IU Internationale Hochschule – Fernstudium

**Exposé** (01/2025)

Name: Matrikelnummer:

Evsin Rahmiev 32105477

Betreuungsperson: Studiengang:

Prof. Dr. Thorsten Fröhlich Wirtschaftsinformatik B. Sc.

Vorläufiger Titel:

Anwendungsentwicklung mittels Low-Code-Entwicklung im Vergleich zu High-Code-Entwicklung

Forschungsfrage/n:

1) Kann eine Anwendung zum Sprachenlernen besser langfristig skaliert werden, wenn sie über Low-Code oder über High-Code entwickelt wurde?

2) Hat eine Low-Code- oder eine High-Code-Anwendung eine höhere Anpassungsfähigkeit?

3) Über welche der beiden Entwicklungsmöglichkeiten sind bessere Integrationsmöglichkeiten mit anderen Systemen vorhanden?

Problem und Zielstellung:

Die Softwareentwicklungsfirma QuantumCraft hat vor, eine innovative Anwendung zum Lernen von Fremdsprachen zu entwickeln und zu veröffentlichen. Die Anwendung soll „LexiCode“ benannt werden und Inhalte zum Lernen der deutschen Sprache anbieten, bevor sie dann auch für das Lernen von weiteren Sprachen weiterentwickelt wird. Da QuantumCraft über ein teilweise junges Entwicklungsteam verfügt, das noch nicht viel Erfahrung im Bereich der Softwareentwicklung sammeln konnte, stellt sich die Frage, über welche Entwicklungstechnologien das Team eine vollständige Anwendung entwickeln soll. Der Abteilungsleiter bietet die Option an, LexiCode über Low-Code zu entwickeln, damit die Anwendung einerseits schneller umgesetzt werden kann und andererseits die jungen Entwickler bei der kompletten Umsetzung des Projekts mitmachen können.

Methodik:

Damit entschieden werden kann, ob LexiCode über High- oder Low-Code entwickelt werden soll, wird das Entwicklungsteam in zwei Unterteams geteilt: Team A und Team B. Beide Teams haben die Aufgabe, Lösungen für die Implementierung von wichtigen Anwendungsaspekten wie Design, Datenspeicherung, Schnittstellenanbindung, Sicherheit und Skalierbarkeit zu planen und entwerfen. Team A soll die Lösungen über die Low-Code Entwicklungsplatform „Zoho Creator“ umsetzen und Team B über die Standard Web-Entwicklungstechnologien HTML, CSS und JavaScript. Die Entwickler werden den Teams so zugewiesen, dass in beiden Teams sowohl unerfahrene als auch eher erfahrene Entwickler vorhanden sind, damit die Unterschiede im Erfahrungsgrad ausgeglichen werden. Die Entwickler von Team A müssen sich selbstverständlich auch in Zoho Creator einarbeiten, damit sie ihre Aufgabe erfüllen können. Beide Teams können je nach Wahl entweder echten Programmcode zu den Systemkomponenten umsetzen oder Pseudocode vorbereiten.

Struktur:

Team A und Team B haben eine Bearbeitungszeit von zwei Monaten, um die Planung bzw. Implementierung von vier Hauptaspekten von LexiCode durchzuführen. Diese sind folgende:

1. UX/UI-Design

2. Datenspeicherung

3. Skalierbarkeit und Performance

4. Sicherheit und Wartung.

Beide Teams müssen nach Ablauf der Bearbeitungsfrist wenigstens in der darlegen können, wie die aufgezählten vier Systembereiche mit der für ihr Team vorgesehene Technologie (High-Code oder Low-Code) implementiert werden können. Der Zweck hinter dieser parallelen Arbeit ist, die beiden Entwicklungsansätze anhand der Ergebnisse zu vergleichen, um herauszufinden, welcher Ansatz der Anwendung eine bessere Anpassungsfähigkeit und höhere Flexibilität, Leistungsfähigkeit und Skalierbarkeit, weniger Einschränkungen und bessere Integrationsmöglichkeiten vorhanden sind.

Vorläufige Leseliste:

Baumgarten, C., Rainer, E. & Stich, S. (2024): Professionelle Softwareentwicklung mit Low Code optimieren – eine Fallstudie. In: *HMD Praxis der Wirtschaftsinformatik*. Springer Verlag.

Brandt-Pook, H. & Kollmeier, R. (2020): *Softwareentwicklung kompakt und verständlich. Wie Softwaresysteme entstehen*. (3. Auflage). Springer Verlag.

Böhler, T. (2023): Software einfacher, flexibler und leichtfüßiger entwickeln. In: *Produktion*. (12):16.

Elshan, E. & Binzer, B. (2024): Mehr als ein Trend?: Wie Low-Code die digitale Transformation unterstützt. In: *HMD Praxis der Wirtschaftsinformatik*. 1070-1087.

Frank, B. (2023): *ERP:* The next big thing: Sind Lösungen auf Low-Code-Basis die Zukunft? In: *IT-Management*. 50-54.

Hensen, U. (2023): Maßgeschnittene Software mittels Low Code. In: *Factory Innovation*. 33-37.

Schreiner, K. (2024): Entwicklungszyklen halbieren mit Low Code. In: *Digital Engineering*. 44-45.

Spierling, D. (2023): Mehr IT-Nachhaltigkeit und Klimaschutz durch Low-Code Development. In: *Wirtschaftsinformatik & Management*. 107–113.

-Hypothese dazu wäre hilfreich

-Welche theoretische Konzepte (z. B. Wirtschaftlichkeitsanalysen)

-Case Study – Low-Code vs High-Code, Kriterien davor festlegen

-in der Einleitung das, was ich machen will

-im Forschungsdesign beschreiben, dass ich ein AB Test machen will (einmal das, und einmal das)

-asnchließend Analyse und Synthese der Ergebnisse

-mit Requirements arbeiten

-zwei Wochen vor der Abgabe dem Herrn Fröhlich die Arbeit schicken, damit er sie durchschauen kann

Feedback 15.04.2025

-Refernces – Insert Caption für überschriften von Tabellen und rechte Maustaste -> Update Field in dem Verzeichnis,

-Refernces – Insert Citation – Author, Title, Journal Name, Year, Show all Bibliography Fields, Language, dann in den Referenzen Update Citaions and Bibliography

-References – Cross References, bei umnumerierung von Tabellen

Keine **empirische** oder **nutzerzentrierte** Validierung der Ergebnisse.

·        **Methodik bleibt stark technisch-konzeptionell** ohne triangulierte Evidenz oder Metriken wie Entwicklungszeit.

·        Die A/B-Studie bleibt **selbstreferenziell** – der Entwickler ist gleichzeitig Evaluator.

* Alles begründen i n Anwendungsbereich und Einschränkungen
* -Subjektive Wertung – auch begründen

-alle Zitationen in den Fußnoten, keine in den Fußnoten

-Editor Mode – grammatikalische Prüfung und Rechtsschreibfehler, Language Tool – kostenfrei, installieren und Text prüfen lassen