



Low-Code/No-Code-Anwendungs- entwicklung aus Businessperspektive

ÜBERBLICK ÜBER RELEVANTE PLATTFORMEN IN DEUTSCHLAND, ÖSTERREICH
UND DER SCHWEIZ AUS SICHT VON FACHBEREICHEN UND GESCHÄFTSLEITUNG

Arnold Vogt, PAC
März 2025

TOPLOGIC
THE AUTOMATED APPLICATION ENGINE

Inhalt

Einführung	3
Marktanalyse	4
Erkenntnisse	6
Über TopLogic	7
Über PAC.....	7

Dieses Whitepaper wurde im Auftrag von TopLogic erstellt. Weitere Informationen erhalten Sie unter www.pacanalyst.com.

Haftungsausschluss

Die Inhalte dieses Whitepapers wurden mit der größtmöglichen Sorgfalt zusammengestellt, eine Gewähr für ihre Richtigkeit kann jedoch nicht übernommen werden. Einschätzungen und Beurteilungen spiegeln unseren gegenwärtigen Wissensstand im März 2025 wider und können sich jederzeit ändern. Das gilt insbesondere, aber nicht ausschließlich, für zukunftsgerichtete Aussagen. In dieser Studie vorkommende Namen und Bezeichnungen sind möglicherweise eingetragene Warenzeichen.

Nutzungsrechte

Dieses Whitepaper ist urheberrechtlich geschützt. Jegliche Vervielfältigung oder Weitergabe an Dritte – auch auszugsweise – bedarf der vorherigen expliziten Einwilligung des Auftraggebers. Auch die Veröffentlichung oder Weitergabe von Tabellen, Grafiken etc. in anderen Publikationen bedarf der vorherigen Genehmigung.

Unabhängigkeit und Datenschutz

Dieses Whitepaper wurde allein von PAC erstellt. Der Auftraggeber hatte keinen Einfluss auf die Erstellung des Whitepapers.

Einführung

Anwendungen bilden die Grundlage für digitale und zunehmend automatisierte Unternehmensprozesse. Differenzierungspotenzial im Markt entsteht aber weniger durch etablierte Standardanwendungen wie SAP, sondern vielmehr durch eigene, unternehmens-spezifische Anwendungen. Das Thema Anwendungsentwicklung ist daher von zentraler Bedeutung für die Wettbewerbsfähigkeit von Unternehmen. Immer neue technische Möglichkeiten, sich schnell verändernde Kundenbedürfnisse und die rasch zunehmende Anzahl und Komplexität der Apps schaffen ein dynamisches Marktumfeld für die Anwendungsentwicklung. Wegen begrenzter Entwicklerressourcen ist die Auswahl modernster Werkzeuge für die Softwareentwicklung unerlässlich. Die Kernfrage: Wo stehen wir hierbei heute im Markt? An zwei Aspekten kann man die langfristige Entwicklung in der Anwendungsentwicklung gut nachvollziehen: einerseits an Programmiersprachen und anderen Werkzeugen für die technische Realisierung von Anwendungen und andererseits an den methodischen Ansätzen zur Strukturierung des Entwicklungsprozesses.

Die Anfänge von Programmiersprachen gehen zurück auf die 1950/60er Jahre, als mit Fortran, COBOL und BASIC die ersten höheren Programmiersprachen eingeführt wurden, dank derer erstmalig überhaupt komplexere Anwendungen entwickelt werden konnten. C folgte in den 70er-Jahren und bildete die Grundlage für weitere Allzweck-Programmiersprachen die wir heute kennen, wie Java und Python. Neben diesen breit nutzbaren Programmiersprachen haben sich in den 1980/90er-Jahren zunehmend auch domänenspezifische Sprachen wie SQL etabliert, die zwar nur ein begrenztes Anwendungsspektrum abdecken, dafür aber einfacher nutzbar sind. Zusätzlich entstanden grafische Benutzeroberflächen (Graphical User Interfaces, GUI), die es ermöglichten, Code via objektorientierter Drag-and-Drop-Funktion zu ändern. Damit war der Grundstein gelegt für Plattformen zur vereinfachten Anwendungsentwicklung mit deutlich weniger Programmieraufwand (Low Code) in den 2000er-Jahren. Seit den 2010er-Jahren sind Low-Code-Plattformen vom Markt nicht mehr wegzudenken, und sie entwickeln sich mehr und mehr zu No-Code-Plattformen, die auch ganz ohne Programmierkenntnisse einsetzbar sind. Letzteres schafft die Voraussetzung für weitere Beschleunigung sowie Kostensenkungen in der App-Entwicklung. Zudem sind auch Nicht-Programmierer in der Lage, schnell und einfach Anwendungen zu entwickeln.

Parallel zu den Programmiersprachen und Low-Code/No-Code-Werkzeugen hat sich auch die Methodik der Anwendungserstellung stetig weiterentwickelt. Die Methodik befasst sich mit dem gesamten Software-

entwicklungsprozess vom Entwurf über das Testen bis hin zur Implementierung und Dokumentation von Software. Ziel ist hierbei ein systematischer Ansatz der Softwareentwicklung für eine kundenorientierte und effiziente Projektdurchführung.

Das Wasserfallmodell war hierbei bis in die 1970er Jahre der vorherrschende Ansatz. Dieser Ansatz folgte einer streng linearen und sequenziellen Struktur, bei der jede Entwicklungsphase (Anforderungen, Design, Implementierung, Test und Wartung) der Reihe nach abgearbeitet wurde. Dieser Ansatz – dazu zählt auch das bekannte V-Modell – bot zwar eine effiziente Struktur, war jedoch zeitintensiv, unflexibel und konnte nicht gut an geänderte Kundenanforderungen angepasst werden. Zur Behebung dieser Einschränkungen entstanden seit den 1980er Jahren iterative Modelle wie das Spiralmodell oder das Rapid Application Development (RAD). Diese Modelle führten das Konzept sich wiederholender Zyklen oder Iterationen ein, das eine kontinuierliche Verfeinerung von Anforderungen und Funktionen ermöglichte. 2001 wurde das Agile Manifest veröffentlicht, das die Softwareentwicklungslandschaft weiter veränderte, hin zu kontinuierlichen Entwicklungszyklen. Agile Methoden wie Scrum und Kanban fokussierten auf Schnelligkeit, Flexibilität, Zusammenarbeit mit dem Kunden und die schrittweise Bereitstellung funktionaler Software durch sogenannte Sprints. Aufbauend darauf entstand in den Folgejahren der DevOps-Ansatz um Entwicklungs- (Dev) und Betriebsteams (Ops) besser zu vernetzen. DevOps steigert die Effizienz von agilen Methoden und fördert eine Kultur der Zusammenarbeit, Automatisierung und Integration über den gesamten Softwareentwicklungszyklus hinweg. Die Entwicklung von Methoden ist damit aber sicher nicht abgeschlossen. KI und Prozessautomatisierung werden gesteigert eingesetzt, um Softwareentwicklungsprozesse effizienter, flexibler und schneller zu gestalten. Sie tragen zur Verkürzung der Entwicklungszyklen sowie zur besseren Anpassung an Kundenanforderungen bei.

Zusammenfassend kann man sagen, dass agile, kooperative und zunehmend integrierte und automatisierte Softwareentwicklungsprozesse in Kombination mit Low-Code/No-Code (LC/NC)-Werkzeugen heute eine zentrale Rolle in der Softwareentwicklung spielen. Das vorliegende Whitepaper hat den Anspruch, die relevanten LC/NC-Plattformen für den DACH-Markt (Deutschland, Österreich und die Schweiz) zu identifizieren und den Markt aus Kundenperspektive zu segmentieren. Das schafft Transparenz und erlaubt Kunden somit ihren Bedürfnissen entsprechend eine schnelle und einfache Identifikation von relevanten Anbietern im lokalen Markt.

Marktanalyse

Definition des Themas

LC/NC-Plattformen verfolgen einen visuellen Ansatz bei der Softwareentwicklung, bei dem Benutzer Anwendungen und Prozesse mit wenig bzw. ganz ohne Programmiercode erstellen können. Anstatt textbasierter Programmiersprachen nutzen Benutzer hierbei eine integrierte Entwicklungsumgebung mit visuellen Drag-and-Drop-Funktionen, um Geschäftsanwendungen schneller und einfacher zu erstellen.

Ziel der Marktanalyse

Es werden relevante LC/NC-Plattformen für den DACH-Markt identifiziert und der Markt aus Kundensicht segmentiert. Mit diesem Ansatz erhalten Fachabteilungen und Geschäftsleitung einen schnellen Überblick über relevante Anbieter im lokalen Markt – entsprechend verschiedenen Kundenbedürfnissen.

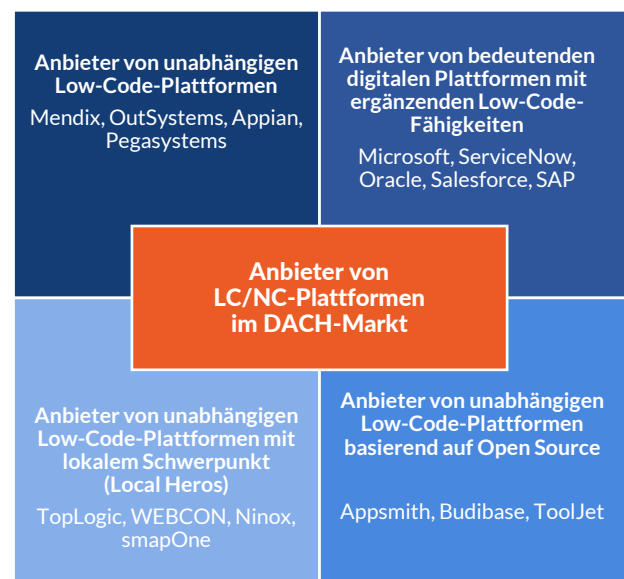
Kriterien für die Anbietersauswahl

Entsprechend der obigen Definition hat PAC Plattformanbieter ausgewählt, die relevante Kundenreferenzen im DACH-Markt nachweisen können. Da unser Fokus auf Plattformen liegt, die für eine Vielzahl von verschiedenen Projekten zur Anwendungsentwicklung eingesetzt werden können, wurden all jene Anbieter nicht berücksichtigt, die nur einen begrenzten Fokus auf sehr bestimmten, oft branchenspezifischen Prozessen haben, wie zum Beispiel MES

(Manufacturing Execution Systems). Ebenfalls ausgeschlossen wurden reine Workflow Automation Tools, die sich im Wesentlichen um das Routing von Informationen zwischen Anwendungen kümmern, und nicht um die Erstellung einer Anwendung.

Betrachtete Anbieter

Die für diese Marktbetrachtung ausgewählten 16 LC/NC-Plattformanbieter können in vier Gruppen unterteilt werden:



© PAC 2025

Marktsegmentierung

PAC differenziert relevante LC/NC-Plattformen in dieser Marktbetrachtung nach zwei Kategorien: nach priorisierten Anwendungsfällen und nach Lizenzmodellen. Diese Sichtweise auf den Markt bietet Anwenderunternehmen eine hilfreiche Grundlage, um schnell und einfach, relevante Plattformanbieter für eine tiefergehende Betrachtung im nächsten Schritt auszuwählen.

Lizenzmodell-Perspektive

Die hier vorliegende Marktbetrachtung zeigt, dass neben den üblichen Lizenzmodellen inzwischen auch schon eine Reihe von Open-Source-basierten Plattformen im DACH-Markt verfügbar sind. Aufgrund der damit verbundenen Kosteneffizienz („No-frills“-Ansatz), sowie der Flexibilität bei der Auswahl des Lizenzmodells (z.B. nach „Nutzungsart“ und/oder „Support-Level“), stellen diese Plattformen heute bereits eine interessante Alternative für Kunden dar.

Anwendungsfall-Perspektive

Verschiedene Anwendungsfälle stellen unterschiedlich hohe Anforderungen an Datenmodelle (die Art der Datenstrukturierung), Anwendungslogik (die Arbeitsabläufe in der Applikation) und an die graphische Benutzerschnittstelle (Graphical User Interface, GUI), welche das Front-End für den Anwendungsnutzer bildet. Diese Marktbetrachtung unterscheidet daher drei grundlegende Anwendungsfälle:

- **Entwicklung von einfachen Anwendungen für den allgemeinen Gebrauch (eher unternehmensintern):** Dieser Anwendungsfall repräsentiert den typischen Einstiegspunkt für Unternehmen in das Thema LC/NC-Entwicklung. Die Designanforderungen an GUI, Anwendungslogik und Datenmodelle sind hierbei moderat. Häufig handelt es sich um einfache Apps rund um bestehende Unternehmensanwendungen wie ERP (z.B. Genehmigungsprozess von Bestellverfahren). Aufgrund der Vielzahl von Anwendungen und Entwicklern, die dabei häufig orchestriert werden müssen, spielen Governance-Frameworks und DevOps-Methoden für gut inte-

grierte und automatisierte Entwicklungsprozesse eine wichtige Rolle.

- **Entwicklung von Anwendungen mit anspruchsvoller User Experience (besonders für Kunden):** Dieser Anwendungsfall ist im Vergleich zum ersten Anwendungsfall deutlich anspruchsvoller. Während die Designanforderungen an Anwendungslogik und Datenmodelle weiterhin eher moderat sind, sind die Anforderungen an das User Experience Design deutlich gestiegen. Selfservice-Apps für Kunden sind ein typisches Beispiel. Aufgrund der Zielgruppe (Kunden) sind agile Methoden in Kombination mit reibungslosen DevOps-Prozessen angebracht.
- **Entwicklung von komplexen Anwendungen (eher unternehmensintern):** Dieser Anwendungsfall ist im Vergleich zu den beiden anderen wesentlich anspruchsvoller. Während die Anforderungen an das Design von Anwendungslogik und/oder Datenmodellen hoch sind, sind die Anforderungen an das User Experience Design häufig eher moderat. Aufgrund der hohen Komplexität sind agile Methoden mit hoher Flexibilität, enger Zusammenarbeit mit Fachspezialisten und die schrittweise Bereitstellung funktionaler Software hierbei von zentraler Bedeutung. Anwendungen mit Echtzeittransparenz über komplexe Produktions- und Produktentwicklungsprozesse, bei denen Datenmodelle verschiedener Anwendungen integriert werden müssen, sind eines der Beispiele in diesem Zusammenhang.

Komplexitätsmuster von Anwendungsfällen
(0=gering / 1=moderat / 2=hoch)



© PAC, März 2025

Vendor Landscape von Low-Code/No-Code-Plattformen im DACH-Markt

ANWENDUNGSFÄLLE NACH KOMPLEXITÄTSGRAD	Entwicklung von komplexen Anwendungen	ServiceNow Appian Pegasystems	TopLogic
	Entwicklung von Anwendungen mit anspruchsvoller User Experience	Mendix Salesforce OutSystems	
	Entwicklung von einfachen Anwendungen für den allgemeinen Gebrauch	Microsoft Oracle SAP WEBCON Ninox smapOne	Appsmith Budibase ToolJet
		Standard	Flexibel (Open-Source-basiert)
LIZENZMODELLE			

© PAC 2025

Anmerkung: Die Positionierung der Anbieter erfolgt nach Lizenzmodell und priorisiertem Anwendungsfall.

Erkenntnisse

Zusammenfassung der zentralen Erkenntnisse aus der hier vorliegenden Marktanalyse:

- LC/NC-Fähigkeiten sind heute von zentraler Bedeutung für die Softwareentwicklung, und es gibt bereits eine ganze Reihe relevanter LC/NC-Plattformanbieter im DACH-Markt.
- Ein Großteil der hier betrachteten Plattformen fokussiert sich auf eher einfache Anwendungsfälle mit moderatem Komplexitätsgrad. Neun von 16 Anbietern haben hier ihren Schwerpunkt. Nur vier von 16 Anbietern in unserer Analyse zeigen einen klaren Schwerpunkt auf komplexen Anwendungsfällen (Design von anspruchsvoller Anwendungslogik und/oder Datenmodellen).
- Open-Source-basierte LC/NC-Plattformen stellen bereits heute eine relevante Anbietergruppe im DACH-Markt dar, und weitere interessante Lizenzmodelle kommen hinzu. Ein gutes Beispiel ist das Duallizenzierungsmodell von TopLogic. Der Kunde entscheidet bei der Inbetriebnahme, ob er proprietär oder Open Source lizenzieren möchte – und das bei identischem Funktionsumfang.
- Die Nutzung der LC/NC-Fähigkeiten von den hier erwähnten globalen Anbietern bedeutender digitaler Plattformen, wie Microsoft, ServiceNow, Oracle, Salesforce und SAP, kann Synergien erzeugen, wenn diese Plattformen bereits im Unternehmen im Einsatz sind. Zu bedenken ist jedoch, ob eine dieser Plattformen auch eine dominierende, strategische Rolle im Unternehmen einnehmen soll (Stichwort „Vendor-Lock-in“).
- Während DevOps-Methoden für gut integrierte und automatisierte Entwicklungsprozesse eine besonders wichtige Rolle spielen bei der Entwicklung und Pflege von vielen einfachen Anwendungen, spielen agile Entwicklungsmethoden mit hoher Flexibilität, enger Zusammenarbeit und schrittweiser Bereitstellung eine ganz entscheidende Rolle bei der Entwicklung von komplexen Anwendungen.

Über TopLogic



TopLogic ist eine führende No-Code-Plattform für modellzentrierte Softwareentwicklung. Mit Open-Source-Technologie und industrieller Skalierbarkeit ermöglicht sie schnellere, kosteneffizientere Lösungen. Insbesondere für die Bereitstellung komplexer und wettbewerbsdifferenzierender Anwendungen wird TopLogic von Großunternehmen auch in near- und offshore Kooperationen eingesetzt.

Weitere Informationen erhalten Sie unter www.top-logic.com und www.opensource.top-logic.com.

Über PAC



Wir sind ein Content-basiertes Unternehmen mit Berater-DNA. PAC ist das führende europäische Marktanalyse- und Beratungsunternehmen für die IT-Branche. Wir unterstützen Softwareanbieter und IT-Dienstleister weltweit. Seit 1976 stehen wir unseren Kunden zur Seite, sowohl bei der Interpretation der Marktdynamik als auch bei ihrer Umsatzsteigerung und Profilschärfung. Mit unserem unübertroffenen Verständnis für Marktentwicklungen in Europa und unseren fundierten Analysen unterstützen wir führende Marktakteure bei der Strategiefindung, Go-to-Market-Optimierung und der Gewinnung zusätzlicher Marktanteile. Mit einem Team von über 100 Experten und Expertinnen in Europa bietet PAC Beratung auf Basis von Marktanalysen. Unsere Marktforschung deckt über 30 Länder weltweit ab und stützt sich auf die drei Portfoliosäulen Guidance, Insights und Visibility sowie unsere renommierte SITSi® Research-Plattform. Nähere Informationen finden Sie auf www.pacanalyst.com.

Kontakt:

Arnold Vogt

Principal Analyst,
Head of Digital & IoT Practice

a.vogt@pacanalyst.com

+49 171 865 11 84

PAC Deutschland

Holzstr. 26,
80469 München

www.pacanalyst.com

www.sitsi.com

