely to be on an exam.
  - Definitions, enumerations (lists), comparisons, process steps, and foundational
  - concepts are more important than anecdotal stories.
- \* @Language Adaptation:@ You must automatically detect the primary language of the
  - lecture slides (German or English) and conduct your entire analysis and all
  - subsequent card generation in that language.
- \* @The "One Thing" Rule:@ Your card estimations should be guided by the principle that
  - a great flashcard tests exactly one piece of information.

To begin, please confirm you have understood these instructions by responding with:

- "Anki-Architect protocol initiated. Ready to receive lecture materials."@ Do not
- say anything else.

## ANHANG E: PROMPT FÜR PHASE 2: ANKI-BLUEPRINT ERSTELLUNG (MIT KONSOLIDIERUNG)

## # Objective:

Phase 2: Content Analysis & **Consolidated** Blueprint Generation. Your task is to analyze  
 ↳ the provided lecture materials and create a high-yield, efficient "Anki Blueprint."

## # Prime Directive: The Principle of Surgical Consolidation

Your absolute primary goal is to design a blueprint that results in a **maximum of 25 Anki**  
 ↳ **cards** for the entire lecture. This is a hard constraint.

This does **not** mean making every card simplistic. It means adopting a **surgical approach**.

- ↳ Your goal is to preserve high-detail, high-quality cards for core, examinable
- ↳ knowledge while efficiently grouping or summarizing less critical, supporting
- ↳ information.

To achieve this balance, you must follow this hierarchy:

1. **Protect the Core (Keep Detailed):** Prioritize detailed, atomic, and high-quality  
 ↳ cards for the most important information.
2. **Consolidate the Support (Group & Summarize):** Apply high-level synthesis and grouping  
 ↳ strategies primarily to secondary or illustrative information.

## # Core Philosophical Shift: Strategic Violation of the Atomic Principle

Normally, Anki cards should be "atomic". For this task, you **MUST** strategically violate  
 ↳ this principle.

## # Your Consolidation Strategies MUST Include:

1. **Grouping Lists & Enumerations:** Create **one single card** that asks for the entire list.
2. **Synthesizing Concepts:** Create **one card** that asks to "Compare and Contrast" them.
3. **Creating High-Level Questions:** For example: "What is the central argument of the  
 ↳ 'Transaction vs. Transformation Systems' slide?"
4. **Combining Definitions:** **Group several closely related terms on the back of a single**  
 ↳ **card**.

## ANHANG F: PROMPT FÜR PHASE 3: FINALE YAML-GENERIERUNG

## # Objective:

Phase 3: Anki Card Generation. The Anki Blueprint you generated is approved. You will  
 ↳ now generate the Anki cards as planned.

## # Command:

Your entire output must be a single YAML code block. Adherence to this protocol is  
 ↳ paramount, as a script will parse your output directly. Do not include any  
 ↳ conversational text.

## ### YAML Format Specification (Reminder)

- \* `'deck'` (String): The Anki deck name (`'Course Name::Lecture Number'`).
- \* `'model'` (Required, String): Choose either `'Basic'` or `'Basic (and reversed card)'`.
- \* `'cards'` (Required, List of Objects): A list of card data objects.

## ### CRITICAL Rules &amp; Requirements

## #### 1. UID Uniqueness

Every card object **MUST** have a unique `'uid'` in the format `'xxxx-xxxx-xxxx'`.

## #### 2. Formatting Requirements

You **MUST** use Markdown. Adherence to the following rules is essential.

- \* `@Lists:` A list `@MUST` be separated from any preceding text by a blank line.
- \* `@Tables:` Use `@GitHub Flavored Markdown (GFM) syntax`.
- \* `@Formulas (LaTeX):`
  - \* `@Inline Formulas:` Use `'\(...\).'`
  - \* `@Block Formulas:` Use `'\[\... \]'`
  - \* **@CRITICAL FORBIDDEN RULE:** `@NEVER` wrap LaTeX formulas in backticks (`'` ` '`).
  - \* **@CRITICAL FORBIDDEN SYNTAX:** You `@MUST NOT` use single dollar signs (`'$ ... $'`).

## ### Card Object Specification (Reminder)

- \* `'uid'` (Required, String): A `@unique` identifier.
- \* `'fields'` (Required, Dictionary): Keys must be `'Front'` and `'Back'`.
- \* `'tags'` (Required, List of Strings): Include relevant tags.

## ANHANG G: OPTIONALE FOLGE-PROMPTS ZUR STEUERUNG DES PROZESSES

## Listing 5. Einfacher Fortsetzungs-Prompt

Great, proceed with the next at most 10 cards from the blueprint. If there are none left to  
 ↳ generate, say so.

## Listing 6. Prompt zum Neustart des Prozesses mit einer neuen Vorlesung

Excellent. We will now restart the entire process for a new lecture.

You are now being reset to the Anki Blueprint Generation. Your task is to analyze the  
 ↳ brand-new lecture materials provided below and create an Anki Blueprint.

Prime Directive:

You are to follow exactly the same method and principles that were successfully used for the  
 ↳ previous lecture. You must maintain the established strategy regarding the level of  
 ↳ detail and the target number of cards.

New Lecture Materials:

- \* Lecture
- \* Slides
- \* Transcript

Final Instruction:

Please analyze the new materials and generate the Anki Blueprint based on the previously  
 ↳ established rules.

## ANHANG H: PYTHON-SKRIPT (PROCESS\_AND\_IMPORT.PY)

```

1 import yaml
2 import requests
3 from pathlib import Path
4 from datetime import datetime
5 import json
6 import markdown2
7 import re # Importiert die Bibliothek für reguläre Ausdrücke
8
9 # --- KONFIGURATION ---
10 ANKI_CONNECT_URL = "http://localhost:8765"
11 WORKING_FILE = Path("work_in_progress.yaml")
12 ARCHIVE_DIR = Path("archive")
13
14 # --- HELPER-FUNKTION ---
15 def anki_request(action, **params):
16     """Ein robuster Wrapper für Anfragen an AnkiConnect."""
17     payload = {'action': action, 'version': 6, 'params': params}
18     try:
19         response = requests.post(ANKI_CONNECT_URL, json=payload)
20         response.raise_for_status()
21         result = response.json()
22         if result.get('error'): return None, result['error']
23         return result.get('result'), None
24     except requests.exceptions.RequestException as e:
25         return None, f"Verbindung zu AnkiConnect fehlgeschlagen: {e}"
26
27 # --- HAUPTLOGIK DES SKRIPTS ---
28 def process_batch():
29     print("[INFO] Starte sichere Stapelverarbeitung...")
30     if not WORKING_FILE.exists() or WORKING_FILE.stat().st_size < 5:
31         print("[INFO] Arbeitsdatei ist leer. Vorgang wird beendet.")
32         return
33
34     # 1. YAML parsen
35     try:
36         with open(WORKING_FILE, 'r', encoding='utf-8') as f:
37             batch = yaml.safe_load(f)
38             if not batch or 'cards' not in batch:
39                 print("[FEHLER] YAML ist ungültig, 'cards' fehlt oder die Datei ist leer."); return
40     except yaml.YAMLError as e:
41         print(f"[FEHLER] Konnte {WORKING_FILE} nicht parsen. Grund: {e}"); return
42
43     # 2. Metadaten extrahieren & Konverter initialisieren

```

```

44 deck_name = batch.get('deck', 'Default')
45 default_model = batch.get('model', 'Basic')
46 global_tags = batch.get('tags', [])
47 # WICHTIG: Der Konverter hat jetzt die 'tables'-Erweiterung aktiviert
48 html_converter = markdown2.Markdown(extras=["fenced-code-blocks", "tables"])
49
50 # 3. Vorab-Prüfungen (Pre-flight Checks)
51 try:
52     model_name = batch.get('model', default_model)
53     available_models, error = anki_request('modelName')
54     if error: print(f"[FEHLER] Konnte Modellnamen nicht abrufen: {error}"); return
55     if model_name not in available_models:
56         print(f"\n[FATALER FEHLER] Der Notiztyp (Modell) '{model_name}' wurde nicht gefunden."); return
57
58     existing_decks, error = anki_request('deckNames')
59     if error: print(f"[FEHLER] Konnte Decknamen nicht abrufen: {error}"); return
60     if deck_name not in existing_decks:
61         print(f"[INFO] Deck '{deck_name}' existiert nicht. Es wird erstellt...")
62         _, error = anki_request('createDeck', deck=deck_name)
63         if error: print(f"[FEHLER] Deck '{deck_name}' konnte nicht erstellt werden: {error}"); return
64 except Exception as e:
65     print(f"[FEHLER] Unerwarteter Fehler bei den Vorab-Prüfungen: {e}"); return
66
67 notes_to_add, notes_to_update, skipped_count, collision_count = [], [], 0, 0
68
69 # 4. Karten verarbeiten
70 for card_data in batch.get('cards', []):
71     if not all(k in card_data for k in ['uid', 'fields']):
72         print(f"[WARNUNG] Überspringe Karte wegen fehlender 'uid' oder 'fields'. Daten: {card_data}");
73         continue
74
75     # --- START: Robuste Umwandlung von Feldern (mit korrigiertem Platzhalter) ---
76     processed_fields = {}
77     for key, value in card_data['fields'].items():
78         formulas = []
79
80         # Diese Hilfsfunktion findet eine Formel, speichert sie und gibt einen sicheren Platzhalter zur
81         # ück
82         def store_formula_and_get_placeholder(match):
83             formulas.append(match.group(0))
84             # KORRIGIERTER PLATZHALTER: Enthält keine Markdown-Sonderzeichen
85             return f"FORMELPLATZHALTERNUMMER{len(formulas)-1}"
86
87         # Konvertiere Wert zu String, um Fehler bei leeren Feldern zu vermeiden
88         text_with_placeholders = str(value or "")
89
90         # Finde und ersetze zuerst Block-Formeln ([...]), dann Inline-Formeln (...))
91         # re.DOTALL sorgt dafür, dass auch mehrzeilige Formeln korrekt gefunden werden
92         text_with_placeholders = re.sub(r'\[\[.*?\]\]', store_formula_and_get_placeholder,
93 text_with_placeholders, flags=re.DOTALL)
94         text_with_placeholders = re.sub(r'\(.*?\)', store_formula_and_get_placeholder,
95 text_with_placeholders, flags=re.DOTALL)
96
97         # Konvertiere den "sicheren" Markdown-Text (ohne Formeln) zu HTML
98         html_content = html_converter.convert(text_with_placeholders)
99
100        # Ersetze die Platzhalter im finalen HTML wieder durch die originalen, unversehrten Formeln
101        for i, formula in enumerate(formulas):
102            # KORRIGIERT: Ersetze den neuen, sicheren Platzhalter
103            placeholder = f"FORMELPLATZHALTERNUMMER{i}"
104            html_content = html_content.replace(placeholder, formula)
105
106        processed_fields[key] = html_content
107
108    card_data['fields'] = processed_fields
109    # --- ENDE: Robuste Umwandlung ---
110
111    uid_tag = f"uid:{card_data['uid']}"
112    query = f"tag:{uid_tag}"
113    existing_note_ids, error = anki_request('findNotes', query=query)
114    if error: print(f"[FEHLER] AnkiConnect-Abfrage für UID {card_data['uid']} fehlgeschlagen: {error}")
115    ; continue
116
117    if not existing_note_ids:
118        # Fall 1: Keine Notiz mit dieser UID vorhanden. Es ist eine neue Karte.
119        card_data['tags'] = sorted(list(set(global_tags + card_data.get('tags', [])) | {uid_tag}))
120        notes_to_add.append(card_data)

```

```

116 elif len(existing_note_ids) == 1:
117     # Fall 2: Genau eine Notiz gefunden. Prüfe, ob ein Update erlaubt und nötig ist.
118     note_id = existing_note_ids[0]
119
120     if card_data.get('overwrite') is True:
121         info_list, error = anki_request('notesInfo', notes=[note_id])
122         if error or not info_list:
123             print(f"[FEHLER] Konnte Info für Notiz-ID {note_id} nicht abrufen. Überspringe.");
124         continue
125
126         note_info = info_list[0]
127         if note_info['fields'] != card_data['fields']:
128             update_payload = {
129                 'id': note_id,
130                 'fields': card_data['fields'],
131                 'tags': sorted(list(set(global_tags + card_data.get('tags', [])) | {uid_tag}))
132             }
133             notes_to_update.append(update_payload)
134         else:
135             skipped_count += 1
136     else:
137         print(f"[WARNUNG] Karte mit UID '{card_data['uid']}' existiert bereits, aber 'overwrite:
138         true' ist nicht gesetzt. Update wird übersprungen.")
139         collision_count += 1
140     else:
141         # Fall 3: Mehrere Notizen mit derselben UID gefunden.
142         print(f"[WARNUNG] {len(existing_note_ids)} Notizen mit derselben UID '{card_data['uid']}'
143         gefunden. Überspringe zur Sicherheit.")
144         collision_count += 1
145
146 # 5. Stapelaktionen durchführen
147 if skipped_count > 0: print(f"[INFO] {skipped_count} Karten sind bereits aktuell.")
148 if collision_count > 0: print(f"[WARNUNG] {collision_count} Karten wurden wegen existierender UID ohne
149 'overwrite'-Flag oder anderer Anomalien übersprungen.")
150
151 if notes_to_add:
152     print(f"\n[INFO] Versuche, {len(notes_to_add)} neue Notizen hinzuzufügen...")
153     anki_notes_payload = [{'deckName': deck_name, 'modelName': n.get('model', default_model), 'fields':
154     n['fields'], 'tags': n['tags']} for n in notes_to_add]
155     results, error = anki_request('addNotes', notes=anki_notes_payload)
156     if error: print(f"[FEHLER] Die 'addNotes'-Aktion ist fehlgeschlagen: {error}")
157     else:
158         success_count = len(list(filter(None, results)))
159         if success_count > 0: print(f"[INFO] {success_count} / {len(notes_to_add)} Notizen erfolgreich
160         hinzugefügt.")
161
162 if notes_to_update:
163     print(f"[INFO] Aktualisiere {len(notes_to_update)} existierende Notizen...")
164     for note in notes_to_update:
165         anki_request('updateNoteFields', note={'id': note['id'], 'fields': note['fields']})
166         anki_request('setTags', notes=[note['id']], tags=" ".join(note['tags']))
167
168 # 6. Zusammenfassung und Archivierung
169 print("\n--- Zusammenfassung des Stapels ---")
170 added_count = len(list(filter(None, locals().get('results', [])))) if 'results' in locals() else 0
171 print(f"Hinzugefügt: {added_count} | Aktualisiert: {len(notes_to_update)} | Bereits aktuell: {
172 skipped_count} | Übersprungen/Kollisionen: {collision_count}")
173 print("-----")
174
175 timestamp = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
176 archive_filename = f"{timestamp}_{deck_name.replace(':', '-')}.yaml"
177 archive_path = ARCHIVE_DIR / archive_filename
178 ARCHIVE_DIR.mkdir(exist_ok=True)
179 WORKING_FILE.rename(archive_path)
180 print(f"[INFO] Arbeitsdatei archiviert nach: {archive_path}")
181 WORKING_FILE.touch()
182 print("[ERFOLG] Workflow abgeschlossen.")
183
184 if __name__ == "__main__":
185     _, err = anki_request('deckNames')
186     if err: print(f"[FEHLER] AnkiConnect nicht erreichbar. Bitte stellen Sie sicher, dass Anki läuft und
187     das Add-on installiert ist.")
188     else: process_batch()

```

Listing 7. Vollständiges Python-Skript zum Parsen der YAML-Datei und zum Importieren der Karten in Anki via AnkiConnect.