

Fünf Schlüsselkompetenzen für Low-Code-Entwickler



© AdobeStock/SFIO CRACHO



Autor:
Philipp Erdkönig
Consultant
WEBCON
www.webcon.com/de

Der Markt rund um Low-Code-Technologien wird 2023 um 19,6 Prozent anwachsen, wie das Research- und Beratungsunternehmen Gartner prognostiziert. Der Einsatz von Low-Code-Plattformen soll demnach sogar um 25 Prozent ansteigen, das größte Wachstum verzeichnen dabei Citizen-Automation-Development-Plattformen mit 30,2 Prozent. Unternehmen haben deutlich erkannt, welche Vorteile sie durch die Business Process Automation mithilfe von Plattformen zur Low-Code-Anwendungsentwicklung haben: Denn diese ermöglicht die Entwicklung von Applikationen, ohne eine Zeile Code schreiben zu müssen. Dies soll nicht nur die professionelle Entwicklung vorantreiben, sondern auch die Digitalisierung in Unternehmen durch Mitarbeiter in nicht-technischen Positionen, die sogenannten Citizen Developer. Das Ergebnis ist ein rapider Anstieg der Quantität von Anwendungen – wenn auch deren Qualität oft zu wünschen übrig lässt: So scheitern laut einer in den Vereinigten Staaten von der CIMI Corporation durchgeführten Untersuchung 54 Prozent der Citizen-Development-Projekte innerhalb des ersten Jahres ihrer Laufzeit, weitere 28 Prozent erzielen nur marginale Ergebnisse.

Entwicklungsgrundlagen sind notwendig

Das liegt vor allem an der Art und Weise, wie Citizen Developer Low-Code-Tools nutzen; frei nach dem Motto: Gibt man jemandem einen Hammer, wird er nicht automatisch zum Zimmermann. Es besteht ein erheblicher Unterschied zwischen dem Zusammenbau eines Ikea-Bausatzes und der Anfertigung eines maßgefertigten Schrankes. In anderen Worten: Mit Low-Code-Technologien müssen Nutzer zwar nicht programmieren können, sie müssen aber trotzdem wie ein Entwickler denken. Es gibt schließlich einen Grund, warum man sie „Citizen Developer“ nennt.

Voraussetzungen

Was müssen Low-Code-Benutzer also wissen, um erfolgreich Anwendungen zu entwickeln? Das weiß Philipp Erdkönig, Consultant bei Webcon. Er beschreibt die fünf Schlüsselkompetenzen, die Citizen Developer für die erfolgreiche Low-Code-Anwendungsentwicklung heute benötigen.

1. Das Modellieren der Prozesslogik

Die meisten Low-Code-Anwendungen werden erstellt, um Aktivitäten zu automatisieren oder rationalisieren, die andernfalls manuell durchgeführt werden müssten. Dabei folgen sie der sogenannten Workflow-Logik: Auf einen bestimmten Sachverhalt folgt ein fest definierter nächster Schritt. Low-Code-Entwickler müssen also in der Lage sein, diese Bedingungslogik zu modellieren – im Kopf, auf Papier oder mithilfe eines Grafik-Tools. Sie müssen die entsprechenden Bedingungen prüfen, bevor der Prozess fortgesetzt wird, und dabei das logische Ergebnis des Vorgangs vor Augen haben.

Interessant hierbei: Der Versuch, eine Tätigkeit zu automatisieren, ist oft das erste Mal, dass diese offiziell erfasst wird. Daher muss geprüft werden, was der Vorgang genau beinhaltet und welche Anforderungen sich daraus ergeben.

2. Das Erfassen von Anforderungen

Konzipieren Entwickler eine Anwendung, ermitteln sie zunächst deren Anforderungen. So sollen die Probleme des Unternehmens identifiziert und festgelegt werden, wie der Erfolg aussehen soll. Dabei müssen Developer auch beachten, welche anderen Personen vom Prozess betroffen sind – und wie. Das klingt für viele nach einer überflüssigen Übung, denn die Ziele scheinen offensichtlich. Doch das Auslassen einer Anforderungserfas-

sung kann weitreichende Folgen haben: die Ausweitung des Projektumfangs, nachträgliche Anpassungen oder ein Gefühl der Frustration, wenn eine Low-Code-Anwendung aufgrund der fehlenden Anforderungserfassung ihr Ziel verfehlt.

3. Das Handhaben von Ausnahmen

Bei der Modellierung der Prozesslogik und dem Erfassen von Anforderungen liegt der Schwerpunkt in der Regel auf dem Ablauf des Prozesses. Dabei gibt es jedoch zwangsläufig Ausnahmen, beispielsweise bei der Dateneingabe von ausländischen Adressen oder bei Prozessschritten, wenn Schlüsselpersonen out-of-office sind.

Menschen sind in der Lage, diese Ausnahmen als normalen Geschäftsverlauf zu handhaben - Anwendungen nicht. Daher müssen Entwickler solche Ausnahmefälle antizipieren und automatisieren. Sie müssen Annahmen also im Vorfeld sorgfältig prüfen und klären, wo Daten oder Prozesse außerhalb fester Parameter liegen könnten. Anschließend legen sie fest, was in diesen Fällen geschieht. Und selbst, wenn Ausnahmen manuell behandelt werden, sollten diese zu einem späteren Zeitpunkt mit den normalen Aktivitäten in Einklang gebracht werden können.

4. Das Modellieren von Daten

Zum Prozess der Datenmodellierung zählt nicht nur die Modellierung der Prozesslogik, sondern auch die sinnvolle Strukturierung

der Daten. Bei der Entwicklung einer Low-Code-Anwendung geht es hierbei primär um die Beschreibung von Geschäftseinheiten und deren Beziehungen – ein umfangreiches Thema. Tools können dabei viele, ansonsten manuell zu treffende, Designentscheidungen übernehmen. Einige Fragen lassen sich jedoch nicht vermeiden: Wie geht man mit sich wiederholenden Daten um? Wenn Unternehmen und Kontakte erfasst werden wollen, clustert man die Kontakte dann zu einer sich wiederholenden Gruppe innerhalb jedes Unternehmens? Oder erstellt man getrennte Tabellen für Unternehmen und Personen und fügt einen Mechanismus zur Pflege ihrer Beziehungen mithilfe einer Tabelle hinzu, die aus der Personen-ID, der Unternehmens-ID und der Berufsbezeichnung besteht?

Viele Low-Code-Tools tragen dazu bei, dass Anwender in die Thematik der Datenmodellierung nicht mehr so tief einsteigen müssen und sich beispielsweise Fragen der Entity Normalization und Foreign Keys erledigen. Trotzdem müssen sie verstehen, was diese Tools tun und warum – um Daten auch langfristig nutzen zu können.

5. Die konsistente Entwicklung von Schnittstellen und Anwendungen

Mit dem Einsatz von Low-Code-Tools zur Entwicklung von Anwendungen steigt auch die Zahl der Applikationen und der Benutzer. Entsprechend sollten diese konsistent und klar aufgebaut sein (Buttons an den gleichen

Stellen, identisch aussehende Datenfelder, usw.). Dabei geht es vor allem um Praktikabilität. Wenn das Design nicht jedes Mal neu überdacht werden muss, lassen sich Anwendungen leichter skalieren. Und je ähnlicher sie aussehen und funktionieren, desto schneller finden sich Nutzer damit zurecht. Umgekehrt bedeuten mehrere Möglichkeiten der Nutzung verschiedener Anwendungen auch viele Lernkurven. Kontextwechsel zwischen den einzelnen Interfaces werden erschöpft.

Fazit

Wie eingangs erwähnt, sind all das Dinge, die Entwickler wissen müssen. Das soll aber nicht heißen, dass Low-Code-Tools in den Händen von Nicht-Profis nichts taugen – ganz im Gegenteil. Je besser Mitarbeiter verstehen, wie Entwickler zu denken, wenn sie diese Tools verwenden, desto erfolgreicher werden sie gute Anwendungen entwickeln können. Und je besser diese Anwendungen sind, desto mehr Zuspruch finden sie im gesamten Unternehmen, die wiederum umso besser größere geschäftliche Herausforderungen und Chancen angehen können.

Wer schreibt:

WEBCON ist ein führender Softwareanbieter mit Sitz in Europa, der eine Low-Code-Plattform für die Automatisierung und das Management von Geschäftsprozessen anbietet, die Unternehmen auf der ganzen Welt dabei hilft, effizienter zu arbeiten und das Potenzial des digitalen Wandels zu nutzen. ◀

