

Einleitung (ca. 8 Seiten)

Ausgangssituation und Motivation

In den vergangenen Jahren hat die digitale Transformation mittelständische Softwareanbieter gezwungen, ihre Produkte neu zu bewerten. Betroffen sind vor allem Systeme die über lange Jahre nur in Windows-Umgebungen vertrieben wurden. Diese stoßen bei Cloud-, Web- und Mobile-Szenarien an technische sowie organisatorische Grenzen und fallen in der technologischen Schuld immer weiter Zurück. Eine technologische Weiterentwicklung ist nicht möglich und an einer Neuentwicklung führt oft kein Weg vorbei. Dokumentierte Anforderungen und Code sind allerdings selten. Das meiste Wissen steckt implizit im Code oder in der Köpfen der verbleibenden Entwickler.

Die c-entron GmbH in Ulm ist von diesem Szenario voll Betroffen. 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 erwarten zwischenzeitlich aber plattformunabhängige Oberflächen, Self-Service-Funktionen und flexible Betriebsmodelle wie z.B. SaaS (Software as a Service). Die bestehende Anwendung ist aber in ihrer Skalierung, Deployment und Abrechnung limitiert. Eine Migration auf eine webbasierte Plattform ist somit zwingend erforderlich.

Eine einfache Neuimplementierung auf Basis vorhandener Anforderungen oder Code Dokumentation ist aber aus oben geschilderten Gründen nicht einfach möglich. Die Herausforderung die sich stellt ist mit möglichst geringem Aufwand eine vollständige Beschreibung für ein vollständiges ERP System zu erarbeiten. Eine Manuelle Auswertung des Codes oder Oberfläche auf Funktionalitäten ist aufgrund der extremen Komplexität, geschuldet der langjährigen Weiterentwicklung, nur mit sehr hohem Personalaufwand möglich und daher nicht realisierbar.

In den letzten Jahren hat sich hierzu nun ein neues Instrumentarium etabliert. Large Language Models wie Chat GPT-5 oder Claude.ai können durch agentische CLIs (Codex, Claude Code) große Mengen an Quellcode analysieren, Anforderungen erarbeiten 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. Diese Arbeit adressiert dies und untersucht, wie KI-gestützte Verfahren für eine systematische Anforderungsanalyse eingesetzt werden können.

Problemstellung

Für das ERP Software Produkt der c-entron fehlen strukturierte und dokumentierte Requirements. Die Analyse der bestehenden Codebasis ist zeitintensiv, ressourcenintensiv und anfällig für Insel- und Metawissen. 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. Zeitgleich sind Workaround of Symptom einer mangelhaften Erfassung der Anforderungen bei originalen Implementierung und ein Zeichen fehlender Weitsicht.
- Technische Schuld: Entwickler investieren viel Zeit in das Verständnis historischer Strukturen, statt aktiv an der neuen Plattform zu arbeiten. Veralterte Muster werden unreflektiert übernommen. Neue Mitarbeiter sind auch nicht bewandt in alten Technologien und es fehlt das Verständnis für historische Zwänge und Zusammenhänge.
- Implizites Wissen: Domänenwissen liegt bei wenigen langjährigen Mitarbeitenden. Personalwechsel führen zu Wissensverlust und Verzögerungen. Gleichzeitig führt die langjährige Arbeit mit dem bestehenden System zu eingeschränkter Offenheit beim Design neuer Lösungen ("Das haben wir schon immer so umgesetzt").

- 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. Große Teile des Codes sind auch generisch und lassen sich nicht einer konkreten Anforderung zuordnen, wie zum Beispiel das anzeigen einer Tabelle. Konkrete Datenflüsse lassen sich hier nur am laufenden System beobachten was die Analyse nochmal um eine Größenordnung Ressourcenintensiver macht.

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.

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.
- Durchführung und Vergleich von drei Claude-Code basierten Versuchen mit unterschiedlicher Tooling-Tiefe (Prompt-only, Agenten, Agenten+MCP).
- Definition eines Evaluationsrahmens mit quantitativen und qualitativen Kriterien (Vollständigkeit, Verständlichkeit, Redundanzfreiheit, Aufwandseinsparung).
- Integration von Stakeholder-Wissen durch Interviews, um die Qualität der KI ergebnisse zu bewerten und nicht direkt aus dem Code ableitbare Anforderungen zu ergänzen .

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?

Aufbau der Arbeit

Die Arbeit ist in acht Kapitel gegliedert:

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. Ergebnisse: Vollständige Ergebnisdarstellung der drei Versuche inkl. Artefaktlisten und beispielhafter Requirements/Use Cases aus den Ergebnisverzeichnissen.
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.