Kapitel 1 - Motivation zum experimentellen Prototyping

1. Kurzüberblick zum Projekt TAHAI

Beim Projekt TAHAI (TrustAdHocAI) handelt sich um ein durch das Institut für angewandte Forschung Berlin (IFAF) gefördertes Vorhaben mit einem starken Praxisbezug. Realisiert wurde das Projekt im Zeitraum von März 2023 bis Mai 2025, wobei vielfältige Projektpartner aus Wissenschaft, Industrie und öffentlicher Verwaltung beteiligt waren. Neben der Projektführerschaft durch die Hochschule für Wirtschaft und Recht (HWR) Berlin und die Hochschule für Technik und Wirtschaft (HTW) Berlin war als weiterer Forschungspartner das Fraunhofer-Institut für Experimentelles Software Engineering (IESE) in Kaiserslautern beteiligt. Darüber hinaus waren die folgenden Berliner Praxispartner involviert: die DB Station & Service AG, die Bundespolizei, die AURISCON GmbH, das Landesforstamt Berlin und die Bundes-Arbeitsgemeinschaft für Familien-Mediation (BAFM) e.V.

Aus inhaltlicher Sicht verfolgte das Projekt die folgende Zielstellung (siehe auch Anlage A):

> *„Das Projekt TAHAI beschäftigt sich mit einer vereinfachten und vor allem fachgetriebenen Implementierung von Anwendungsszenarien der künstlichen Intelligenz. Konkrete Szenarien beziehen sich auf die Mediationsforschung, die Forstwirtschaft und die Eisenbahninfrastruktur. Entsprechende Tests sollen Aufschlüsse hinsichtlich benötigter Prozesse im Software Engineering geben, aber auch Ansätze zur Bewertung der Vertrauenswürdigkeit liefern.“*[1]

Insgesamt wurden im Projekt TAHAI die folgenden Arbeitspakete bearbeitet:

- Implementierung eines Wikis als Informationsdrehscheibe zum Projekt.
- Generischer Ansatz zur fachgetriebenen Identifikation von KI-Szenarien.
- Bewertungsansatz für die bei KI-Lösungen eingesetzten Datenquellen.
- Auswahlkriterien für den entwicklerorientierten Einsatz von KI-APIs.
- Analyse der Möglichkeiten zum automatischen Testen von KI-APIs.
- Implementierung fachgetriebener experimenteller KI-Prototypen.

[1] Schmietendorf, A.: TAHAI – Vertrauen in fachgetriebene Ad-Hoc-KI-Lösungen (TrustAdHocAI), Exposé zum Projektstatus, Institut für angewandte Forschung Berlin (IFAF Berlin), Juni 2024

- Risiko- und Robustheitsanalyse für cloudbasierte KI-Lösungen.
- Evaluierungs- und Zertifizierungsmöglichkeiten einer KI-Lösung.
- Rechtliche Aspekte bzw. Einhaltung von Compliance-Vorgaben.
- Ethische Aspekte bereitgestellter KI-Analysen.

Die Umsetzung des experimentellen Prototypings reflektiert nur einen Ausschnitt der vielfältigen Projektaufgaben. Aus Sicht der Transferleistung zwischen Forschung und Praxis handelt es sich dabei allerdings um das ingenieurwissenschaftliche Herzstück des Projekts. Im Sinne der Validation galt es hier die Potentiale, Grenzen und Risiken eines vereinfachten KI-Einsatzes (KI-APIs in Kombination mit High- und Low-Code-Ansätzen) gegenüber den Stakeholdern des Projekts bzw. potentiell Interessierten zu verdeutlichen.

2. Ziele des experimentellen Prototypings

Bevor ein Überblick zu den im Projekt TAHAI realisierten Prototypen gegeben wird, erfolgt zunächst eine grundsätzliche Auseinandersetzung mit den Zielen eines experimentellen Prototypings im Software Engineering, mit dessen Hilfe eine „Brücke“ zwischen Theorie und Praxis im Sinne eines Technologietransfers etabliert werden sollte. Folgende Zielstellungen wurden insbesondere verfolgt:

- Realisierung von Machbarkeitsstudien zur Überprüfung der Umsetzbarkeit methodischer bzw. technischer KI-Ansätze.
- Vergleich bzw. Benchmarking unterschiedlicher KI-Modelle hinsichtlich definierter Qualitätsmerkmale anhand realer Problemstellungen.
- Frühzeitiges Erkennen von domänenspezifischen, konzeptionellen aber auch technischen Probleme mit Hilfe forschungsorientierter Prototypen.
- Hypothesen und wissenschaftliche Konzepte lassen sich mit Hilfe praxisorientierter Problemstellung einer Validation unterziehen.
- Prüfung der Eignung vorhandener Daten bzw. Unterstützung des notwendigen Trainings eingesetzter KI-Modelle mit Prototypen.
- Einbindung der späteren Nutzer zur Gewinnung fachlicher Anforderungen und Gewährleistung eines zeitnahen Feedbacks.
- Etablierung von Demonstratoren die sowohl im wissenschaftlichen als auch industriellen Kontext zur Diskussion gestellt werden.
- Diskussion rechtlicher und ethischer Fragen anhand realer und durch die Prototypen nachvollziehbarer Anwendungsszenarien.
- Aufzeigen der Möglichkeiten und Grenzen zur KI-Erklärbarkeit und dem resultierenden Vertrauen in entsprechende Lösungen.

Mit Hilfe einer fachübergreifenden und interdisziplinären Betrachtung der entwickelten Prototypen sollte letztlich die Grundlage für eine sachliche Auseinandersetzung mit den Möglichkeiten potentieller KI-Anwendungsszenarien jenseits firmenspezifisch gehypter Produktansätze geschaffen werden.

3. Einordnung der Prototypen des Projekts TAHAI

Allgemein fokussierten die implementierten Prototypen, projektintern auch als kontrollierte Experimente bezeichnet, die folgenden Sichtweisen (Klassifikation a und b) auf das Software Engineering:

a. KI-Lösungen zur Unterstützung des Software Engineerings – Kapitel 2
b. Software Engineering für fachorientierte KI-Lösungen – Kapitel 2-6

Die Reihenfolge der Kapitel korrespondiert mit dem Zeitpunkt der Umsetzung im Projekt, d.h. im Kapitel 6 findet sich dem entsprechend der zuletzt implementierte Prototyp.

- Kapitel 2 (Klasse a) – der vorgestellte Prototyp fokussiert auf ein Large Language Model (kurz LLM) unterstütztes Rapid Prototyping, was im weiteren Sinn als Vorläufer Low-Code-orientierter Softwareentwicklungsansätze interpretiert werden kann. Im Mittelpunkt stand neben einer Aufnahme des wissenschaftlichen Sachstands der Vergleich dreier LLMs hinsichtlich signifikanter Leistungsmerkmale. Aus fachlicher Sicht galt das Interesses insbesondere der Bewertung mit Hilfe eingesetzter LLMs generierter Text- und Quellcode-Fragmente.
- Kapitel 3 (Klasse b) – hier galt es mit Hilfe von Prototypen die Möglichkeiten einer KI-basierten Anonymisierung von transkribierten Mediationssitzungen zu bewerten. Derartige Transkripte implizieren zwangsläufig den Umgang mit personenbezogenen Daten, weshalb eine qualitätsgesicherte Anonymisierung unerlässlich ist. Neben der Klärung der Grundlagen und des Bezugsbereichs der domänenspezifischen Anonymisierung erfolgte die prototypische Implementierung sowohl auf der Basis zweier NER_2-Frameworks als auch mit Hilfe zweier Produkte.
- Kapitel 4 (Klasse b) – die prototypischen Tests dieses Kapitels setzen unmittelbar auf den Ergebnissen des vorhergehenden Kapitels auf. Die anonymisierten Transkripte wurden dabei mit Hilfe unterschiedlicher Analysemethoden verarbeitet. Im Detail erfolgten statistische Auswertungen (Einsatz Regular Expressions) zur Ermittlung der prozentualen Verteilung

[2] Named Entity Recognition

von Gesprächsanteilen, Sentimentanalysen (Test von unterschiedlichen 5 LLMs) zur Erfassung von Stimmungen im Verlauf durchgeführter Mediationssitzungen und schließlich die fachlich orientierte Befragung der Transkripte auf der Grundlage von RAG (Retrieval-Augmented Generation) bzw. Agentic RAG, jeweils lokal ausgeführte Lösungen.

- Kapitel 5 (Klasse b) – auch dieser Prototyp knüpft direkt an die inhaltliche Ausrichtung der beiden vorhergehenden Kapitel an. Nachdem die im Kaptiel 4 beschriebenen Prototypen mit Hilfe der Entwicklungsumgebung Juniper Notebook (vgl. https://jupyter.org) und dem entsprechend in Pythoncode erfolgten, sollten hier die Möglichkeiten zur Implementierung von statistischen- und Sentiment-Analysen mit Hilfe einer als Produkt vertriebenen Low-Code-Applikation Plattform untersucht wurden. Verglichen mit den Python-basierten Prototypen wurden insbesondere die erreichbare Produktivität, die interne Struktur der Applikation und die Möglichkeiten und Grenzen einer primär fachgetriebenen Softwareentwicklung (allg. Citizen Development).
- Kapitel 6 (Klasse b) – während die vorhergehenden Prototypen textorientierte Datenquellen verwendeten, galt das Interesse bei diesem Prototyp einer KI-basierten Bild- und Videoverarbeitung. Dieser neue Anwendungsbereich fokussierte auf die Erkennung von Graffitis (allgemein Vandalismus) an eisenbahntechnischen Infrastrukturen (hier konkret am Bahnhof Berlin Südkreuz). Neben der drehbuchbasierten Erzeugung entsprechender Videosequenzen erfolgte die Analyse selbiger mit Hilfe von vier unterschiedlichen KI-gestützten Analyseverfahren. Entsprechende Bewertungen bezogen sich u.a. auf nicht erkannte, richtig erkannte und falsch erkannte Vandalismusvorfälle. Ebenso erfolgte ein Vergleich der einhergehenden Präzision und der erreichbaren Verarbeitungseffizienz. Darüber hinaus erfolgte im Zusammenhang mit diesem Prototyp eine umfängliche Auseinandersetzung mit Rechtsfragen der KI-gestützten Videoanalyse auf Bahnhöfen der Deutschen Bahn AG. Ausgewählte Aspekte dazu finden sich in Anlage 4 zu Kapitel 6.

Die prototypischen Implementierungen für den Bereich der Forstwirtschaft (Partner: Landesforstamt Berlin), verantwortet durch die HTW Berlin, sind nicht Gegenstand dieser Monografie. Im Detail setzten sich diese mit der drohnengestützten Totholzerkennung unter Einsatz von Computer Vision Modellen auseinander. Ein kurzer Überblick zu den einhergehenden Herausforderungen und der implementierten Analyseplattform (Forrest Analyzer) findet sich in den Anlagen dieses Buchs. Darüber hinaus finden sich entsprechende Erläuterungen dazu auch in der

Gesamtdokumentation[3] des Projekts, welche ebenfalls beim Logos-Verlag publiziert wurde.

4. Organisation und Prozess zur Implementierung

Die vorgestellten Prototypen konnten nur auf der Basis eines interdisziplinär zusammengesetzten Teams erstellt werden. Im Detail galt es die folgenden Kompetenzbereiche im Diskurs der hier beschriebenen Prototypen abzudecken, in den Klammern finden sich jeweils die primär agierenden Projektpartner:

- Bereitstellung domänenspezifischer Kenntnisse, hier im Zusammenhang mit dem Anwendungsgebiet der Professionsforschung im Diskurs Mediation (*BAFM e.V.*) bzw. der Vandalismusbekämpfung an Bahnhöfen (*DB InfraGO AG* und *Bundespolizei*).
- Fundierte Kenntnisse im Software Engineering, d.h. es galt Programmiersprachen wie Java, JavaScript und Python im Diskurs moderner Entwicklungsumgebungen sicher einzusetzen. Darüber hinaus bedurfte es des Umgangs mit KI-Modellen bzw. KI-APIs (*HWR Berlin*).
- Experten im Bereich des KI-basierten Software Engineerings bzw. des Software Engineerings für KI-Lösungen. In diesem Zusammenhang galt es u.a. Fragen der Erklärbarkeit zu bearbeiten bzw. verbleibende Risiken des KI-Einsatzes einzuschätzen (*Fraunhofer IESE*).
- Experten bezüglich der Berücksichtigung rechtlicher Aspekte, wobei es sich insbesondere um Fragen der Haftung, AGBs aber auch Daten- und Immaterialgüterschutz handelt. Ebenso galt es ethische Aspekte, wie z.B. die Diskriminierungsfreiheit, zur berücksichtigen (*HTW Berlin*).
- Experten zur Bewertung der KI-Sicherheit, im Detail gilt es mögliche Angriffsvektoren bzw. -arten (vgl. z.B. OWASP-KI-Angriffsvektoren[4]) zu identifizieren und die mit konkreten Anwendungsszenarien einhergehenden Risiken zu bewerten (*AURISCOM GmbH*).

Die Implementierung der Prototypen erfolgte auf der Grundlage eines agilen Vorgehens (u.a. Iterationen/Inkremente). Damit einher geht die Möglichkeit eines zeitnahen Feedbacks und die Unterstützung einer lernenden Organisation (originäres Selbstverständnis im Projekt TAHAI).

---

[3] Schmietendorf, A.; Rodner, E.; Schnieders, R.: Zusammenfassende Darstellung der Ergebnisse des Forschungsprojekts TAHAI (TrustAdHocAI), 120 Seiten, Monografie, Logos-Verlag, Berlin, ISBN 978-3-8325-5906-x

[4] OWASP Top 10 für LLM & Generative KI (2025), Quelle: https://genai.owasp.org/resource/die-owasp-top-10-fur-llm-generative-ki-2025/ (letzter Zugriff 30.09.2025)

Im Detail waren es unzählige Abstimmungsrunden zur sukzessive Ausgestaltung der funktionalen Umfänge und zu testenden Aspekte. Auf internen und schließlich auch öffentlichen Workshops wurden die gewonnenen Ergebnisse schließlich einer kritischen Diskussion unterzogen.

5. Aktualität der Ergebnisse

Im Zusammenhang mit dem Einsatz hochgradig innovativer KI-Ansätze gilt es fortwährend aktuelle Technologietrends zu beobachten und ggf. auch zu berücksichtigen. Gerade bei technologisch orientierten Prototypen, wie im Kapitel 2 beschrieben, kann die Aktualität der gewonnenen Ergebnisse nur für einen kurzen Zeitraum gewährleistet werden. Aufgrund der stetig am Markt neu platzierten Methoden, Techniken und Tools (hier speziell neue Versionen von KI-Modellen) werden gewonnene Erkenntnisse schnell obsolet. Ebenso werden technisch orientierte Problembereiche schnell durch existierende Communities bzw. potentielle Anbieter einer Lösung zugeführt.

Anders verhält es sich bei den eher domänenspezifisch ausgerichteten Prototypen der weiteren Kapitel. Hier gilt die Bewertung der Frage, inwieweit sich fachlich spezifizierte Problemstellungen mit Hilfe vorgefertigter KI-Komponenten (APIs, Frameworks, Tools) umsetzen lassen. Dem entsprechend sollten gewonnene Erkenntnisse längerfristig von Interesse sein. Typische Sachverhalte beziehen sich auf:

- Identifikation und Bewertung einsetzbarer Datenquellen.
- Prozess zur fachgetriebenen Auswahl nutzbarer KI-Modelle.
- Rahmenbedingungen zum Trainieren von KI-Modellen.
- Generische Bewertungs- und Testansätze für KI-Lösungen.
- Risikoeinschätzungen im Diskurs der fachlichen Anwendungsdomäne.

In jedem Fall lassen sich aus den hier aufgezeigten Untersuchungen Erfahrungen im Umgang mit prozessualen und organisatorischen Sachverhalten beim Prototyping gewinnen. Auch die jeweils verwendeten Bewertungsmodelle sollten als Grundlage für aufsetzende Untersuchungen weiterverwendet werden können. Für KI-unerfahrene Leser bietet sich darüber hinaus ein guter Überblick (Aspekt der Weiterbildung) zu möglichen KI-Szenarien im Zusammenhang mit domänenspezifischen Anwendungsgebieten.

6. Dank

Ein besonderer Dank gilt den Autoren der Monografie. Ohne eure intrinsische Motivation wäre es nie zu den hier veröffentlichen Projektergebnissen gekommen! Ein

Dank gilt aber auch den industriellen, öffentlichen und wissenschaftlichen Projektbeteiligten. Eure konstruktive, kritische und jederzeit partnerschaftliche Mitwirkung hat ganz erheblich zur Reife der implementierten Prototypen beigetragen!

Kapitel 2 - Rapid Prototyping - Supported by Large Language Models

1. Motivation und Überblick

Die Ursprünge des Rapid Prototypings können bereits auf eine mehr als 30-jährige Historie zurückblicken. Im Mittelpunkt des damals als Rapid Application Development (kurz RAD) bezeichneten Ansatzes standen vor allem die modellgetriebene Softwareentwicklung für primär datenbankorientierte Anwendungen. Heute bietet sich mit dem Rapid Prototyping ein methodisch reifer und vor allem agiler Ansatz zur Softwareentwicklung. Mit Hilfe schnell, interaktiv und zunehmend grafisch erstellter Prototypen lassen sich zeitnah Entwickler- und Nutzerfeedbacks gewinnen und das fachliche und technische Design der Anwendung sukzessive verfeinern.

Das Aufkommen großer Sprachmodelle (Large Language Models, LLMs) hat das Potenzial, das Rapid Prototyping abermals grundlegend zu verändern. Entsprechende Möglichkeiten beziehen sich auf die Generierung von Text, Quellcodes aber auch Architekturpattern. Im Rahmen dieses Kapitels soll auf die Möglichkeiten des Einsatzes von LLMs zur Unterstützung des Rapid Prototypings eingegangen werden. Dafür sollten verschiedene auf LLMs basierende Assistenten miteinander verglichen werden und ein Prototyp für einen spezifischen Anwendungsfall entwickelt werden. Allgemeines Ziel war es, potentielle Herausforderungen und Grenzen eines LLM-gestützten Entwicklungsprozesses zu verdeutlichen.

2. Verfolgte Forschungsfragen

Große Sprachmodelle (Large Language Models, LLMs) können die agile Entwicklung leistungsfähiger und ggf. auch alternativer Prototypen massiv unterstützen. Aktuelle Herausforderungen bestehen u.a. bei der nahtlosen Integration dieser Modelle in komplexe und ggf. DevOps-orientierte Entwicklungsumgebungen. Kapitel 2 beschäftigt sich mit dem Potenzial, aber auch mit den Hürden und potentiellen Risiken beim Einsatz von LLMs im Prototyping beleuchtet.

- *Forschungsfrage 1 (FQ1):* Wie können große Sprachmodelle (LLMs) effektiv genutzt werden, um den Prozess des Rapid Prototypings in der Softwareentwicklung – insbesondere in den Bereichen Codegenerierung, Texterstellung und Refactoring – zu verbessern?
- *Forschungsfrage 2 (FQ2):* Welche Herausforderungen und Einschränkungen sind mit dem Einsatz von LLMs im Rapid Prototyping verbunden, und

welche potenziellen Lösungen können zur Überwindung dieser Einschränkungen verfolgt werden?

Neben den Forschungsfragen galt es sich mit den grundlegenden Rahmenbedingungen eines LLM-Ansatzes innerhalb der Softwareentwicklung vertraut zu machen. In diesem Zusammenhang sollte die prototypische Nutzung verschiedener LLMs einen benötigten Kompetenzaufbau im Projekt unterstützen.

Selbstverständlich gibt es weitere Herausforderungen, die mit dem LLM-Einsatz innerhalb der Softwareentwicklung einhergehen, welche allerdings nicht Gegenstand der folgenden Abschnitte sind. Beispiele beziehen sich auch auf die Qualitätssicherung mit Hilfe von LLMs erzeugter Text- und Quellcodefragmente. Ebenso gilt es Fragen des Urheberrechts, der Compliance, der IT-Sicherheit aber auch ethische Aspekte zu berücksichtigen.

3. Existierende Arbeiten

Rapid Prototyping (RP) bezeichnet die schnelle Erstellung vereinfachter Softwaremodelle. Ziel ist es Nutzerbedürfnisse besser zu verstehen und ein zeitnahes Feedback einzuholen, um letztendlich kostspielige Nacharbeiten zu vermeiden. Es ermöglicht die frühzeitige Identifikation potenzieller Probleme und verbessert die Benutzererfahrung durch das Experimentieren mit Benutzeroberflächen und Nutzerabläufen. RP verkürzt den Zyklus von der fachlich orientierten Idee (Was soll umgesetzt werden?) bis zum Feedback und unterstützt so die iterative Softwareentwicklung. Zu den Prototyp-Typen zählen „Throwaway“-Prototypen für spezifische Ideen, „Evolutionary“-Prototypen für eine schrittweise Entwicklung, „horizontale“ Prototypen für Benutzeroberflächen sowie „vertikale“ Prototypen zur Überprüfung technischer Machbarkeit. Werkzeuge wie Figma, Balsamiq und Skriptsprachen unterstützen den RP-Prozess und harmonieren gut mit agilen Methoden; insbesondere profitieren sie von Fortschritten in Low-Code-Plattformen.

Ein Large Language Model (LLM) ist ein neuronales Netz mit Milliarden von Parametern, das auf der Grundlage großer Textmengen mittels selbstüberwachtem oder teilüberwachtem Lernen trainiert wird. Seit etwa 2018 ermöglichen LLMs herausragende Leistungen bei Aufgaben wie Texterkennung, Zusammenfassung, Übersetzung, Vorhersage und Generierung. Sie basieren auf Deep Learning – einem Teilbereich der künstlichen Intelligenz und des maschinellen Lernens – und werden mithilfe erheblicher Rechenleistung vortrainiert. LLMs gelten als eine der erfolgreichsten Anwendungen des Transformer-Modells, wobei es sich hier um eine Deep-Learning-Architektur mit integrierten Aufmerksamkeitsmechanismus handelt. (vgl. [11])