

Low-Code

Anwendungen ohne Programmierkenntnisse erstellen

Entwickelten Unternehmen Software bisher für umfangreiche, komplexe Aufgaben, rücken zunehmend kleine zu digitalisierende Prozesse in den Fokus. Low-Code-Plattformen versprechen, dass jeder auf Basis einer grafischen Bedienschnittstelle vollständige Fachanwendungen entwickeln kann. Software-Entwickler steuern lediglich fehlende Bausteine bei.

Definition

Eine Low-Code-Plattform stellt Bausteine für Dateneingabe, Prozesssteuerung und Datenausgabe bereit. Dateneingaben sind etwa mit UI-Elementen oder Event-Listenern möglich, die Prozesssteuerung durch Verzweigungen oder Schleifen, die Datenausgabe ebenfalls mit UI-Element oder gar dem Aufruf fremder APIs. Je nach Spezialisierung der Plattform existieren fachliche Komponenten, etwa zur Mehrwertsteuerberechnung. Durch standardisierte Schnittstellen lassen sich individuelle Bausteine implementieren und der Low-Code-Plattform hinzufügen. Anwender verknüpfen die Bausteine miteinander und definieren so ihre fachlichen Anwendungen. Die Plattform generiert im Anschluss ein lauffähiges Produkt und stellt es unter Umständen direkt produktiv bereit.

Referenzszenario

Ein Unternehmen hat viele abteilungsspezifische, einfache, kleine Prozesse und möchte diese digitalisieren. Allerdings steht nur eine begrenzte Anzahl an IT-Fachkräften für deren Entwicklung und Betrieb zur Verfügung. Unter anderem soll ein überschaubarer Antragsworkflow inklusive eines einfachen Genehmigungsprozesses abgebildet werden. Entsprechend erhalten Fachanwender eine Einführungsschulung in eine Low-Code-Plattform. Sie beginnen mit gelegentlicher Unterstützung der IT-

Software Engineering

- hoher Entwicklerbedarf
- fehlendes Fachwissen bei Entwicklern
- klassisches Software Engineering meist übertrieben

technische Infrastruktur

- steht schneller bereit
- wiederholbare Ergebnisse durch Configuration-as-Code
- bestehende Open-Source-Bibliotheken

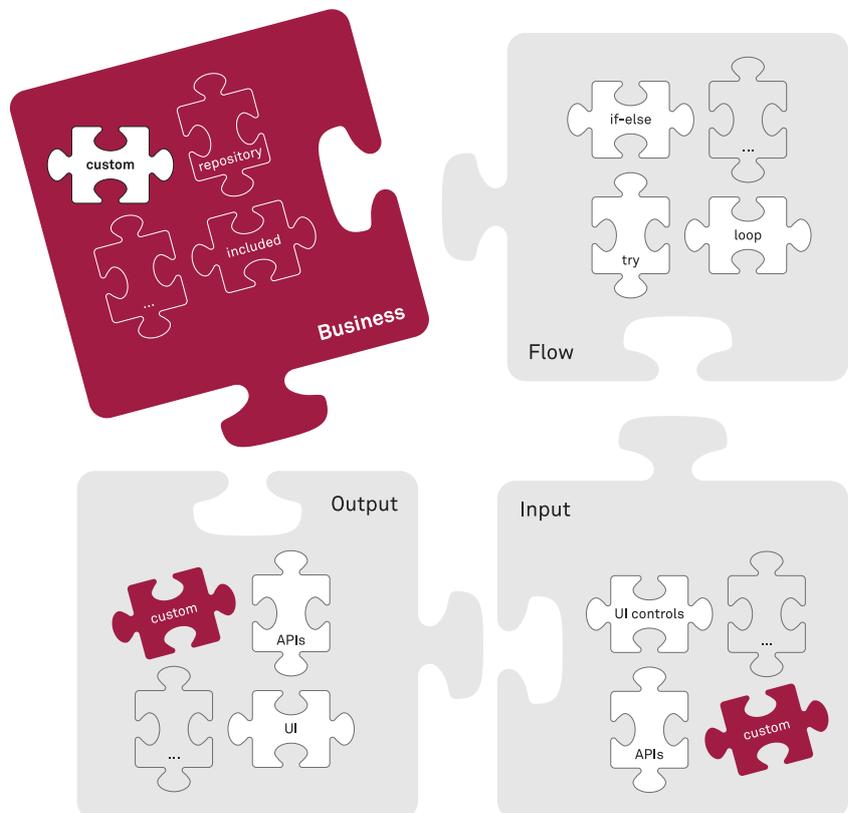
LCO

fachliche Standardisierung

- stark standardisierte Fachprozesse
- Standards in der Prozessmodellierung
- mehr gleichartige Anwendungen

technische Standardisierung

- vielfältige Standard-APIs
- Webstandards für UI-Elemente
- viele öffentliche Datenquellen



Fachkräfte, ihre Anforderungen direkt auf der Plattform zu modellieren. Wegen der hohen fachlichen Nähe der Modellierenden entfällt nahezu jeder fachliche Wissenstransfer. Die IT-Fachkräfte begleiten lediglich die Fachanwender, um deren fehlendes IT-Know-How zu kompensieren und etablierte Good Practices aus der IT in das Modell einfließen zu lassen. Aufgrund der vorgefertigten Bausteine steht bereits nach sehr kurzer Zeit ein erster Prototyp bereit, mit dem sich die Praxistauglichkeit prüfen lässt. Nach wenigen weiteren Iterationen ist eine produktive Applikation fertig.

Potenzial

Low-Code-Plattformen versetzen Unternehmen in die Lage, nach einer geringen Einarbeitungszeit einfache Prozesse selbst zu digitalisieren. IT-Fachkräfte nehmen primär eine unterstützende Rolle ein. Die Anbindung an individuelle Systeme erfordert zwar die Entwicklung spezieller Bausteine, die lassen sich aber in allen Prozessen wiederverwenden. Die vorgefertigten Bausteine und die oft inkludierte Generierungsstrecke verkürzen die Entwicklungszeit gegenüber klassischen Software-Projekten. Weil die fachlichen Wissensträger die einzelnen Anwendungen entwickeln, entstehen hoch akzeptierte Lösungen.

Reifegrad

Das Potenzial von Low-Code-Plattformen

ist erkannt, die Adaptionrate bereits hoch, ebenso die Erwartungshaltungen. Für kleine, überschaubare Anwendungen fallen die Ergebnisse vielversprechend aus. Bei größeren Anwendungen, die über einen längeren Zeitraum gepflegt werden sollen, fehlen allerdings noch Erfahrungswerte. Die meisten Low-Code-Plattformen stellen viele fachliche Funktionen in Form von Standardbausteinen bereit, decken dafür technische Aspekte wie Versionierung und Staging aber häufig nur bedingt ab. Aufgrund der Erwartungshaltung an die fachlichen Anwender fließt meist zu wenig IT-Know-How in die Anwendungen. Das kann Probleme bei Sicherheit, Wartbarkeit oder Stabilität aufwerfen.

Marktübersicht

Low-Code-Plattformen lassen sich grob in drei Kategorien aufteilen: generalistische, technische und branchenfokussierte. Generalistische wie Microsoft Powerapps oder Amazon Honeycode verzichten auf spezifische Fachlichkeit. Technische wie Node-RED oder IFTTT spezialisieren sich darauf, unterschiedliche Systeme miteinander zu verknüpfen. Branchenfokussierte versuchen speziell die Anforderungen einer Branche zu erfüllen, etwa Monday für Teammanagement oder WIX für Webseiten. Aufgrund stetiger Standardisierung der APIs und einer sehr aktiven Open-Source-Gemeinschaft lassen sich für viele Anwendungsfälle ferti-

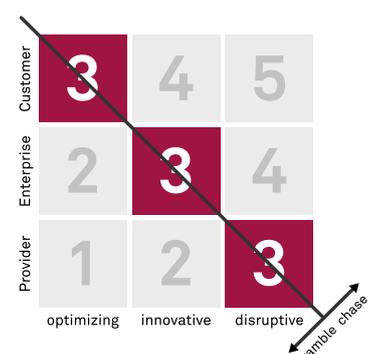
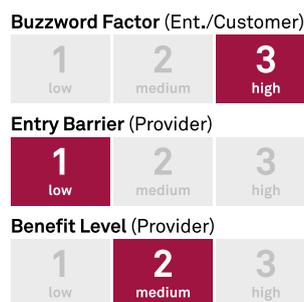
ge Implementierungen finden, die gegebenenfalls noch auf die spezifische Low-Code-Plattform anzupassen sind.

Alternativen

Die von Low-Code-Plattformen bereitgestellten Funktionalitäten existieren häufig als Open-Source-Bibliotheken für unterschiedliche Programmiersprachen. Kleine fachliche Applikationen sind also auch weiterhin mittels klassischer Software-Entwicklung möglich. Diese Bibliotheken gilt es aber zu suchen und zu integrieren. Frameworks vereinfachen dabei die Integrationsleistung. Ferner unterstützen einige Entwicklungsumgebungen die Entwicklung grafischer Bedienschnittstellen mittels Drag-and-Drop von UI-Elementen und Komponenten. No-Code-Plattformen verzichten wiederum komplett auf individuellen Code.

Fazit

- + geringe Einstiegshürde
- + schnelle Ergebnisse
- + fertige Lösungen für Querschnittsthemen
- + weniger fehleranfällig
- bildet nur bedingt komplexe Logiken ab
- starke Plattformabhängigkeit
- keine spezifische Performanceoptimierung
- Good Practices des Software Engineerings kaum anwendbar



<https://msg.direct/techrefresh>

Stand: Dezember 2021