

# **Markdown als Schreib- und Produktionsumgebung für wissenschaftliche Arbeiten**

*Eine kriteriologische Bewertung von mdedit.io im Vergleich zu Word und LaTeX*

**Max Beispiel**

Masterarbeit · Universität Beispiel · 2026-05-10

## Abstract

Die Erstellung wissenschaftlicher Arbeiten bewegt sich heute zwischen drei dominanten Logiken: der textverarbeitungsorientierten Logik von Microsoft Word, der satzorientierten Logik von LaTeX und der strukturierten Logik markdown-basierter Workflows. Diese Arbeit untersucht, unter welchen Bedingungen ein browserbasierter Markdown-Editor am Beispiel von mdedit.io als primäre Produktionsumgebung für wissenschaftliche Abschlussarbeiten dienen kann. Ausgangspunkt ist die Beobachtung, dass die Eignung von Schreibwerkzeugen in der akademischen Praxis häufig über Gewohnheiten, Fachkulturen oder Tool-Mythen beurteilt wird, während ein explizites Bewertungsmodell für Strukturtransparenz, Quellenarbeit, Exportstabilität und Betreuungskompatibilität oft fehlt.

Methodisch folgt die Arbeit einer kriteriengeleiteten Artefaktanalyse mit designorientierter Bewertungslogik. Auf Basis der Literatur zum wissenschaftlichen Arbeiten, zu Markdown und Pandoc sowie zu designwissenschaftlichen Bewertungsansätzen werden fünf Bewertungsdimensionen abgeleitet: Strukturtransparenz, Quellen- und Zitationssicherheit, Layout- und Exportstabilität, Review- und Betreuungskompatibilität sowie Governance und Betriebsfähigkeit ([Booth u. a. 2016](#); [Swales und Feak 2012](#); [Hevner u. a. 2004](#); [Wieringa 2014](#)). Diese Dimensionen werden zunächst auf Word, LaTeX und markdown-basierte Workflows bezogen und anschließend für mdedit.io anhand offizieller Dokumentation, institutioneller Vorgaben und eines reproduzierbaren Referenzartefakts bewertet.

Die Ergebnisse zeigen, dass Markdown in Verbindung mit Spezifikation, Metadaten und Konvertern ein belastbares Zwischenformat für wissenschaftliche Langtexte sein kann ([Gruber 2004](#); [CommonMark Contributors 2026](#); [MacFarlane 2025](#)). Zugleich gewinnt Markdown im Kontext generativer KI eine besondere Rolle als strukturstabiles Arbeits- und Austauschformat: Überschriften, Listen, Tabellen, Codeblöcke und Zitationsmarker bleiben für Menschen lesbar und für Modelle gezielt bearbeitbar. Für mdedit.io ergibt sich dadurch eine starke Eignung in der Struktur- und Schreibphase, in der Navigation großer Dokumente über Outline und Tree View, in der KI-gestützten Dokumentarbeit, in der frühen Layoutkontrolle sowie in kontrollierten PDF- und DOCX-Exportpfaden ([Hertel 2026c, 2026b](#)). Zusätzlich ist relevant, dass der wissenschaftliche Exportpfad aktuelle mdedit-Erweiterungen für Bibliografie, Citeproc und Referenzsektionen integriert ([Hertel 2026d](#)). Grenzen bleiben bei note-style-Fußnoten, stark normierten Sondervorlagen und semantisch reichen Referenzsystemen. Die Arbeit leistet damit keinen empirischen Nutzernachweis, wohl aber ein kriteriologisches Entscheidungsmodell und ein prüfbares Referenzartefakt für die Bewertung thesistauglicher Markdown-Workflows.

## Schlagwörter

Markdown, wissenschaftliche Abschlussarbeit, mdedit.io, Design Science, LaTeX, Microsoft Word, Pandoc, KI-Anwendungen, Schreibwerkzeuge, Exportworkflow

Abbildungsverzeichnis .....	4
Tabellenverzeichnis .....	4
1. Einleitung .....	4
1.1 Forschungsfrage .....	5
1.2 Arbeitsthesen .....	5
1.3 Methodisches Vorgehen .....	6
1.4 Aufbau der Arbeit .....	6
2. Markdown: Begriff, Ursprung und Entwicklung .....	6
2.1 Was Markdown ist und warum es entstand .....	6
2.2 Warum Markdown nicht bei seinem Ursprung stehenblieb .....	7
2.3 Markdown im erweiterten Ökosystem: Pandoc als Brücke .....	7
2.4 Markdown als Arbeitsformat in KI-Applikationen .....	7
3. Optionen für die Erstellung wissenschaftlicher Arbeiten .....	8
3.1 Besondere Anforderungen wissenschaftlicher Dokumente .....	8
3.2 Microsoft Word .....	9
3.3 LaTeX .....	9
3.4 Markdown-basierte Workflows .....	10
3.5 Vergleich von Word, LaTeX und Markdown .....	10
4. mdedit.io als Untersuchungsfall .....	11
4.1 Markdown als wissenschaftliches Zwischenformat .....	11
4.2 Funktionsprofil des Untersuchungsfalls .....	11
4.3 Vorläufige Einordnung im Bewertungsraster .....	11
4.4 Absehbare Grenzen des Untersuchungsfalls .....	12
5. Kriteriengeleitete Bewertung von mdedit.io .....	12
5.1 Bewertungslogik und Untersuchungsartefakt .....	12
5.2 Strukturtransparenz und Schreibprozess .....	13
5.3 Quellenapparat und wissenschaftliche Integrität .....	14
5.4 Layout-, Satz- und Exportstabilität .....	15
5.5 Review-, Betreuungs- und Kollaborationsfähigkeit .....	16
5.6 Ausschlusskriterien und Grenzfälle .....	16
5.7 Belastungsprobe an anspruchsvollen Hochschulprofilen .....	17
5.8 Gestaltungsimplicationen für institutionelle Profile .....	18

5.9 Gründe gegen mdedit.io als Werkzeug der Wahl für die eigene Masterthesis .....	18
6. Governance, Rechte und Betriebsfähigkeit .....	19
6.1 Lizenzierung und Rechte .....	19
6.2 Nutzungszugang und praktische Nutzbarkeit .....	20
6.3 Verfügbarkeit und Betrieb .....	20
7. Diskussion .....	20
8. Fazit .....	20
Literaturverzeichnis .....	21
Anhang A. Reproduzierbares Referenzartefakt zur Layout- und Funktionsprüfung .....	23
A.1 Abbildungsprobe .....	23
A.2 Wissenschaftliche Tabelle .....	23
A.3 Kompakte Tabelle .....	23
A.4 Mathematik- und Fußnotenprobe .....	24
A.5 Mehrspaltenprobe .....	24
A.6 Unicode-, Querverweis- und Listingprobe .....	24

## Abbildungsverzeichnis

Abb. 1: Neutrales Bildartefakt für Caption-, Listen- und Printtests .....	23
---	----

## Tabellenverzeichnis

Tab. 1: Häufig genutzte KI-unterstützte Chats und ihre typischen Arbeitsformate. ....	8
Tab. 2: Vergleich zentraler Werkzeuglogiken für wissenschaftliche Arbeiten. ....	10
Tab. 3: Bewertungsdimensionen der Analyse. ....	12
Tab. 4: Zusammenfassende Bewertung von mdedit.io im Anforderungsraster. ....	16
Tab. 5: Kriteriensatz für die Endprüfung einer wissenschaftlichen Arbeit in mdedit.io. ....	23
Tab. 6: Typische Aufgaben vor der Abgabe. ....	23

## 1. Einleitung

Wissenschaftliche Arbeiten sind heute nicht mehr nur Textprodukte, sondern organisierte Schreib- und Überarbeitungsprozesse. Eine Masterthesis entsteht über Wochen oder Monate hinweg in einem Wechselspiel aus Materialsammlung, Strukturierung, Rohfassung, Rückmeldung, sprachlicher Revision, formaler Kontrolle und finalem Export. Die Qualität des eingesetzten Werkzeugs entscheidet dabei nicht über die wissenschaftliche Güte der Argumentation, wohl aber über Reibungsverluste, Sichtbarkeit der Struktur und

die Stabilität des Produktionsprozesses. Genau deshalb ist die Frage nach geeigneten Schreibumgebungen kein Nebenaspekt, sondern Teil akademischer Arbeitsorganisation.

In der Praxis dominieren weiterhin Microsoft Word und LaTeX. Word ist institutionell weit verbreitet, für Reviewprozesse eingespielt und durch Vorlagenkulturen fest in Hochschulabläufe eingebunden. LaTeX wiederum ist in mathematik-, informatik- und naturwissenschaftsnahen Milieus der Referenzpunkt für typografisch anspruchsvolle, formal stark geregelte Dokumente ([LaTeX Project 2026](#)). Markdown-basierte Workflows werden dagegen häufig noch als informell, technisch oder blognah wahrgenommen. Diese Einschätzung greift jedoch zu kurz, weil moderne Markdown-Ökosysteme durch Spezifikationen wie CommonMark, Konverter wie Pandoc und darauf aufbauende Editoren deutlich mehr leisten als reine Notiz- oder Web-Authoring-Szenarien ([CommonMark Contributors 2026](#); [MacFarlane 2025](#)).

Zusätzlich verändert der Aufstieg generativer KI die Bewertung von Markdown noch einmal. Viele KI-Applikationen arbeiten nicht auf der Ebene sichtbarer Seitenformatierung, sondern mit linearem, segmentierbarem Text. Genau hier ist Markdown stark: Überschriften, Listen, Tabellen, Codeblöcke und Metadaten bleiben im Chat-Kontext explizit, kopierbar und transformierbar. Für wissenschaftliche Schreibumgebungen ist das relevant, weil KI-Unterstützung dann nicht an Screenshots oder lokaler Sichtformatierung ansetzt, sondern am dokumentierten Strukturkern.

Vor diesem Hintergrund untersucht die vorliegende Arbeit, welche Rolle Markdown und speziell mdedit.io für wissenschaftliche Arbeiten spielen können. Im Zentrum steht nicht die Behauptung, ein einzelnes Werkzeug könne alle konkurrierenden Produktionslogiken ersetzen. Gefragt wird vielmehr, für welche Anforderungsprofile wissenschaftlicher Abschlussarbeiten ein browserbasierter Markdown-Workflow methodisch vertretbar, technisch stabil und institutionell anschlussfähig ist. Zusätzlich führt die Arbeit ein reproduzierbares Referenzartefakt mit, das die für wissenschaftliche Langtexte relevanten Satz- und Exportelemente praktisch prüfbar macht.

## 1.1 Forschungsfrage

Die leitende Forschungsfrage lautet: Unter welchen Bedingungen kann ein browserbasierter Markdown-Editor am Beispiel von mdedit.io als primäre Produktionsumgebung für wissenschaftliche Abschlussarbeiten dienen, und unter welchen formalen Randbedingungen bleiben Word oder LaTeX vorzugswürdig?

Zur Beantwortung werden drei Teilfragen verfolgt:

1. Welche Anforderungen sind für wissenschaftliche Abschlussarbeiten konstitutiv und damit für die Werkzeugwahl entscheidend?
2. Welche Produktionslogiken unterscheiden Word, LaTeX und markdown-basierte Workflows in Bezug auf diese Anforderungen?
3. An welcher Stelle erweist sich mdedit.io als tragfähiger Primärworkflow, und wo beginnen Ausschlusskriterien oder institutionelle Sonderfälle?

## 1.2 Arbeitsthesen

Die Arbeit basiert auf vier Arbeitsthesen:

1. Markdown ist nicht mehr nur eine einfache Web-Syntax, sondern ein ernstzunehmendes Zwischenformat für strukturierte Langtexte, sofern Spezifikation, Metadaten und Exportwerkzeuge zusammenwirken. Dieselbe Eigenschaft macht es zugleich zu einem besonders anschlussfähigen

Arbeitsformat für KI-gestützte Redaktions- und Transformationsprozesse ([Gruber 2004](#); [CommonMark Contributors 2026](#); [MacFarlane 2025](#)).

2. Die Eignung eines Werkzeugs für Masterarbeiten entscheidet sich weniger an isolierter Typografie als an Strukturtransparenz, Quellenintegrität, Reviewfähigkeit und reproduzierbaren Exportpfaden.
3. mdedit.io ist besonders dort stark, wo Longform-Struktur, frühe Layoutkontrolle, kontrollierter Multi-Export, dokumentnahe KI-Unterstützung und browserbasierte Zusammenarbeit im Vordergrund stehen ([Hertel 2026c](#)).
4. mdedit.io bleibt dort begrenzt, wo semantisch reiche Referenzsysteme, note-style-Fußnoten oder starre Hochschulvorlagen produktiv zwingend sind ([Hertel 2026e](#), [2026f](#)).

### 1.3 Methodisches Vorgehen

Methodisch folgt die Arbeit einer kriteriengeleiteten Artefaktanalyse im Sinne designorientierter Forschung ([Hevner u. a. 2004](#); [Wieringa 2014](#)). Im ersten Schritt wird aus der Literatur zum wissenschaftlichen Arbeiten und aus institutionellen Zitiervorgaben ein Anforderungskatalog für wissenschaftliche Abschlussarbeiten rekonstruiert ([Booth u. a. 2016](#); [Swales und Feak 2012](#); [University of Reading Library 2026](#)). Im zweiten Schritt werden die drei dominanten Werkzeuglogiken Word, LaTeX und Markdown vergleichend auf diese Anforderungen bezogen. Im dritten Schritt erfolgt für mdedit.io eine vertiefte Fallanalyse anhand offizieller Produktdokumentation, wissenschaftsnaher Projektkonzepte und des in dieser Arbeit mitgeführten Referenzartefakts ([Hertel 2026c](#), [2026e](#), [2026f](#)).

Die Bewertung erfolgt entlang fünf Dimensionen: Strukturtransparenz, Quellen- und Zitationssicherheit, Layout- und Exportstabilität, Review- und Betreuungskompatibilität sowie Governance und Betriebsfähigkeit. Ein Werkzeug gilt in dieser Logik dann als thesis-tauglich, wenn es die Produktionskette einer Abschlussarbeit in diesen Dimensionen ohne methodisch kritische Brüche tragen kann. Die Arbeit beansprucht keine Verallgemeinerung aus Nutzerzahlen oder Experimentdaten, sondern eine analytisch nachvollziehbare Eignungsprüfung.

### 1.4 Aufbau der Arbeit

Kapitel 2 rekonstruiert Ursprung und Entwicklung von Markdown als wissenschaftlich relevantes Zwischenformat und ergänzt diese Perspektive um seine besondere Rolle in KI-Applikationen. Kapitel 3 leitet aus der Literatur und aus institutionellen Vorgaben das Bewertungsraster für wissenschaftliche Schreibwerkzeuge ab und positioniert Word, LaTeX und Markdown vergleichend. Kapitel 4 bestimmt mdedit.io als Untersuchungsfall. Kapitel 5 führt die kriteriologische Bewertung durch und spiegelt den Befund an drei anspruchsvollen Hochschulprofilen. Kapitel 6 betrachtet Governance-, Rechts- und Betriebsaspekte. Kapitel 7 diskutiert Beitrag und Grenzen der Untersuchung, bevor Kapitel 8 die Forschungsfrage beantwortet. Der Anhang fungiert als reproduzierbares Referenzartefakt für layout- und exportkritische Dokumentelemente.

## 2. Markdown: Begriff, Ursprung und Entwicklung

### 2.1 Was Markdown ist und warum es entstand

Markdown wurde 2004 von John Gruber als Text-zu-HTML-Werkzeug eingeführt. Der Kern des Ansatzes besteht darin, dass ein Dokument bereits als Rohtext gut lesbar sein soll und erst im zweiten Schritt in strukturell valides HTML umgewandelt wird ([Gruber 2004](#)). Diese doppelte Logik ist bis heute entscheidend: Markdown ist zugleich Syntax und Konvertierungsidee. Sein ursprünglicher Zweck war nicht wissenschaftliches Publizieren, sondern lesbares Schreiben für das Web. Gerade darin liegt jedoch eine

bleibende Stärke. Markdown trennt Inhalt, Struktur und spätere Darstellung stärker voneinander als klassische WYSIWYG-Textverarbeitung.

Die normative Pointe liegt im Gestaltungsziel der Lesbarkeit. Während HTML-Tags den Rohtext visuell überfrachten können, sollte ein Markdown-Dokument auch ohne Rendering als strukturierter Text begreifbar bleiben (Gruber 2004). Diese Eigenschaft macht Markdown für wissenschaftliche Prozesse interessant, weil akademisches Schreiben gerade in frühen Phasen weniger von perfekter Formatierung als von orientierbarer Gliederung lebt.

## 2.2 Warum Markdown nicht bei seinem Ursprung stehen blieb

Die Offenheit der frühen Markdown-Beschreibung war zugleich Erfolg und Problem. Weil die Syntax ursprünglich nicht unmissverständlich spezifiziert war, entwickelten sich zahlreiche Implementierungen mit voneinander abweichendem Verhalten. CommonMark beschreibt diese historische Lage explizit als Problem divergierender Parser und fehlender Testsuiten (CommonMark Contributors 2026). Für wissenschaftliche Dokumente ist genau diese Divergenz kritisch, weil Reproduzierbarkeit und Portabilität zentrale Anforderungen sind.

Mit CommonMark wurde Markdown deshalb in Richtung einer formaleren, testbaren und interoperablen Spezifikation weiterentwickelt (CommonMark Contributors 2026). Die Entwicklung von Markdown lässt sich damit als Verschiebung von einer pragmatischen Web-Schreibpraxis zu einem standardisierbaren Format für strukturierte Dokumente lesen. Dieser Schritt ist für wissenschaftliches Arbeiten zentral, weil er die Voraussetzung für verlässliche Konverter und stabile Renderpfade schafft.

## 2.3 Markdown im erweiterten Ökosystem: Pandoc als Brücke

Pandoc markiert die zweite entscheidende Entwicklungsstufe. Während Grubers Markdown vor allem auf HTML zielte, macht Pandoc aus Markdown ein universelles Zwischenformat. Nach eigener Beschreibung konvertiert Pandoc zwischen zahlreichen Markup-, Dokument- und Office-Formaten, darunter Markdown, DOCX, LaTeX und verschiedene PDF-Pfade (MacFarlane 2025). Zugleich unterstützt Pandoc Metadaten, Tabellen, Fußnoten, Mathematik sowie automatische Zitationen und Bibliografien (MacFarlane 2025).

Für wissenschaftliche Arbeiten ist das von erheblicher Bedeutung. Erst die Verbindung aus spezifizierter Syntax, Metadaten und Konverter ermöglicht einen Workflow, in dem Rohtext, wissenschaftliche Struktur und Ausgabemedien systematisch miteinander gekoppelt werden. Damit wird Markdown nicht automatisch zum besten Werkzeug für jede Thesis. Es wird aber zu einer belastbaren Alternative für Autorinnen und Autoren, die Struktur und Portabilität höher gewichten als permanente Sichtformatierung.

## 2.4 Markdown als Arbeitsformat in KI-Applikationen

Generative KI-Anwendungen operieren im Kern nicht mit Seitengeometrie, sondern mit tokenisiertem Text, semantischen Segmenten und expliziten Strukturmarkern. Genau deshalb hat Markdown in KI-Applikationen eine besondere Rolle: Überschriften, Listen, Tabellen, Zitatblöcke, Code-Fences und YAML-artige Metadaten bleiben für Menschen lesbar und für Modelle zugleich vergleichsweise gut adressierbar. Während visuelle Office-Formatierung oft implizit oder binär gebunden ist, stellt Markdown Struktur in einer linearen, kopierbaren und promptfähigen Form bereit.

Für KI-gestützte Schreibumgebungen ist das mehr als eine Komfortfrage. Ein Modell kann auf einem Markdown-Dokument nicht nur stilistisch antworten, sondern Abschnitte gezielt umstellen, Tabellen ergänzen, Listen konsolidieren, Frontmatter anpassen oder Zitationsmarker unverändert erhalten. Markdown

fungiert damit zugleich als Darstellungsformat, Transformationsformat und Austauschformat zwischen Autorenschaft, Chat-Oberfläche, Agentenlogik und nachgelagerten Konvertern.

Tab. 1: Häufig genutzte KI-unterstützte Chats und ihre typischen Arbeitsformate.

System	Primäre Interaktion	Typische Arbeitsformate	Rolle von Markdown
ChatGPT	Dialogischer Chat mit Datei- und Bildkontext	Freitext, Markdown, PDF, Office-Dateien, Tabellen, Bilder, Code	Markdown eignet sich als robustes Zwischenformat für strukturierte Prompts, Listen, Tabellen und formatstabile Antworten
Claude	Dokumentenzentrierter Chat mit langem Kontext	Freitext, Markdown, PDF, DOCX, Bilder, Code	Markdown unterstützt lange Abschnitte, Revisionen und klare Abschnittsreferenzen
Gemini	Such- und workspace-naher Chat	Freitext, Markdown, PDF, Workspace- und Office-Dokumente, Tabellen, Bilder	Markdown erleichtert den Wechsel zwischen Chat, Dokument und Export ohne versteckte Layoutlogik
Perplexity	Suchorientierter Recherche-Chat	Freitext, Markdown, Webquellen, PDF, Bilder	Markdown ist nützlich für strukturierte Synthesen, Listen und zitiernahe Notizen
mdeedit.io KI-Chat	Dokumentgebundener Editor-Chat	Markdown-Dokumente, Frontmatter, Zitationsmarker, Layout-Kommandos	Markdown ist hier Primärformat des Arbeitsdokuments und nicht nur ein Nebenformat

Gerade im wissenschaftlichen Kontext ist dieser Unterschied relevant. Wer mit KI auf einem Markdown-Kern arbeitet, bearbeitet nicht lediglich Textoberflächen, sondern explizite Dokumentstruktur. Dadurch sinkt das Risiko, dass Revisionen semantische Gliederung, Zitationsmarker oder Exportpfade unbeabsichtigt zerstören. Für wissenschaftliche Langtexte ist Markdown deshalb nicht nur ein schlankes Schreibformat, sondern zunehmend auch das geeignetere Interfaceformat für KI-gestützte Redaktionsprozesse.

### 3. Optionen für die Erstellung wissenschaftlicher Arbeiten

#### 3.1 Besondere Anforderungen wissenschaftlicher Dokumente

Wissenschaftliche Arbeiten unterscheiden sich von Alltagsdokumenten durch eine Reihe spezifischer Anforderungen. Dazu gehören mindestens eine stabile Kapitelhierarchie, konsistente Zitation, sauber geführte Literaturverzeichnisse, reproduzierbare Tabellen- und Abbildungsintegration, kontrollierbare Exportpfade,

Rückmeldeschleifen mit Betreuenden und formale Anschlussfähigkeit an Hochschulvorgaben. Hinzu kommen praktische Anforderungen wie Versionssicherheit, Reviewfähigkeit, Datensparsamkeit und langfristige Lesbarkeit. In der einschlägigen Literatur zum wissenschaftlichen Arbeiten wird diese Verbindung aus Erkenntnislogik, Strukturierung, Zitierdisziplin und redaktioneller Konsistenz seit langem als Kern wissenschaftlicher Textproduktion beschrieben ([Booth u. a. 2016](#); [Swales und Feak 2012](#)).

Diese Anforderungen machen deutlich, dass kein Werkzeug allein aufgrund schöner Typografie oder schneller Bedienbarkeit ausreicht. Entscheidend ist vielmehr, wie gut ein Werkzeug das Wechselspiel von Schreiben, Strukturieren, Kommentieren, Exportieren und formaler Kontrolle organisiert. Hinzu kommt, dass Zitier- und Referenzsysteme disziplinär variieren: APA, Chicago, MHRA, OSCOLA und Vancouver sind keine äquivalenten Geschmackssachen, sondern fachkulturell unterschiedliche Regime mit jeweils eigenen Erwartungen an In-Text-Zitate, Fußnoten, Literaturverzeichnis und Quellenapparat ([University of Reading Library 2026](#); [American Psychological Association 2019](#); [Paver, Nelson, und Davies 2024](#); [Faculty of Law, University of Oxford 2026](#); [Imperial College London Library Services 2026](#)).

Aus diesen Anforderungen leitet die Arbeit ein fünfdimensionales Bewertungsraster ab: (1) Strukturtransparenz, (2) Quellen- und Zitationssicherheit, (3) Layout- und Exportstabilität, (4) Review- und Betreuungskompatibilität sowie (5) Governance und Betriebsfähigkeit. Dieses Raster dient im weiteren Verlauf als Vergleichs- und Bewertungsinstrument für die drei Werkzeuglogiken und für den Fall mdedit.io.

### 3.2 Microsoft Word

Microsoft Word ist an Hochschulen weithin etabliert. Ein wesentlicher Vorteil liegt in seiner institutionellen Anschlussfähigkeit: Viele Betreuende arbeiten mit Word-Kommentaren, Vorlagen und Korrekturmodi. Besonders der Reviewbereich mit Track Changes, verschiedenen Darstellungsmodi, reviewerbezogener Filterung und Reviewing Pane ist für klassische Betreuungs- und Korrekturprozesse stark ausgebaut ([Microsoft Support 2026](#)).

Seine Schwäche liegt jedoch oft in der engen Kopplung von Inhalt und unmittelbarer Formatierung. Zwar bietet Word Formatvorlagen und professionelle Layoutfunktionen, in der Praxis werden lange Dokumente jedoch häufig durch lokale Formatkorrekturen, inkonsistente Styles und schwer nachvollziehbare Umbrüche instabil. Gerade bei stark überarbeiteten Masterarbeiten führt das nicht selten zu Formatdrift, also zu einem schleichenden Verlust an struktureller Disziplin.

### 3.3 LaTeX

LaTeX versteht sich selbst nicht als Textverarbeitung, sondern als Dokumentvorbereitungssystem für hochwertige Typografie, insbesondere in wissenschaftlichen und technischen Kontexten ([LaTeX Project 2026](#)). Das System ist für große Dokumente, Querverweise, Tabellen, Formeln, Bibliografien und Register ausgelegt und entlastet Autorinnen und Autoren von vielen direkten Designentscheidungen ([LaTeX Project 2026](#)).

Genau darin liegt die klassische Stärke von LaTeX: saubere Trennung von Inhalt und Satz, starke mathematische und typografische Kompetenz sowie hohe Stabilität bei großen Dokumenten. Die Kehrseite ist die höhere Einstiegshürde. Wer LaTeX nicht bereits beherrscht, investiert signifikant Zeit in Toolchain, Fehlersuche, Paketlogik und Vorlagensysteme. Für viele Studierende ist LaTeX deshalb eher dann attraktiv, wenn Fachkultur, Vorwissen oder formale Komplexität diese Investition rechtfertigen.

### 3.4 Markdown-basierte Workflows

Markdown-basierte Workflows besetzen die Zwischenzone zwischen Word und LaTeX. Sie bieten weniger unmittelbare Layoutsteuerung als Word und weniger typografische Tiefenkontrolle als LaTeX, können aber Struktur, Lesbarkeit im Rohtext und Portabilität besonders gut verbinden. In Verbindung mit Pandoc kommen Metadaten, Zitationen, Bibliografien sowie DOCX- und PDF-Exporte hinzu ([MacFarlane 2025](#)).

Markdown ist damit kein Ersatz für jede Fachkultur und jede Hochschulvorgabe. Seine besondere Stärke liegt dort, wo das Schreiben selbst, die Sicht auf Struktur und ein kontrollierter Multi-Export wichtiger sind als pixelgenaue Formatarbeit während der Rohfassung. Im Kontext KI-gestützter Schreibearbeit kommt hinzu, dass Markdown seine Struktur explizit mitführt und dadurch besser für Transformation, Revision und Weiterverarbeitung geeignet ist als rein visuell orientierte Formate.

### 3.5 Vergleich von Word, LaTeX und Markdown

Die Werkzeugklassen unterscheiden sich weniger entlang einer simplen Rangordnung als entlang ihrer Produktionslogik. Word ist review- und institutionenfreundlich, LaTeX ist satz- und formalstark, Markdown ist struktur- und portabilitätsorientiert.

Tab. 2: Vergleich zentraler Werkzeuglogiken für wissenschaftliche Arbeiten.

Kriterium	Word	LaTeX	Markdown-basierter Workflow
Einstieg	Niedrig bis mittel	Mittel bis hoch	Niedrig bis mittel
Review mit Betreuenden	Sehr stark	Oft indirekt oder PDF-basiert	Stark, wenn Editor Sharing und Kommentare unterstützt
Typografische Kontrolle	Mittel bis hoch	Sehr hoch	Mittel
Rohtext-Lesbarkeit	Niedrig	Mittel	Hoch
Portabilität	Mittel	Hoch	Hoch
Zitationsreife	Hoch	Sehr hoch	Hoch mit Pandoc-Citeproc
Gefahr lokaler Formatdrift	Hoch	Niedrig	Niedrig
Eignung für spontane Überarbeitung	Hoch	Mittel	Hoch

Aus dieser Gegenüberstellung folgt kein pauschaler Sieger. Vielmehr hängt die Eignung vom Profil der Arbeit ab. Wo eine Hochschule starre Word-Vorlagen erzwingt, bleibt Word faktisch dominant. Wo mathematische Satzqualität und komplexe Referenzsysteme im Zentrum stehen, bleibt LaTeX stark. Wo jedoch Strukturtransparenz, editorische Leichtigkeit, Portabilität, Web-Review und frühe Exportkontrolle im Vordergrund stehen, gewinnt Markdown erheblich an Attraktivität.

## 4. mdedit.io als Untersuchungsfall

### 4.1 Markdown als wissenschaftliches Zwischenformat

Markdown verbindet drei Vorteile, die für wissenschaftliche Langtexte besonders relevant sind. Erstens erzwingt es eine strukturorientierte Schreibweise. Zweitens bleibt der Rohtext dauerhaft lesbar und portabel. Drittens lässt es sich über Konverter wie Pandoc in etablierte Zielformate überführen ([Gruber 2004](#); [MacFarlane 2025](#)). Genau diese Eigenschaften reduzieren die Gefahr, dass eine Arbeit in lokalen Formatkorrekturen versinkt, bevor die Argumentation stabil ist.

Markdown ist besonders dann plausibel, wenn Schreiben als Denken in Strukturen verstanden wird. Die Frage lautet dann nicht mehr, welche Schrift die Zwischenüberschrift aktuell hat, sondern ob der Abschnitt überhaupt eine klare Funktion im Argumentationsgang besitzt. Das ist für wissenschaftliche Texte keine Nebensache, sondern oft der Unterschied zwischen editorischer Aktivität und wissenschaftlicher Stringenz. Dieselbe Strukturorientierung erklärt zugleich seine wachsende Bedeutung in KI-Applikationen: Wenn Modelle an expliziten Textsegmenten statt an versteckter Layoutsemantik arbeiten, wird Markdown zum effizienteren Interfaceformat.

### 4.2 Funktionsprofil des Untersuchungsfalls

Nach der Projektdokumentation ist mdedit.io ein browserbasierter Markdown-Editor für Longform-Schreiben, strukturierte Dokumente und exportierbare Entwürfe. Die Anwendung kombiniert Live-Preview, dokumentbasierte Outline-Navigation, Kollaboration, KI-Assistenz und Export nach Markdown, DOCX und PDF in einer selbst gehosteten Webanwendung ([Hertel 2026c](#)). Die Hilfeoberfläche beschreibt den rechten Arbeitsbereich explizit als Kombination aus Vorschau, Baumansicht und KI-Chat; außerdem nennt sie einen Layout-Editor sowie drucknahe Paged-Preview als eigene Arbeitsmodi ([Hertel 2026b](#)). Für die vorliegende Untersuchung ist daran vor allem relevant, dass mdedit.io nicht nur Textbearbeitung, sondern die Kopplung von Schreibprozess, Dokumentstruktur und Ausgabeformat adressiert.

Der Fall eignet sich deshalb als Untersuchungsobjekt, weil hier mehrere für Abschlussarbeiten zentrale Anforderungen gleichzeitig berührt werden: Sichtbarkeit der Kapitelstruktur, frühe Vorschau des Satzbilds, browserbasierter Zugang ohne lokale Desktop-Installation und die Möglichkeit, einen Markdown-Kern in unterschiedliche Abgabe- und Reviewformate zu überführen ([Hertel 2026c](#)). Hinzu kommt ein weiterer Punkt: Der KI-Chat arbeitet nicht an einem fremden, binär formatierten Dokument, sondern am expliziten Markdown-Quelltext. Damit wird mdedit.io zu einem Fall, in dem Markdown nicht nur Exportbasis, sondern unmittelbares Bearbeitungsmedium für menschliche und KI-gestützte Redaktion ist.

### 4.3 Vorläufige Einordnung im Bewertungsraster

Aus theoretischer Sicht ist mdedit.io vor allem dort relevant, wo wissenschaftliche Arbeit nicht primär als Layoutproduktion, sondern als strukturierte Dokumententwicklung verstanden wird. Die Outline reduziert den Abstand zwischen Gliederung und Rohtext, die Vorschau verkürzt den Weg zur drucknahen Kontrolle, und der Multi-Export macht denselben Dokumentkern für unterschiedliche institutionelle Situationen verwendbar ([Hertel 2026c](#)).

Damit positioniert sich mdedit.io zwischen der review- und institutionsnahen Logik von Word und der satz- und formalstärkeren Logik von LaTeX. Diese Zwischenposition ist gerade für Abschlussarbeiten interessant, weil viele Projekte weder reine Office-Routine noch voll entwickelte TeX-Workflows voraussetzen, wohl aber eine stabile, nachvollziehbare Produktionsumgebung benötigen.

## 4.4 Absehbare Grenzen des Untersuchungsfalls

Die eigene wissenschaftliche Konzeptdokumentation von mdedit.io benennt zugleich klare Grenzen. Der Hauptengpass liegt weniger im Seitenlayout als in semantischen Dokumentfunktionen wie YAML-Frontmatter, Labels, Nummerierung, Cross-References und belastbaren Zitationspfaden (Hertel 2026e). Ebenfalls werden echte seitengebundene Fußnoten für note-style-Zitierstile nicht als gesichert im aktuellen HTML- und Paged-Stack versprochen (Hertel 2026e).

Die sachliche Schlussfolgerung lautet daher: mdedit.io ist für wissenschaftliche Langtexte ein ernstzunehmender Untersuchungsfall, aber kein universell vollständiges System. Seine Eignung steigt dort, wo Autorinnen und Autoren bewusst mit Struktur, Frontmatter, Citeproc-Export und Layoutregeln arbeiten. Sie sinkt dort, wo absolute Kompatibilität mit sehr spezifischen Vorlagensystemen oder LaTeX-zentrierten Fußnotenstilen gefordert ist.

## 5. Kriteriengeleitete Bewertung von mdedit.io

### 5.1 Bewertungslogik und Untersuchungsartefakt

Die Bewertung folgt nicht der Frage, ob mdedit.io jedes Einzelmerkmal von Word oder LaTeX repliziert. Entscheidend ist vielmehr, ob das Werkzeug die wissenschaftliche Produktionskette einer Abschlussarbeit ohne methodisch kritische Brüche tragen kann. Als kritisch gelten in dieser Arbeit Brüche, wenn sie zu Strukturverlust, inkonsistenter Zitation, nicht kontrollierbaren Exporten, betreuungspraktischen Reibungen oder institutionell problematischem Betrieb führen.

Tab. 3: Bewertungsdimensionen der Analyse.

Dimension	Leitfrage	Beobachtbare Indikatoren
Strukturtransparenz	Bleibt der Argumentationsgang im Arbeitsprozess sichtbar und stabil?	Gliederung, Rohtext-Lesbarkeit, Umstrukturierbarkeit
Quellenapparat	Können Zitationen datenbasiert, konsistent und stiladaptiv verarbeitet werden?	Bibliografiepfad, Citeproc, Referenzsektion, Stilwechsel
Layout und Export	Lassen sich drucknahe Ausgaben früh und reproduzierbar kontrollieren?	Seitenansicht, PDF, DOCX, Listen, Caption-Verhalten
Review und Betreuung	Ist der Workflow mit Feedback- und Abgabekulturen vereinbar?	Teilen, Kommentierbarkeit, Export in Review-Formate
Governance und Betrieb	Ist das Werkzeug rechtlich und technisch für längere akademische Nutzung tragfähig?	Lizenz, Self-Hosting, Portabilität, Abhängigkeiten

Die Analyse stützt sich nicht nur auf Funktionsbeschreibungen, sondern auch auf das im Anhang dokumentierte Referenzartefakt. Dieses Artefakt enthält bewusst Abbildungen, Tabellen, Mathematik,

Fußnoten, Spalten und Seitenumbrüche, weil gerade diese Elemente in wissenschaftlichen Abschlussarbeiten typische Fehlerquellen des Exportpfads markieren.

## 5.2 Strukturtransparenz und Schreibprozess

Im Kriterium Strukturtransparenz erzielt mdedit.io eine hohe Eignung. Die Kombination aus Plain-Text-Basis, Kapitelhierarchie und Outline-Navigation erleichtert es, Argumentationssegmente früh zu verschieben, Lücken sichtbar zu machen und Rohfassungen ohne Formatballast zu überarbeiten (Hertel 2026c). Gegenüber Word reduziert dies die Versuchung lokaler Formatkorrekturen; gegenüber LaTeX sinkt die initiale Toolchain-Komplexität.

Für Masterarbeiten ist dieser Punkt nicht trivial, weil Abschlussarbeiten typischerweise mehrfach restrukturiert werden. Ein Werkzeug, das Gliederungsstabilität unter Überarbeitung trägt, besitzt daher einen echten methodischen Mehrwert. In dieser Dimension ist mdedit.io nicht nur benutzbar, sondern für viele Schreibsituationen dem klassischen WYSIWYG-Paradigma sogar analytisch überlegen.

### 5.2.1 Navigation in großen Dokumenten und Tree View

Für umfangreiche wissenschaftliche Dokumente reicht eine lineare Editoransicht jedoch nicht aus. mdedit.io adressiert dieses Problem über eine heading-basierte Outline und eine Baumansicht, die laut Produktdokumentation ausschließlich auf Überschriftenebenen H1 bis H6 basiert (Hertel 2026c). Die Hilfe bezeichnet die Baumansicht dabei nicht als dekorativen Zusatz, sondern als eigene Arbeitsfläche für Vorschau, Baum und KI-Chat (Hertel 2026b). Gerade in langen Theoriekapiteln, Methodenabschnitten und Anhängen wird dadurch sichtbar, ob Kapitelgewicht, Unterkapitelstruktur und argumentative Hierarchie noch tragfähig sind.

Methodisch ist das mehr als Komfort. Eine Masterthesis scheitert häufig nicht an fehlenden Einzelsätzen, sondern an schlechter Makrostruktur. Ein Tree View, der Abschnittstiefe, Reihenfolge und Schwerpunktsetzung sofort sichtbar macht, verkürzt den Weg zwischen Gliederungsentscheidung und Textrevision. Seine Grenze liegt allerdings ebenfalls offen: Wenn die Baumansicht ausschließlich headings-basiert ist, bildet sie semantische Referenzobjekte wie Abbildungen, Tabellen, Listings oder bibliografische Knoten nicht mit eigener Logik ab. Damit bleibt sie ein starkes Strukturwerkzeug, aber kein vollständiges semantisches Navigationssystem.

### 5.2.2 KI-Integration für die Inhaltsbearbeitung

mdedit.io integriert KI nicht nur als externen Prompt-Kanal, sondern als eingebettetes Arbeitsmittel für Dokumentbearbeitung. Die README spricht von AI-assisted editing, die Hilfe von einem integrierten KI-Panel (Hertel 2026c, 2026b). Die API-Dokumentation zeigt zudem, dass der Bearbeitungspfad nicht auf reine Chat-Antworten beschränkt ist, sondern strukturierte Aktionen wie `REPLACE`, `INSERT`, `APPEND`, `PREPEND` und `ADVICE` vorsieht (Hertel 2026a). Für die Arbeit an einer Masterthesis ist das vor allem dort relevant, wo Einleitungen gestrafft, Übergänge umformuliert, Zusammenfassungen verdichtet oder alternative Gliederungsvarianten schnell erzeugt werden sollen.

Der methodisch entscheidende Punkt liegt jedoch tiefer: Die KI arbeitet in mdedit.io auf einem kanonischen Markdown-Kern. Dadurch bleiben Überschriftenhierarchie, Listen, Tabellen, Zitationsmarker und Frontmatter nicht nur sichtbar, sondern direkt bearbeitbar. Im Unterschied zu visuell dominierten Dokumentformaten muss die KI die Struktur also nicht erst aus impliziter Formatierung rekonstruieren. Gerade für wissenschaftliche Texte ist das relevant, weil die Bearbeitung dadurch näher am semantischen Dokumentzustand als an einer bloßen Oberflächenumformulierung bleibt.

Die Eignung dieser Integration ist gleichwohl konditional. Für wissenschaftlich legitime Nutzung eignet sich KI zur sprachlichen Revision, zur Verdichtung, zur Erzeugung von Zwischenüberschriften, zur Umstellung von Argumentationsfolgen und zur Explikation stillschweigender Strukturannahmen. Sie eignet sich nicht als Quelle eigener empirischer Aussagen, nicht für unbelegte Literaturbehauptungen und nicht für die Simulation fachlicher Evidenz. Die in mdedit.io sinnvolle Rolle der KI ist deshalb editorisch und heuristisch, nicht epistemisch.

### 5.2.3 KI-Integration für die Layoutbearbeitung

Die KI-Integration von mdedit.io ist nach dem aktuellen Intent-Modell nicht auf Fließtext beschränkt. Im Server werden auch layoutbezogene Bearbeitungsanliegen wie `layout`, `seitenrand`, `typografie`, `spalten`, `header`, `footer` und `inhaltsverzeichnis` explizit als Edit-Intent erkannt (Hertel 2026d). Zusammen mit dem Layout-Editor, den drucknahen Preview-Presets und dem Paged-Workflow entsteht damit ein integrierter Bearbeitungspfad, in dem Layoutregeln nicht nur manuell gesetzt, sondern auch durch KI-gestützte Formulierungs- oder Anpassungsanfragen vorbereitet werden können (Hertel 2026b).

Gerade für Masterarbeiten ist diese Kopplung interessant, weil Layoutfragen selten isoliert auftreten. Wer etwa die Abbildungsdichte in einem Ergebniskapitel erhöht, muss oft zugleich Spalten, Caption-Stile, Seitenumbrüche oder Inhaltsverzeichnis-Tiefe anpassen. Die relevante Einschränkung besteht darin, dass diese KI-Integration kein Garant für typografische Richtigkeit ist. Sie kann Layoutarbeit beschleunigen, aber sie ersetzt nicht die Sichtprüfung in der Seitenansicht und nicht die formale Endkontrolle gegen institutionelle Vorgaben.

## 5.3 Quellenapparat und wissenschaftliche Integrität

Im Bereich Quellenapparat ist mdedit.io nur dann thesis-tauglich, wenn Zitate konsequent datenbasiert und nicht als manuell formatierter Text behandelt werden. Die vorliegende Referenzdatei operationalisiert diese Logik bereits über YAML-Frontmatter, `citation-source: embedded`, einen dokumentgebundenen `mdedit-bibliography`-Block, verlinkte Zitationen und die explizite Referenzsektion. In Verbindung mit Pandoc und Citeproc ist dies für autor-jahr- und numerische Workflows tragfähig (MacFarlane 2025; Hertel 2026e, 2026d).

### 5.3.1 Integration von Literaturverzeichnissen und Bib-Standards

Gerade mit Blick auf die aktuellen mdedit-Erweiterungen ist festzuhalten, dass der wissenschaftliche Quellenpfad inzwischen über klassisches Plain-Text-Markdown hinausgeht. Der Server wertet in wissenschaftlichen Dokumenten vor allem YAML-Felder wie `citation-source`, `cs1` und `reference-section-title` aus und erkennt zusätzlich `link-citations`, `link-bibliography` sowie `nocite` als relevante Metadaten für den Export (Hertel 2026d). Der normative lokale Quellenpfad ist dabei ein `mdedit-bibliography`-Block mit eingebettetem CSL-JSON, während der Export selbst über Pandoc mit `--citeproc` aufgebaut wird (MacFarlane 2025; Hertel 2026d). Dateibasierte `bibliography:`-Pfade werden in der aktuellen Serverlogik zwar noch erkannt, aber bewusst nur noch für klare Migrationsfehler ausgewertet und nicht mehr als gültiger Exportworkflow akzeptiert (Hertel 2026d, 2026f).

Für die vorliegende Referenzdatei ist besonders wichtig, dass sie diesen eingebetteten Pfad nun selbst verwendet. Das Dokument enthält die Bibliothek als lokalen Snapshot im Quelltext und testet damit nicht nur Zitationssyntax, sondern den kompletten dokumentgebundenen Quellenmodus. Zusätzlich bleibt der aktuelle mdedit-Marker `#refs` bewusst erhalten. Dieser Marker ist im Quelltext nicht als sichtbare Endfassung gedacht, sondern Teil der gegenwärtigen Scientific-Schnittstelle: Im Exportpfad wird er serverseitig in eine

echte Literaturverzeichnis-Sektion mit Referenzanker normalisiert ([Hertel 2026d](#)). Genau deshalb ist `#refs` hier keine editorische Nachlässigkeit, sondern ein End-to-End-Test der aktuellen Bibliografieintegration.

Die Import- und Exportperspektive bleibt davon zu unterscheiden. In der Produktplanung sind Adapter für BibTeX, RIS und CSL JSON auf dem lokalen Dokumentkern vorgesehen, damit bestehende Literaturbestände in die eingebettete Bibliothek übernommen und aus ihr wieder exportiert werden können ([Hertel 2026f](#)). Damit wird das integrierte mdedit-Format nicht als Abschottung gegen bestehende Standards gedacht, sondern als reproduzierbarer Dokumentkern, an den Standardimporte und -exporte anschließen.

### 5.3.2 Optionen zur Einbindung externer Online-Verzeichnisse

Die Planung für wissenschaftliche Dokumente unterscheidet den dokumentgebundenen Quellenmodus `embedded` sowie die späteren Modi `zotero` und `hybrid` ([Hertel 2026f](#)). Daraus folgt eine für Abschlussarbeiten wichtige Architekturentscheidung: Externe Online-Verzeichnisse sollen Recherche, Auswahl und Metadatenanreicherung unterstützen, aber nicht die einzig normative Exportquelle der eingereichten Arbeit sein.

Als erster externer Connector ist ein read-only-Zotero-Pfad vorgesehen. Quellen sollen dort gesucht und anschließend in die lokale Bibliothek oder in dokumentgebundene Snapshots übernommen werden können, ohne dass der Export oder das Teilen des Dokuments zur Laufzeit von einer Live-Zotero-Verbindung abhängt ([Hertel 2026f](#)). Parallel dazu ist OpenAlex als offener Lookup für Titel-, DOI- oder Autorensuche geplant; auch hier sollen Treffer in die lokale Bibliothek übernommen werden, während OpenAlex selbst nicht die normative Exportquelle des Dokuments bleibt ([Hertel 2026f](#)).

Für thesis-taugliche Workflows ist genau diese Snapshot-Logik zentral. Wer externe Online-Verzeichnisse einbindet, sollte sie als vorgelagerte Recherche- und Importebene behandeln, nicht als einzige Speicherstelle der abgaberelevanten Bibliografie. Reproduzierbar wird der Workflow erst dann, wenn der für die Abgabe maßgebliche Quellenstand als eingebettete mdedit-Bibliothek oder als kontrollierter Hybrid-Snapshot im Dokumentkontext fixiert bleibt ([Hertel 2026f, 2026e](#)).

Die Grenze verläuft dort, wo note-style-Fußnoten, komplexe Cross-References oder hochschulspezifische Sonderlogiken produktiv vorausgesetzt werden. Für solche Regime reicht die gegenwärtige Sicherheit des HTML- und Paged-Stacks nach den eigenen Projektquellen noch nicht aus ([Hertel 2026e](#)). Das Urteil fällt daher bewusst konditional aus: hoch anschlussfähig für datenbasierte Standardworkflows, eingeschränkt für juristisch oder vorlagentechnisch stark normierte Spezialfälle.

### 5.4 Layout-, Satz- und Exportstabilität

Im Kriterium Layout und Export befindet sich mdedit.io in einer starken, aber nicht universellen Position. Scientific-Preset, Seitenansicht und PDF- sowie DOCX-Export erlauben eine frühe Kontrolle von Umbrüchen, Tabellen, Abbildungen und Verzeichnissen ([Hertel 2026c](#)). Für die vorliegende Referenzdatei ist bewusst kein lokaler `layout`-Block gesetzt; damit wird geprüft, wie tragfähig bereits der globale wissenschaftliche Preset-Pfad ist.

Der methodische Vorteil dieser Entscheidung liegt darin, dass kein institutionsspezifischer Spezialfall verdeckt wird. Wenn das Dokument unter dem globalen Scientific-Preset stabil bleibt, spricht das für eine robuste Basistauglichkeit. Wenn Sonderfälle erst mit lokalen Layout-Overrides auflösbar sind, müssen sie als profilgebundene Zusatzanforderungen ausgewiesen werden und dürfen nicht als allgemeine Thesis-Tauglichkeit missverstanden werden.

## 5.5 Review-, Betreuungs- und Kollaborationsfähigkeit

Für eine reale Masterarbeit reicht technische Satzfähigkeit allein nicht aus; ebenso wichtig ist die Anschlussfähigkeit an Betreuung. Hier bleibt Word durch Track Changes, kommentierbare DOCX-Workflows und institutionelle Routine stark ([Microsoft Support 2026](#)). mdedit.io setzt stattdessen auf browserbasiertes Teilen, direkte Verfügbarkeit und Multi-Export ([Hertel 2026c](#)). Das ist für digitale Betreuungsszenarien plausibel, aber nicht voll äquivalent zu etablierten Word-Markup-Routinen.

Die Bewertung lautet deshalb nicht, mdedit.io sei generell besser als Word, sondern dass es unter anderen Betreuungsbedingungen tragfähig ist. Sobald Betreuende zwingend .docx-Kommentarworkflows erwarten, muss der Exportpfad Teil des methodischen Arbeitsplans werden. Unter dieser Bedingung ist mdedit.io kompatibel; ohne sie kann es zu sozial-organisatorischen und nicht zu technischen Brüchen kommen.

## 5.6 Ausschlusskriterien und Grenzfälle

Aus der Analyse ergeben sich klare Grenzfälle, in denen mdedit.io gegenwärtig nicht als primäre Umgebung empfohlen werden sollte: erstens bei strikt vorgeschriebenen Word-Schablonen mit fragiler Formatsemantik, zweitens bei juristischen note-style-Zitierregimen und drittens bei Projekten, die semantisch reichhaltige Cross-References und automatische Referenzobjekte zwingend voraussetzen ([Hertel 2026e](#), [2026f](#)).

Umgekehrt ist mdedit.io besonders geeignet für textzentrierte, argumentativ strukturierte und exportorientierte Arbeiten in Geistes-, Sozial-, Wirtschafts- und Teilen der Informatik- oder Lebenswissenschaften, sofern die formale Zielarchitektur mit autor-jahr- oder numerischen Zitationen sowie einem kontrollierten PDF- oder DOCX-Pfad kompatibel ist.

Tab. 4: Zusammenfassende Bewertung von mdedit.io im Anforderungsraster.

Bewertungsdimension	Urteil	Begründung
Strukturtransparenz	hoch	Outline, Rohtext-Lesbarkeit und geringer Formatballast stabilisieren die Schreibphase
Quellenapparat	mittel bis hoch	stark für datenbasierte Standardworkflows, eingeschränkt bei note-style und Speziallogiken
Layout und Export	hoch	frühe Seitenkontrolle und Multi-Export, aber Sonderfälle bleiben profilabhängig
Review und Betreuung	mittel	stark bei Teilen und Browserzugang, eingeschränkt gegenüber institutionellen Word-Routinen
Governance und Betrieb	hoch	Self-Hosting, Lizenzklarheit und portable Dokumentbasis sprechen für institutionelle Tragfähigkeit

## 5.7 Belastungsprobe an anspruchsvollen Hochschulprofilen

Um die Basistauglichkeit nicht nur generisch zu behaupten, wird mdedit.io gegen drei anspruchsvolle Hochschulprofile gespiegelt. Diese Profile stehen exemplarisch für juristische, informatische und lebenswissenschaftliche Randbedingungen und markieren damit unterschiedliche Belastungspunkte des Systems.

### 5.7.1 University of Oxford, Law

Das Oxford-Law-Profil ist der fachlich strengste Kandidat für juristische Arbeiten. OSCOLA ist explizit für die präzise Zitation von Rechtsprechung, Gesetzgebung und juristischen Sekundärquellen konzipiert und wird von der Oxford Law Faculty herausgegeben ([Faculty of Law, University of Oxford 2026](#)). Schon die offizielle Quick-Reference zeigt, dass sich das Regime deutlich von klassischen Autor-Jahr-Systemen unterscheidet: Fälle, Gesetze, Kommentare, Aufsätze und Reports folgen unterschiedlichen, formal stark normierten Mustern ([Faculty of Law, University of Oxford 2026](#)).

Für mdedit.io folgt daraus: Das Scientific-Preset eignet sich für Grundtypografie, Randruhe und Tabellenlayout, aber ein echtes Oxford-Law-Profil bräuchte zusätzlich note-style-Fußnoten, juristische Kurzformen, saubere Quellenabkürzungen und einen legal-spezifischen Exportpfad. Genau deshalb markiert Oxford Law kein generisches Erfolgsszenario, sondern ein bewusstes Ausschluss- oder Entwicklungsprofil.

### 5.7.2 ETH Zurich, Computer Science

Das ETH-Profil ist für Informatik besonders interessant, weil die offizielle D-INFK-Memo die Masterarbeit als eigenständige wissenschaftliche Forschung oder konstruktive Entwicklungsarbeit mit schriftlichem Bericht und mündlicher Präsentation definiert ([Department of Computer Science, ETH Zurich 2025](#)). Die Memo benennt zudem klare Rahmenbedingungen wie die 28-wöchige Vollzeitdauer und die verpflichtende elektronische Einreichung eines PDF inklusive unterschriebener Eigenständigkeitserklärung bis spätestens 23:59 Uhr am Enddatum ([Department of Computer Science, ETH Zurich 2025](#)).

Für mdedit.io ist dieses Profil vergleichsweise gut anschlussfähig. Es verlangt weniger exotische Zitationslogik als Oxford Law, dafür aber einen stabilen wissenschaftlichen Produktionsablauf mit Titelblatt, Abstract, Declaration, Inhaltsverzeichnis, Abbildungs- und Tabellenlogik, Codeanhängen und belastbarer PDF-Abgabe. Als Belastungsprobe zeigt ETH damit ein realistisches Hochanspruchsprofil, das mit einem wissenschaftlich disziplinierten Markdown-Workflow grundsätzlich erreichbar ist.

### 5.7.3 Imperial College London, Medicine and Life Sciences

Das Imperial-Profil repräsentiert einen natur- und lebenswissenschaftlichen Kontext mit numerischer Zitation. Die offizielle Vancouver-Guidance beschreibt ein nummeriertes Zitiersystem, in dem Verweise im Text numerisch erscheinen und die Referenzliste am Ende in numerischer Reihenfolge geführt wird; nicht zitierte, aber konsultierte Quellen können optional als alphabetische Bibliografie separat erscheinen ([Imperial College London Library Services 2026](#)). Für längere Zitate wird zudem ein eigener eingerückter Absatz gefordert ([Imperial College London Library Services 2026](#)).

Technisch ist dies das am leichtesten produktivisierbare Profil für mdedit.io. Vancouver lässt sich über Citeproc und CSL gut auf ein Scientific-Preset aufsetzen. Ein Imperial-nahes Profil wäre damit kein Spezialexperiment, sondern ein realistischer Zielzustand für numerische, exportorientierte und figuresensible Abschlussarbeiten.

## 5.8 Gestaltungsimplicationen für institutionelle Profile

Aus der Belastungsprobe ergibt sich, dass sich Hochschulprofile seltener in der eigentlichen Seitengeometrie unterscheiden als in Zitation, Pflichtsektionen, Validierungsregeln und Exportanforderungen. Genau daraus folgt eine konkrete Gestaltungsimplication für mdedit.io: Institutionelle Thesis-Profile sollten nicht primär als hart verdrahtete Layout-Sonderfälle modelliert werden, sondern als auf dem Scientific-Preset aufsetzende Profilupakete ([Hertel 2026f](#)).

Ein belastbares Modell wäre fünfstufig:

1. `base=scientific` : Das globale Scientific-Preset bleibt die typografische Grundschrift für alle wissenschaftlichen Profile.
2. Zitationspaket: CSL, Zitiermodus, Bibliografie-Regeln und optional note-style-Exportlogik werden pro Profil gebündelt.
3. Metadatenpaket: Titelblattfelder, Eigenständigkeitserklärung, Abstract-Sprachen, Appendix-Regeln und Pflichtsektionen werden als Profilvergaben geliefert.
4. Validierungspaket: Ein Profil prüft, ob etwa OSCOLA-Fußnoten, Vancouver-Nummern, ETH-Declaration oder fakultätsspezifische Abschnittsreihenfolgen erfüllt sind.
5. Distributionspaket: Profile werden als herunterladbare Manifeste oder kleine Pakete ausgeliefert, damit Hochschulen, Lehrstühle oder Communities sie versionieren und teilen können.

Ein mögliches Manifest könnte konzeptionell so aussehen:

```
id: ethz-dinfk-masterthesis
base: scientific
citationStyle: author-date
bibliographyMode: reference-list
requiredSections:
- title-page
- abstract
- declaration-of-originality
- references
validators:
- declaration-required
- figures-and-tables-numbered
- bibliography-present
exports:
pdf: required
docx: optional
```

Der entscheidende Punkt ist dabei: Nicht jedes Profil braucht ein eigenes Dokument-Layout. Viele Institutionen unterscheiden sich stärker in Zitation, Frontmatter, Pflichtsektionen und Validierung als in der eigentlichen Seitengeometrie. Genau deshalb ist die vorliegende Referenzdatei absichtlich layout-block-frei und eignet sich als Basisszenario für ein späteres institutionelles Profilsystem.

## 5.9 Gründe gegen mdedit.io als Werkzeug der Wahl für die eigene Masterthesis

Auch nach einer wohlwollenden kriteriologischen Bewertung bleibt eine lange Liste legitimer Gründe, mdedit.io gerade nicht als primäres Werkzeug für die eigene Masterthesis zu wählen. Die wichtigsten

Gegenargumente lauten:

1. Die Hochschule schreibt eine starre Word-Vorlage vor, die nicht nur formal, sondern technisch bindend ist.
2. Die Betreuung arbeitet verbindlich mit `Track Changes`, Kommentarspalten und reviewerbezogenen DOCX-Routinen, sodass ein nachgelagerter Export nicht nur bequem, sondern zentral für den Prozess ist ([Microsoft Support 2026](#)).
3. Das Zitationsregime verlangt echte note-style-Fußnoten, juristische Kurzformen oder OSCOLA-ähnliche Nachweise, die im gegenwärtigen mdedit-Stack nicht hinreichend abgesichert sind ([Faculty of Law, University of Oxford 2026](#); [Hertel 2026e](#)).
4. Die Arbeit benötigt semantisch reiche Cross-References zwischen Abbildungen, Tabellen, Formeln, Listings und Anhängen, die nicht auf manuelle Querverweise reduziert werden sollen ([Hertel 2026e](#)).
5. Das Fach verlangt typografische Tiefenkontrolle, spezielle LaTeX-Pakete oder mathematische Satzqualität, die über einen Markdown-Pandoc-Pfad hinausgeht ([LaTeX Project 2026](#)).
6. Die wissenschaftliche Infrastruktur des Lehrstuhls basiert bereits stabil auf Overleaf, LaTeX-Klassen oder institutionell gepflegten Templates, sodass ein Werkzeugwechsel mehr Risiko als Erkenntnisgewinn erzeugen würde.
7. Das Projekt ist stark code-, listing- oder tabellenlastig und verlangt ein semantisch reiches, technisch ausgereiftes Publishing-System statt eines strukturorientierten Zwischensystems.
8. Die Autorin oder der Autor möchte während des gesamten Schreibprozesses maximale Sichtformatierung und unmittelbare WYSIWYG-Kontrolle statt einer Markdown-zentrierten Produktionslogik.
9. Die Betreuungs- oder Prüfungsumgebung ist organisatorisch konservativ und bewertet ungewohnte Exportpfade skeptisch, selbst wenn das Ergebnis formal korrekt ist.
10. Es steht keine verlässliche Self-Hosting- oder Betriebsumgebung zur Verfügung, sodass Datenschutz, Verfügbarkeit oder Langzeitstabilität nicht institutionell kontrolliert werden können ([Hertel 2026c](#)).
11. Der Arbeitsprozess soll bewusst ohne KI-Nähe, ohne Browser-Artefakte und ohne editorische Mehrsystemlogik gehalten werden.
12. Das Projekt ist gegenüber Exportdrift oder Preview-Diskrepanzen extrem risikosensibel und will ausschließlich auf maximal etablierte, langjährig erprobte Abschlussarbeits-Workflows setzen.
13. Die eigene Schreibpraxis ist bereits tief in Zotero-Word- oder TeX-Bibliografie-Workflows integriert, sodass mdedit keinen realen Produktivitätsvorteil mehr bietet.
14. Die Stabilisierungsarbeit am wissenschaftlichen Sonderfall würde mehr Zeit kosten als die Thesis selbst an Erkenntnisgewinn bringt.

Aus dieser Liste folgt kein generelles Urteil gegen mdedit.io. Sie zeigt jedoch, dass die Entscheidung für oder gegen ein Werkzeug der Wahl immer als Passungsentscheidung zwischen Fachkultur, Prüfungslogik, Betreuungspraxis und individueller Arbeitsweise gelesen werden muss.

## 6. Governance, Rechte und Betriebsfähigkeit

### 6.1 Lizenzierung und Rechte

Nach der Projektdokumentation ist der Quellcode von mdedit.io unter Apache License 2.0 lizenziert; ausgenommen sind gesondert gekennzeichnete Brand-Assets und Markenbestandteile ([Hertel 2026c](#)). Das ist für wissenschaftliche Nutzung relevant, weil es Transparenz über die rechtliche Basis schafft und die Anwendung nicht als proprietäre Black Box behandelt werden muss. Für Hochschulen, Arbeitsgruppen oder

Einzelpersonen, die auf nachvollziehbare und selbst betreibbare Werkzeuge Wert legen, ist dies ein substantieller Vorteil.

## 6.2 Nutzungszugang und praktische Nutzbarkeit

Die Nutzbarkeit von mdedit.io ergibt sich vor allem aus vier Faktoren: browserbasierter Zugang, keine verpflichtende Kontoanlage, private Dokumente per Session-Modell und eine Funktionsoberfläche, die sich stark am Schreibprozess orientiert (Hertel 2026c). Gerade für Studierende ist dies relevant, weil der Einstieg ohne lokale Desktop-Installation möglich ist und der Arbeitskontext zwischen verschiedenen Geräten leichter beweglich wird.

## 6.3 Verfügbarkeit und Betrieb

mdedit.io ist als selbst gehostete Webanwendung dokumentiert. Die README beschreibt sowohl einen Docker-basierten Standardpfad als auch einen direkten Node.js-Start für lokale Entwicklung (Hertel 2026c). Zudem werden Browser-Abhängigkeiten lokal gebündelt, sodass der aktive Laufzeitpfad nicht auf externe Browser-CDNs angewiesen ist (Hertel 2026c). Diese Verfügbarkeitslogik ist für datensensible oder institutionell kontrollierte Umgebungen von Bedeutung, weil Betrieb, Update und Datenhaltung nicht zwangsläufig in fremde SaaS-Infrastrukturen ausgelagert werden müssen.

## 7. Diskussion

Die Arbeit beantwortet die Forschungsfrage differenziert. Ein browserbasierter Markdown-Editor kann als primäre Produktionsumgebung für wissenschaftliche Abschlussarbeiten dienen, wenn das Anforderungsprofil der Arbeit durch strukturorientiertes Schreiben, datenbasierte Zitation, kontrollierten Multi-Export und digital anschlussfähige Betreuung gekennzeichnet ist. Unter diesen Bedingungen ist mdedit.io nicht nur ein bequemes Schreibwerkzeug, sondern ein tragfähiges wissenschaftliches Artefakt.

Der wissenschaftliche Beitrag der Arbeit liegt weniger in der Behauptung universeller Überlegenheit als in einem expliziten Bewertungsmodell. Die übliche Debatte über Word, LaTeX und Markdown wird oft normativ oder identitär geführt; die vorliegende Untersuchung übersetzt sie in prüfbare Kriterien und zeigt, dass die Eignung eines Werkzeugs von der Kopplung zwischen Dokumenttyp, Zitationsregime, Betreuungspraxis und Exportarchitektur abhängt. In dieser Perspektive erscheint mdedit.io nicht als vollständiger Ersatz, wohl aber als belastbare Lösung für ein klar umrissenes Segment wissenschaftlicher Abschlussarbeiten.

Die Ergebnisse bleiben gleichwohl begrenzt. Es wurde keine Nutzerstudie, kein Zeitvergleich realer Schreibprozesse und kein breit angelegter Template-Benchmark mit institutionellen Vorlagen durchgeführt. Die Befunde sind daher als analytische Eignungsprüfung und nicht als empirischer Wirksamkeitsnachweis zu lesen. Genau daraus ergibt sich der nächste Forschungsschritt: kontrollierte Fallstudien, Exportparitätstests und institutionelle Profilpakete auf Basis realer Abgabeanforderungen.

## 8. Fazit

Die Forschungsfrage kann wie folgt beantwortet werden: mdedit.io kann dann als primäre Produktionsumgebung für wissenschaftliche Abschlussarbeiten dienen, wenn die Arbeit ein strukturorientiertes, datenbasiertes und exportkontrolliertes Authoring erfordert und keine zwingenden note-style- oder Template-Sonderfälle vorliegen. Unter diesen Bedingungen ist ein browserbasierter Markdown-Workflow methodisch vertretbar, technisch belastbar und institutionell anschlussfähig.

Markdown erweist sich dabei nicht als pauschaler Ersatz für Word oder LaTeX, sondern als eigenständige Produktionslogik für wissenschaftliche Langtexte (Gruber 2004; CommonMark Contributors 2026; MacFarlane 2025). Word bleibt dort stark, wo institutionelle Review-Routinen dominieren; LaTeX bleibt dort überlegen, wo typografische und referenztechnische Tiefenkontrolle entscheidend ist. mdedit.io besetzt die Zone dazwischen: stark in Strukturtransparenz, heading-basierter Tree-Navigation, KI-gestützter Dokumentarbeit, frontmatter- und citeproc-basierter Bibliografieintegration, früher Layoutkontrolle und kontrolliertem Multi-Export, begrenzt bei stark normierten Sonderfällen (Hertel 2026c, 2026b, 2026d, 2026e).

Der Mehrwert der Arbeit liegt deshalb in einer begründeten Entscheidungshilfe. Sie zeigt nicht nur, dass mdedit.io für bestimmte Masterarbeiten tragfähig ist, sondern auch, unter welchen Bedingungen dies nicht gilt. Gerade diese konditionale Antwort ist für wissenschaftliche Praxis wertvoller als ein pauschales Ja oder Nein.

## Literaturverzeichnis

- American Psychological Association. 2019. *Publication Manual of the American Psychological Association*. 7. Aufl. American Psychological Association.
- Booth, Wayne C., Gregory G. Colomb, Joseph M. Williams, Joseph Bizup, und William T. Fitzgerald. 2016. *The Craft of Research*. 4. Aufl. Chicago: University of Chicago Press.
- CommonMark Contributors. 2026. „CommonMark“. <https://commonmark.org/>.
- Department of Computer Science, ETH Zurich. 2025. „Memo: Master’s Theses in Computer Science“.
- Faculty of Law, University of Oxford. 2026. „The Oxford University Standard for Citation of Legal Authorities (OSCOLA), 5th Edition“. <https://www.law.ox.ac.uk/oscola>.
- Gruber, John. 2004. „Markdown“. <https://daringfireball.net/projects/markdown/>.
- Hertel, Matthias. 2026a. „AI Chat API Dokumentation“.
- . 2026b. „mdedit.io Help Page“.
- . 2026c. „mdedit.io README“.
- . 2026d. „mdedit.io Scientific Citation and Export Implementation“.
- . 2026e. „Scientific Documents and Citations Concept“.
- . 2026f. „Scientific Documents Implementation Plan“.
- Hevner, Alan R., Salvatore T. March, Jinsoo Park, und Sudha Ram. 2004. „Design Science in Information Systems Research“. *MIS Quarterly* 28 (1): 75–105.
- Imperial College London Library Services. 2026. „Vancouver Style“. <https://www.imperial.ac.uk/admin-services/library/learning-support/reference-management/vancouver-style/>.
- LaTeX Project. 2026. „An Introduction to LaTeX“. <https://www.latex-project.org/about/>.
- MacFarlane, John. 2025. „Pandoc“. <https://pandoc.org/>.
- Microsoft Support. 2026. „Track Changes in Word“. <https://support.microsoft.com/en-us/office/track-changes-in-word-197ba630-0f5f-4a8e-9a77-3712475e806a>.
- Paver, Chloe, Graham Nelson, und Simon F. Davies, Hrsg. 2024. *MHRA Style Guide*. 4. Aufl. Modern Humanities Research Association.
- Swales, John M., und Christine B. Feak. 2012. *Academic Writing for Graduate Students: Essential Tasks and Skills*. 3. Aufl. Ann Arbor: University of Michigan Press.

University of Reading Library. 2026. „Different Styles and Systems of Referencing“ .  
<https://libguides.reading.ac.uk/citing-references/referencingstyles>.

Wieringa, Roel J. 2014. *Design Science Methodology for Information Systems and Software Engineering*.  
Berlin: Springer.

## Anhang A. Reproduzierbares Referenzartefakt zur Layout- und Funktionsprüfung

Der folgende Anhang ist kein dekorativer Zusatz, sondern Teil des Untersuchungsdesigns. Er bündelt jene Elemente, die im wissenschaftlichen Exportpfad typischerweise fehleranfällig sind, und dient damit als reproduzierbares Prüfartefakt der Arbeit.

### A.1 Abbildungsprobe



Abb. 1: Neutrales Bildartefakt für Caption-, Listen- und Printtests

### A.2 Wissenschaftliche Tabelle

Tab. 5: Kriterienset für die Endprüfung einer wissenschaftlichen Arbeit in mdedit.io.

Kriterium	Erwartung	Zweck
Kapitelstruktur	Vollständig und logisch	Argumentationsführung prüfen
Zitation	Konsistent und datenbasiert	Quellenintegrität sichern
Export	PDF und DOCX kontrolliert	Abgabe- und Reviewpfade absichern
Layout	Seiten, Tabellen, Bilder stabil	Formale Belastbarkeit prüfen

### A.3 Kompakte Tabelle

Tab. 6: Typische Aufgaben vor der Abgabe.

Aufgabe	Status
Forschungsfrage schärfen	offen
Literaturverzeichnis prüfen	offen
PDF-Endkontrolle durchführen	offen
DOCX für Betreuung exportieren	optional

## A.4 Mathematik- und Fußnotenprobe

Die Formel  $E = mc^2$  dient als Inline-Test. Für einen Blocktest eignet sich folgende Darstellung:

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Zusätzlich bleibt eine Fußnote als Satz- und Exportprobe erhalten.<sup>1</sup>

## A.5 Mehrspaltenprobe

Wissenschaftliche Haupttexte brauchen selten Spalten; Anhänge und Sonderseiten können jedoch von einem zweispaltigen Layout profitieren. Die Zweispaltensyntax (`columns{count=2}`) wird im Editor-Preview unterstützt. Für den PDF-Export muss das Layout explizit im Layout-Editor konfiguriert werden.

## A.6 Unicode-, Querverweis- und Listingprobe

Wie in Abbildung 1 und Tabelle 5 sichtbar wird, braucht eine Masterthesis nicht nur stabile Seiten, sondern auch belastbare textuelle Querverweise im Fließtext. Diese Probe bleibt bewusst einfach und testet, wie sich Referenzformulierungen im Satz verhalten, solange semantische Cross-References nicht in jedem Sonderfall automatisiert verfügbar sind.

Die deutsche Typografieprobe enthält absichtlich echtes Unicode: für, äußere, Einführung, „deutsche Anführungszeichen“, Gedankenstrich – und § 1 Abs. 2. Damit lässt sich prüfen, ob Kopierbarkeit, Suchbarkeit und Zeichendarstellung im Export stabil bleiben.

Listing 1: Beispielhafte Profilkonfiguration für ein institutionelles Thesis-Paket.

```
id: example-masterthesis
base: scientific
citationStyle: author-date
requiredSections:
- abstract
- references
```

Zusätzlich dient eine lange URL als Umbruchprobe im Lauftext:  
<https://example.org/ein/sehr/langer/pfad/mit/vielen/segmenten/und-parametern?alpha=123&beta=456&gamma=789>

<sup>1</sup> Die Fußnote prüft, ob Lauftext, Marker und Fußnotenbereich in der Ausgabe stabil lesbar bleiben.

