

KI-gestütztes Reverse Requirements Engineering bei Legacy-Software

Masterarbeit an der Hochschule Neu-Ulm

Masterarbeit

Master of Science

Autor: Christoph Musterfrau

Betreuer: Prof. Dr. Daniel Schallmo

Abgabedatum: 31. März 2026

University of Applied Sciences Neu-Ulm

Eigenständigkeitserklärung

Hiermit versichere ich, die vorliegende Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben. Alle wörtlich oder sinngemäß übernommenen Textstellen sind als solche gekennzeichnet.

Neu-Ulm, _____

Unterschrift: _____

Abstract (ca. 1 Seite)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ergänze hier die Zusammenfassung der Arbeit.

Inhaltsverzeichnis

Eigenständigkeitserklärung	2
Abstract (ca. 1 Seite)	3
Abstract (ca. 1 Seite)	2
1 Einleitung (ca. 8 Seiten)	2
1.1 Ausgangssituation und Motivation	2
1.2 Problemstellung	2
1.3 Zielsetzung	3
1.4 Forschungsleitfragen	3
1.5 Aufbau der Arbeit	3
2 Theoretische Grundlagen (ca. 12 Seiten)	4
2.1 Requirements Engineering und Reverse Requirements Engineering	4
2.2 Large Language Models im Software Engineering	4
2.3 Legacy-Modernisierung und Stand der Forschung	4
3 Fallstudie c-entron GmbH (ca. 6 Seiten)	4
3.1 Unternehmenskontext und Legacy-Software	4
3.2 Migrationsstrategie und spezifische Herausforderungen	4
4 Konzeption und methodisches Vorgehen (ca. 12 Seiten)	4
4.1 Forschungsdesign und Vorgehensmodell	4
4.2 Prozessmodell für KI-gestütztes Reverse Requirements Engineering	4
4.3 Technologieauswahl und LLM-Konfiguration	4
4.4 Stakeholder-Einbindung und Datengrundlage	4
5 Prototypische Umsetzung (ca. 10 Seiten)	4
5.1 Architektur des LLM-Agenten	4
5.2 Toolchain-Integration	4
5.3 Governance, Datenschutz und IP	4
6 Evaluation (ca. 12 Seiten)	5
6.1 Evaluationskriterien und Messgrößen	5
6.2 Durchführung und Ergebnisse	5
6.3 Qualitative Bewertung durch Experten	5
7 Diskussion (ca. 8 Seiten)	5
7.1 Interpretation der Ergebnisse	5
7.2 Chancen und Grenzen	5
7.3 Implikationen für Forschung und Praxis	5
8 Fazit und Ausblick (ca. 4 Seiten)	5
8.1 Zusammenfassung und Beantwortung der Forschungsfragen	5
8.2 Handlungsempfehlungen für c-entron GmbH	5
8.3 Ausblick und zukünftige Forschung	5
9 Literaturverzeichnis (ca. 3 Seiten)	5
10 Anhang (ca. 6 Seiten)	5
10.1 Interviewleitfäden	5
10.2 Zusätzliches Datenmaterial	5
10.3 Konfigurationsdetails des Prototyps	5

Abstract (ca. 1 Seite)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ergänze hier die Zusammenfassung der Arbeit.

1 Einleitung (ca. 8 Seiten)

1.1 Ausgangssituation und Motivation

In den vergangenen Jahren hat die digitale Transformation mittelständische Softwareanbieter gezwungen, ihre gewachsenen Systeme neu zu bewerten. Besonders ERP-Lösungen, die über Jahrzehnte in Windows-Umgebungen gepflegt wurden, stoßen bei Cloud-, Web- und Mobile-Szenarien an technische sowie organisatorische Grenzen. Dokumentierte Architekturentscheidungen sind selten, implizites Wissen steckt in Source-Control-Systemen oder bei einzelnen Entwickler:innen.

Die c-entron GmbH in Ulm repräsentiert diesen Kontext. Das Unternehmen betreibt seit über zwanzig Jahren eine Windows-basierte ERP-Suite für IT-Systemhäuser. Die Lösung deckt Auftragsabwicklung, Lager, Fakturierung und Projektabrechnung ab, ist aber eng mit der bisherigen Client/Server-Architektur gekoppelt. Kunden fordern inzwischen plattformunabhängige Oberflächen, Self-Service-Funktionen und flexible Betriebsmodelle. Die bestehende Anwendung limitiert Skalierung, Deployment und Benutzerführung, wodurch eine Migration auf eine webbasierte Plattform zwingend erforderlich wird.

Parallel dazu hat sich ein neues Instrumentarium etabliert. Large Language Models wie Chat GPT-5 oder Claude.ai können große durch agentische CLIs (Codex, Claude Code) große Mengen an Quellcode analysieren, Muster erkennen und textuell beschreiben. Damit entsteht die Chance, fehlende Anforderungsdokumentationen zumindest teilweise aus dem Code heraus zu rekonstruieren. Die praktische Nutzung dieses Potenzials ist bislang kaum erforscht – insbesondere nicht in mittelständischen Legacy-Projekten. Diese Arbeit adressiert genau diese Lücke und untersucht, wie KI-gestützte Verfahren für eine systematische Anforderungsextraktion eingesetzt werden können.

1.2 Problemstellung

Im Projektumfeld der c-entron GmbH fehlen strukturierte Requirements für die bestehende ERP-Lösung. Die Analyse der Legacy-Codebasis ist zeitintensiv, personengebunden und anfällig für Auslassungen. Daraus ergeben sich mehrere Risiken:

- Re-Implementationsfehler: Edge Cases, Workarounds und kundenindividuelle Anpassungen sind nur im Code sichtbar. Ohne vollständige Erfassung drohen Funktionsverluste nach der Migration.
- Technische Schuld: Entwickler:innen investieren viel Zeit in das Verständnis historischer Strukturen, statt aktiv an der neuen Plattform zu arbeiten. Veraltete Muster werden unreflektiert übernommen.
- Implizites Wissen: Domänenwissen liegt bei wenigen langjährigen Mitarbeitenden. Personalwechsel führen zu Wissensverlust und Verzögerungen.
- Komplexität der Codebasis: Verschachtelte Abhängigkeiten, unterschiedliche Stile und technologiebedingte Zwänge erschweren eine modulare Anforderungsableitung.
- Fehlende Traceability: Ohne Zuordnung zwischen Code und Geschäftsprozess fehlt die Grundlage für Priorisierung, Testkonzeption und spätere Wartung.

Eine rein manuelle Rekonstruktion aller Anforderungen wäre wirtschaftlich kaum tragbar. Deshalb soll geprüft werden, ob KI-gestützte Verfahren Requirements so extrahieren können, dass sie als belastbare Basis für die Modernisierung dienen.

1.3 Zielsetzung

Diese Arbeit verfolgt das Ziel, ein vollständiges Vorgehen für KI-gestütztes Reverse Requirements Engineering im Umfeld eines mittelständischen ERP-Herstellers zu entwickeln und zu bewerten. Die Teilziele lauten:

- Entwicklung eines Prozessmodells, das Vorbereitung, Analyse, Validierung und Übergabe strukturiert.
- Evaluation aktueller LLMs hinsichtlich Kontextfenster, Codeverständnis, Steuerbarkeit, Kosten und Datenschutz.
- Prototypische Umsetzung eines Agenten, der Quellcode verarbeitet, Requirements formuliert und Traceability-Informationen hinterlegt.
- Integration von Stakeholder-Wissen durch Interviews, um nicht direkt aus dem Code ableitbare Anforderungen zu ergänzen.
- Definition eines Evaluationsrahmens mit quantitativen und qualitativen Kriterien (Vollständigkeit, Verständlichkeit, Redundanzfreiheit, Aufwandseinsparung).
- Formulierung konkreter Handlungsempfehlungen für die c-entron GmbH sowie Übertragbarkeit auf ähnliche Unternehmen.

1.4 Forschungsleitfragen

Die Zielsetzung wird über vier Forschungsleitfragen strukturiert:

- F1 - Einsatz von LLMs im Reverse Requirements Engineering: Welche Prozessschritte, Steuerungsmechanismen und Kontrollpunkte sind notwendig, um LLMs reproduzierbar einzusetzen?
- F2 - Kombination von KI-Analyse und Stakeholder-Input: Welche funktionalen und nicht-funktionalen Anforderungen lassen sich aus Code extrahieren, und welche Informationen müssen über Interviews ergänzt werden?
- F3 - Qualitätsbewertung der generierten Requirements: Wie beurteilen Fachexperten Vollständigkeit, Verständlichkeit, Nützlichkeit und Aufwandseinsparung der KI-Ergebnisse?
- F4 - Chancen und Grenzen des Ansatzes: Welche Effizienzgewinne sind realistisch, wo liegen technische oder organisatorische Limitierungen, und welche Risiken (z. B. Halluzinationen, Datenschutz) müssen adressiert werden?

1.5 Aufbau der Arbeit

Die Arbeit ist in acht Kapitel gegliedert und folgt dem in den Vorlagen üblichen Aufbau:

1. Einleitung: Kontext, Problemstellung, Ziele und Forschungsfragen.
2. Theoretische Grundlagen: Requirements Engineering, Reverse Engineering, Large Language Models sowie Qualitätssicherungskriterien.
3. Fallstudie c-entron GmbH: Unternehmensprofil, Produktarchitektur, Migrationsdruck und Rahmenbedingungen.
4. Konzeption und methodisches Vorgehen: Prozessmodell, Technologieauswahl, Stakeholder-Einbindung und Datenbasis.
5. Prototypische Umsetzung: Architektur und Funktionsweise des LLM-Agenten sowie Integration in bestehende Toolchains.
6. Evaluation: Vorgehen, Metriken, Ergebnisse und Expertenfeedback.
7. Diskussion: Interpretation der Resultate, Limitationen und Implikationen für Forschung und Praxis.
8. Fazit und Ausblick: Zusammenfassung, Beantwortung der Forschungsfragen und Perspektiven für weitere Arbeiten.

Damit entsteht eine nachvollziehbare Linie von der Ausgangssituation über das Konzept bis zur Validierung.

2 Theoretische Grundlagen (ca. 12 Seiten)

2.1 Requirements Engineering und Reverse Requirements Engineering

Beschreibe etablierte Verfahren, Grundlagen und Best Practices für die Anforderungserhebung und -rückgewinnung.

2.2 Large Language Models im Software Engineering

Charakterisiere relevante LLM-Konzepte, Fähigkeiten und Grenzen für Software- und Requirements-Analysen.

2.3 Legacy-Modernisierung und Stand der Forschung

Analysiere regulatorische, sicherheitsrelevante und organisatorische Aspekte sowie aktuelle Studien zur Modernisierung.

3 Fallstudie c-entron GmbH (ca. 6 Seiten)

3.1 Unternehmenskontext und Legacy-Software

3.2 Migrationsstrategie und spezifische Herausforderungen

4 Konzeption und methodisches Vorgehen (ca. 12 Seiten)

4.1 Forschungsdesign und Vorgehensmodell

Verbinde Literaturrecherche, Technologieevaluation und Interviews in einem konsistenten Design.

4.2 Prozessmodell für KI-gestütztes Reverse Requirements Engineering

Skizziere Phasen, Rollen und Kontrollpunkte für einen reproduzierbaren Analyse-Workflow.

4.3 Technologieauswahl und LLM-Konfiguration

Dokumentiere Kriterien, Modellwahl und Evaluationsschritte.

4.4 Stakeholder-Einbindung und Datengrundlage

Beschreibe Datenquellen, Interviewleitfäden und Validierungsworkshops.

5 Prototypische Umsetzung (ca. 10 Seiten)

5.1 Architektur des LLM-Agenten

Skizziere die Komponenten, Interaktionsabläufe und Traceability-Konzepte.

5.2 Toolchain-Integration

Bewerte die Einbindung in bestehende Systeme (z. B. Jira, Confluence).

5.3 Governance, Datenschutz und IP

Dokumentiere Maßnahmen zur Sicherstellung von Compliance.

6 Evaluation (ca. 12 Seiten)

6.1 Evaluationskriterien und Messgrößen

Beschreibe Vollständigkeit, Verständlichkeit, Redundanzfreiheit, Stakeholder-Alignment und Aufwandsreduktion.

6.2 Durchführung und Ergebnisse

Erläutere Messaufbau, Workshop-Design, Datenauswertung und die wichtigsten Resultate.

6.3 Qualitative Bewertung durch Experten

Fasse Feedback, Einschätzungen und offene Punkte aus den Experteninterviews zusammen.

7 Diskussion (ca. 8 Seiten)

7.1 Interpretation der Ergebnisse

Vergleiche die Evaluation mit den Forschungsleitfragen.

7.2 Chancen und Grenzen

Diskutiere Potenziale und Limitationen des KI-gestützten Ansatzes.

7.3 Implikationen für Forschung und Praxis

Leitfäden für Unternehmen und offene Forschungsfragen.

8 Fazit und Ausblick (ca. 4 Seiten)

8.1 Zusammenfassung und Beantwortung der Forschungsfragen

Fasse die wichtigsten Erkenntnisse zusammen und beantworte die Leitfragen.

8.2 Handlungsempfehlungen für c-entron GmbH

Formuliere konkrete nächste Schritte für die c-entron GmbH.

8.3 Ausblick und zukünftige Forschung

Zeige zukünftige Forschungsschwerpunkte und Weiterentwicklungsmöglichkeiten des Prototyps.

9 Literaturverzeichnis (ca. 3 Seiten)

Ergänze hier das automatisch generierte Literaturverzeichnis.

10 Anhang (ca. 6 Seiten)

10.1 Interviewleitfäden

10.2 Zusätzliches Datenmaterial

10.3 Konfigurationsdetails des Prototyps