Reengineering einer bestehenden High-Code KI-Applikation zu einer Low-Code Lösung

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

IFAF Institut für angewandte Forschung Berlin

Leitung:
Prof. Dr. Andreas Schmietendorf
Hochschule für Wirtschaft und Recht Berlin & Otto-von-Guericke-Universität Magdeburg

Wissenschaftliche Mitarbeiter:
Sandro Hartenstein
Hochschule für Wirtschaft und Recht Berlin

Walter Letzel
Hochschule für Wirtschaft und Recht Berlin

Studentischer Mitarbeiter:
Ben Rymar
Hochschule für Wirtschaft und Recht Berlin

Zielsetzung und Projektvision

Das Reengineering-Projekt transformiert eine bestehende Python-basierte KI-Applikation zur Analyse von Mediationssitzungen in eine Low-Code-Lösung. Im Kern beschäftigt sich diese mit Häufigkeits- und Sentiment-Analysen transkribierter Mediationssitzungen. Im Zentrum steht die Demokratisierung der Anwendungsentwicklung, d.h. Domänenexperten aus Mediation und Soziologie sollen ohne Programmierkenntnisse aktiv am Entwicklungsprozess partizipieren können.

Auswahl der LCAP

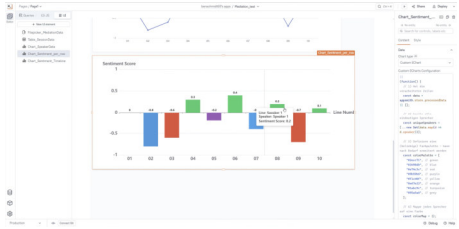
Neben einer exakten Analyse der zu portierenden Anwendung hinsichtlich der abzubildenden Funktionen und der eingesetzten Technologien gilt es eine LCAP Kriterien basiert auszuwählen.

Die Auswahl einer geeigneten Low-Code-Plattform (LCAP) erfolgte anhand projektspezifischer Anforderungen. Zu diesem Zweck wurde ein mehrstufiges Bewertungsverfahren durchgeführt, in dem 21 verschiedene LCAPs anhand zentraler Kriterien (u.a. KI-Funktionalität, Lizenzmodell, Integrationsfähigkeit) analysiert wurden.

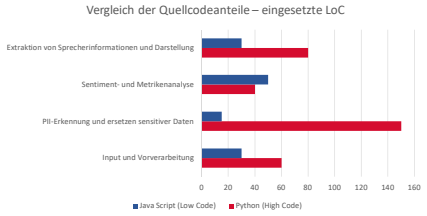
RANG	PLATTFORM	KI-FÄHIGKEITEN (GEWICHTET)	KI-SICHERHEIT UND ZUVERLÄSSIGKEIT (GEWICHTET)	ENTWICKLUNGS- UND INTEGRATIONSFÄHIGKEIT (GEWICHTET)	INTEGRATION APIS (GEWICHTET)	KOSTEN/ NUTZUNGSMODELL (GEWICHTET)	GESAMT
1.	Mendix	24	22	18	13	8	3	89
2.	Appsmith	18	21	19	14	9	9	88
3.	Microsoft Power Apps	24	19	11	11	8	4	87
4.	Amazon Q	23	20	18	10	8	3	87
5.	OutSystems	23	21	18	13	8	3	86
6.	Appian Platform	17	21	19	11	8	4	84
7.	Salesforce Lightning	19	22	16	11	7	4	81
8.	Oracle APEX	23	20	17	12	7	4	80
9.	ServiceNow App Engine	19	20	17	12	7	3	77
10.	Neptune DXP	18	19	16	13	7	4	75
11.	Corvin Studio	18	19	16	13	7	4	74
12.	Apps Script	17	18	12	12	6	3	73
13.	Zoho Creator	17	18	12	12	6	3	72
14.	Budibase	13	19	13	14	8	5	72
15.	NocoDB	13	19	13	14	8	5	72
16.	Knorbase	17	20	17	12	7	2	72
17.	NocoBase	13	19	13	14	7	5	71
18.	Backendless	13	17	13	13	8	4	70
19.	Baserow	13	17	14	14	7	4	69
20.	Adalo	13	17	11	14	7	4	66
21.	Glide	13	15	14	14	6	4	65

Umsetzung in Appsmith

Die Abläufe aus der bestehenden Python-Anwendung konnten in der neuen Umgebung weitgehend erhalten bleiben, indem die bereits bewährten Parameter und Verfahren unverändert übernommen wurden. Während in der High-Code-Lösung Python-Skripte zum Vorbereiten der Quelldaten eingesetzt wurden und die Ergebnisse mit Bibliotheken wie Plotly oder Mat-plotlib visualisiert wurden, ließ sich dieser Prozess in der Low-Code-Plattform über Queries zum Einbinden externer Daten- und Funktionsservices (u.a. REST-APIs), JavaScript-Funktionen und Drag-and-Drop-Widgets abbilden.



Highcode vs Low-Code



Vorteile:
- Erhöhung der Feedbackzyklen (verstärkter fachlicher Input)
- Schnelle Einarbeitung unerfahrener Mitarbeiter (dennoch Entwickler)
- Schnelle, aber vereinfachte KI-Anwendung (begrenzte Anpassbarkeit)

Nachteile:
- Keine direkte Überführung von Python nach Java Script
- Ursprünglich genutzt GPT 3.5, LLAMA 2 und BERT
- Stark vereinfachte Anonymisierung – nur in Bezug auf die Rollen
- Datensicherheit, Vertrauenswürdigkeit und Robustheit der KI-Modelle ungeklärt

[Schmietendorf/Knuth 2024] Schmietendorf, A.; Knuth, M.: Aspekte des Software Engineerings im Diskurs einer Low-Code orientierten Softwareentwicklung, Ausgewählte Ergebnisse des Projekts TAHAI, Logos-Verlag, Berlin

[Kirchhof et al. 2023] Kirchhof, J. C.; Jansen, N.; Rumpe, B.; Wortmann, A.: Navigating the Low-Code Landscape: A Comparison of Development Platforms. In: Proceedings of MODELS. Workshop LowCode, pp. 854 - 862, ACM/IEEE, Oct. 2023.

[Konersmann 2024] Konersmann, M.: Challenges of Low Code PaaS Environments for Future Software Reengineering, in Softwaretechnik-Trends Band 44, Heft 2, Gesellschaft für Informatik e.V., 2024