



Inhalt

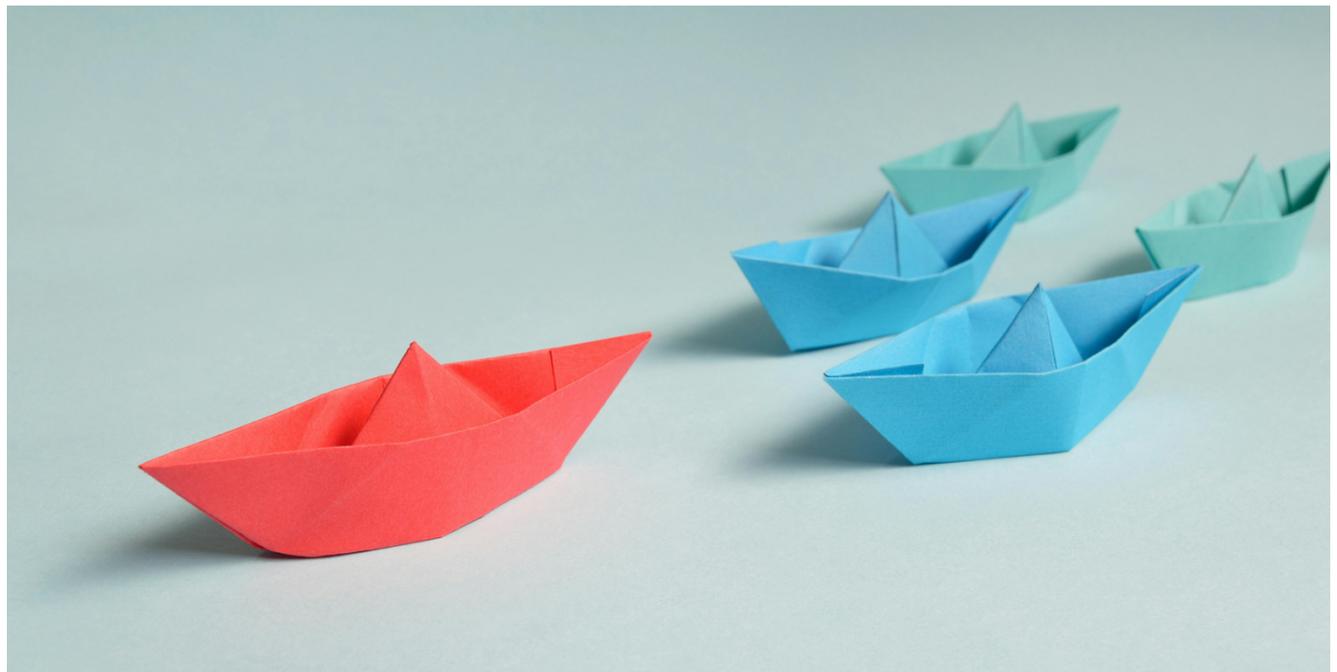
Seite



**Low Code/No Code ohne Regeln –
warum Governance bei LC/NC so wichtig ist 3**

Low Code/No Code ohne Regeln – warum Governance bei LC/NC so wichtig ist

Die sich auf der Beschleunigungsspur befindliche Digitale Transformation und der damit verbundene wachsende Bedarf an Fachkräften für die Umsetzung der Projekte, stehen sich diametral gegenüber. Das Ergebnis dieser Gegensätzlichkeit ist gehemmtes Wachstum. Um dem entgegenzuwirken und Wachstum trotz des Engpasses zu ermöglichen, braucht es neue Wege bei der Entwicklung von Softwarelösungen. Ein Weg ist, die Entwicklung von Lösungen nicht nur durch ausgebildete Software-Ingenieure durchzuführen, sondern jene Akteure miteinzubeziehen, die bereits über das entsprechende Domänenwissen verfügen. Dies gelingt mit Entwicklungsplattformen, die weniger technische Expertise voraussetzen, mit Fachwissen und Nutzungskompetenz aber effizient eingesetzt werden können. Low-Code oder No-Code-Plattformen sind hierfür das geeignete Werkzeug. Mitarbeiter aus dem Fachbereich besitzen das nötige Domänenwissen und können mit diesen Plattformen Lösungen realisieren. Und dies in relativ kurzer Zeit. Die Mitarbeiter aus den Fachbereichen, die anhand von Low-Code oder No-Code-Plattformen Applikationen entwickeln, nennt man Citizen Developer.



In der Rückschau der letzten Jahrzehnte lässt sich feststellen, dass es bereits in vielen Unternehmen nicht unüblich war bzw. ist, Lösungen durch den Fachbereich zu realisieren. Diese Lösungen wurden beispielsweise mit Excel (inkl. Macros) oder Access implementiert. Das Ergebnis dieser Vorgehensweise ist jedoch eine Schatten-IT, mit häufig diversen Herausforderungen im Lebenszyklus dieser Anwendungen. Mitarbeiter, welche die Lösung implementiert haben, wechseln die Abteilung oder verlassen das Unternehmen, entstandene Dateninseln führen zu mehrfacher Datenhaltung und Problemen bzgl. Datensicherheit

und Datennutzung. Häufig genug wurden Technologien eingesetzt, die aus der Wartung gelaufen sind oder Schwachstellen aufweisen. Schwachstellen, die möglicherweise durch Hacker ausgenutzt werden können. Ganz zu schweigen von etwaigen rechtlichen Aspekten und Haftungsfragen. Die Nutzung von Low-Code-Plattformen birgt vergleichbare Gefahren, sodass es heißt, aus der Vergangenheit zu lernen. Mit einem durchdachten Governance-Konzept bietet Low-Code jedoch nicht nur den Vorteil der Effizienz, sondern lässt sich sinnvoll als ein Baustein in die IT-Landschaft integrieren und langfristig einsetzen.

Low-Code Governance

Im Kontext von Low-Code bekommt das Prozesswissen der Citizen Developer eine entscheidende Rolle. Vor- oder nachgelagerte Prozesse sind oft nicht vollständig oder gar nicht bekannt und entsprechend fehlt das nötige Wissen, um etwaige Abhängigkeiten ein- und abzuschätzen. Die Einführung von Regeln, also eine Low-Code-Governance, ist daher unablässig. Wichtig dabei ist, dass das Regelwerk kein Hemmnis für die Einführung und Nutzung von Low-Code-Plattformen ist, vielmehr alle relevanten Aspekte für eine effiziente Umsetzung beleuchtet. Eine solche Betrachtung zeigt, ob man eine Lösung benötigt und wenn ja, ob diese mit einer Low-Code-Plattform umsetzbar ist. Bei der Erstellung des Regelwerks sind folgende Aspekte relevant:

- Es muss definiert werden für welche Projekte und Anwendungen Low-Code eingesetzt werden kann
- Sicherheit muss mitgedacht werden. Es müssen strenge Sicherheitsanforderungen festgelegt und erfüllt werden
- Es muss ein Regelwerk für die Citizen Developer im Unternehmen definiert werden

Damit keine Schatten-IT entsteht ist es unabdingbar, dass die Verantwortung für den Einsatz von Low-Code sowie den im Rahmen der Entwicklung benötigten und genutzten Ressourcen, klar bei der IT-Abteilung liegt. Hierbei geht es zum einen um Hardware- und

Softwarenutzung, aber natürlich auch um die Verwendung von Datenquellen, die für die Entwicklung einer Anwendung mittels einer Low-Code-Plattform verwendet werden können und dürfen. Mit den folgenden Schritten gelingt die Einführung und Governance von Low-Code Development Plattformen (LCDP).

Entwicklungsarten und Entscheidungsmatrix

Die Verwendung von Low-Code Development Plattformen mag auf den ersten Blick eine

sehr verlockende Möglichkeit sein, um Unternehmensanwendungen zu implementieren. Doch ob eine Implementierung mittels Low-Code-Plattform möglich und sinnvoll ist, muss geprüft werden. Hierzu bietet sich eine Matrix zur Bewertung der wichtigsten Aspekte an. Stehen alle relevanten Informationen über vorhandene Datenquellen bereits zur Verfügung? Wie komplex ist die Geschäftslogik? Sind beispielsweise keine Schnittstellen zu Umsystemen vorhanden oder bieten diese nicht die nötige Funktionalität, sind Probleme mit Low-Code vorprogrammiert.

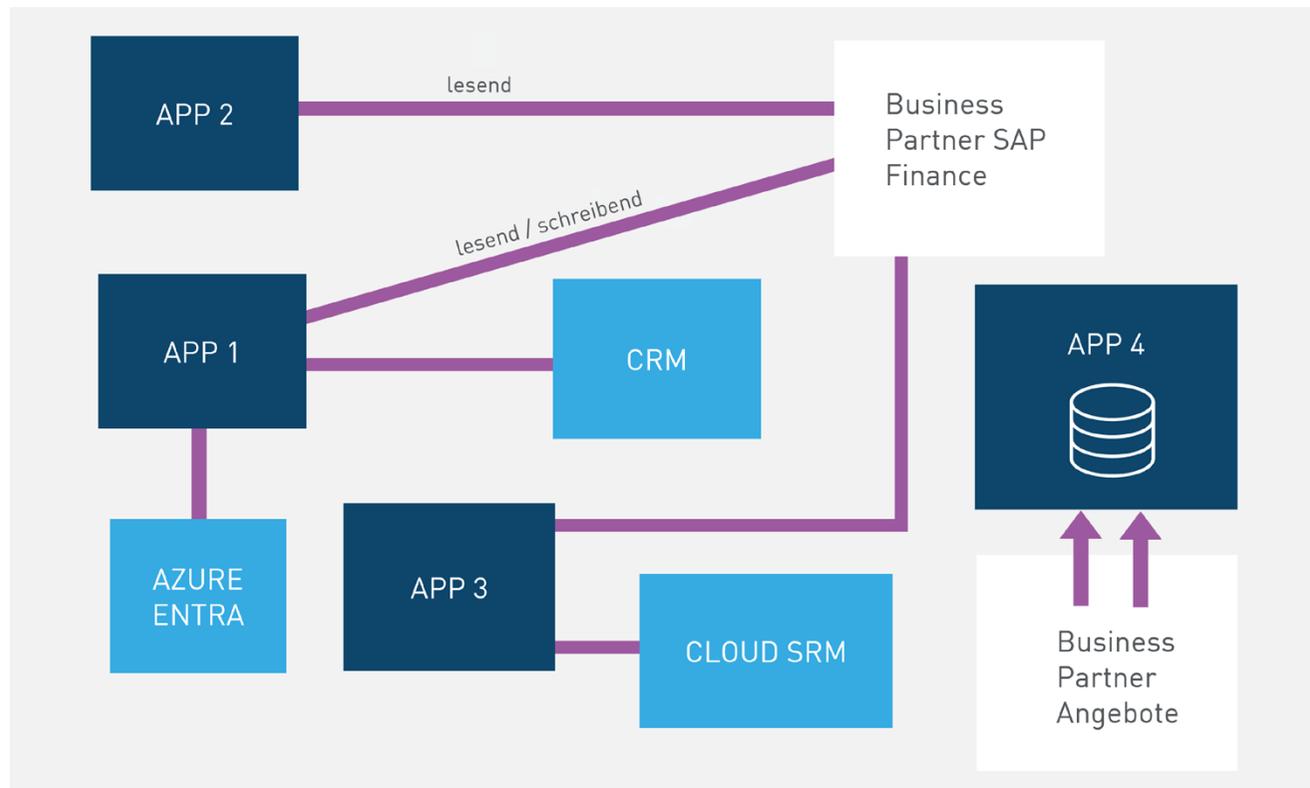
Anforderungen	Bewertung
Komplexität der Benutzeroberflächen	
UX-Konformität der Benutzeroberflächen	
Anforderungen an Usability/Barrierefreiheit	
Anforderungen an eine mobile Nutzung	
Wiederverwendbarkeit der Logik	
Cloud-Fähigkeit	
Datenquelle inkl. Schnittstellen vorhanden	
Komplexität der Geschäftslogik der App	
Komplexität der Datenbanklogik	
Anforderung an die Performance der Datenoperationen	
Spezielle funktionale Anforderungen	
Implementierungsaufwand	

Tabelle 1: Template einer Gewichtungsmatrix

Überblick über Schnittstellen und Kopplungen

Anwendungen, welche mittels LCDP erstellt werden, greifen auf Daten zu und modifizieren diese. Diese Daten werden in Umsystemen gehalten bzw. von diesen geliefert. Die Verantwortlichkeit der benötigten Schnittstellen liegt bei der IT. Dies heißt auch, falls es noch keine vorhandene Schnittstelle gibt, dass diese dediziert für den jeweiligen Use Case geschaffen wird. Für bereits bestehende Schnittstellen muss dokumentiert werden, wer diese nutzt.

Es bedarf einem Gesamtüberblick. Um Wildwuchs zu vermeiden, darf der Fachbereich selbst keine Schnittstellen beauftragen. Nur so kann ein stabiles und nutzbares Gesamtsystem entstehen. Die IT-Abteilung kennt die entstandenen und geplanten Low-Code Applikationen auf einem gewissen Abstraktionsniveau und bringt diese zueinander in Beziehung – grafisch als Schaubild (vgl. Abbildung). Datenflüsse, Abhängigkeiten, Prozessabläufe und Zugriffsrechte sind so erkennbar. Mögliche Duplikate können früh identifiziert werden.



IT-Abteilung als Dienstleister

Low-Code für die Fachabteilung zu ermöglichen, heißt für die IT, neben dem technischen Betrieb der Plattform auch den organisatorischen Rahmen zu schaffen. Synergien und Potentiale, die über diese Plattform erreicht werden können, müssen von der IT-Abteilung (als zentraler Ankerpunkt) proaktiv angegangen werden. Die IT-Abteilung muss hier als Dienstleister im eigenen Haus auftreten. In dieser Rolle fungiert die IT als Vermittler, spricht erkannte Doppelungen an, moderiert den Dialog zwischen Abteilungen. Nahezu alle gängigen Low-Code Plattformen erlauben es, neben den vom Hersteller schon vorhandenen Programmierbausteinen eigene (unternehmensspezifische) Bausteine zu implementieren. Die Pflege dieser Bausteine kann nicht bei einzelnen (Low-Code) Applikationen liegen, sie muss über die IT-Abteilung erfolgen. Ein zentrales Gate in der IT-Abteilung begleitet die Fachbereiche, von der Anforderungsbewertung über die Umsetzung bis hin zur Retrospektive. Anstatt einfach „drauf los“ zu entwickeln, arbeiten IT und Fachbereich Hand in Hand für ein langfristiges, tragfähiges Lösungsnetz. Wiederverwendbarkeit und klare Verantwortlichkeiten sind gesichert. Entstandene Daten lassen sich optimal weiterverwenden und durch andere Geschäftsprozesse nutzen. Hohe Datenqualität ermöglicht es, intelligente Algorithmen perfekt zu integrieren.

Datenschutz und Datensicherheit

Anwendungen erstellen, ändern, verarbeiten und analysieren Daten. So eben auch Low-Code Anwendungen. Die Entwicklung von Anwendungen, welche mit LCDP erstellt sind, ist effizient. Probleme bzgl. Datenschutz und Datensicherheit wirken jedoch lange nach. Die einfache Nutzung von Daten und Datenschnittstellen führt im Worst Case zu Datenleaks und Lücken in der Sicherheitsinfrastruktur von Unternehmen. Daher muss sichergestellt sein, dass LCDP-Nutzer nur die Daten einsehen, für die sie auch berechtigt sind. Dies gilt auch für vorhandene Schnittstellen. Schutzklassen definieren die entsprechenden Berechtigungen. Es bietet sich hierfür beispielsweise die Norm ISO-27001 an. Mit einem verlässlichen Konzept können Unternehmen den nötigen Schutz gewährleisten. Die Richtlinien zur sicheren Programmierung,

Systemarchitekturen und Sicherheitstests müssen auch in der Low-Code Entwicklung berücksichtigt werden. Regelmäßige Audits stellen sicher, dass Richtlinien umgesetzt sind. Ebenso wichtig sind verpflichtende Mitarbeiterschulungen, um Mitarbeiter für das Thema Sicherheit zu sensibilisieren.

Auch im Hinblick auf die Pflege und Erfassung von Daten gibt es im Bereich Low-Code und verteilte Entwicklung kleiner Applikationen in Fachabteilungen neue Herausforderungen – aber auch Chancen. Es werden mitunter nun Daten erfasst, die auch für andere Unternehmensbereiche interessant sind. Ggfs. bauen anderen Low-Code Applikationen auf diesen Daten auf und generieren weitere Mehrwerte! Der bereitstellenden (Low-Code) Applikation ist dies vielleicht gar nicht bewusst. Daten werden mit geringer Qualität eingepflegt (weil es für den eigenen Use Case ausreicht) – oder

Daten werden unbewusst anders interpretiert – oder sogar im Laufe der Zeit in Hinblick auf die Erfassung geändert. Die Bedeutung kann sich dadurch ändern, ohne, dass darauf aufbauende Applikationen dies „mitbekommen“ und ggfs. eigene Prozesse und Entscheidungen am Ende auf falschen Daten beruhen. Die Einordnung und der Überblick im Gesamtkontext sind immens wichtig!

Qualitätssicherung und Testing

Auch wenn es sich bei Low-Code Anwendungen meist um kleinere, abgeschlossene Applikationen handelt, müssen auch diese gewissen Qualitätsstandards entsprechen. Am Ende werden die Applikationen produktiv genutzt und die Ergebnisse und Anzeigen müssen stimmen. Nutzer der Systeme verlassen sich darauf. Hier einige Inspirationen von minimalen Qualitätsstandards:

- Review-Prozess bei Änderungen (Vier-Augen-Prinzip)
- Durchgängige UX
- Nutzung technischer Mittel, um Fehleingaben zu verhindern
- Changelog führen
- Definierte (manuelle) Regressionstests bei Änderungen durchführen
- Forcierung der Nutzung von technischen Tests
- Schulungen für Citizen-Developer

Schutzziel	Vertraulich	Verfügbarkeit	Integrität	Schaden
Schutzklasse von 0 - 3	Offen	Niedrig	Niedrig	Unbedeutend bis gering
	Intern	Mittel	Mittel	Mäßig
	Vertraulich	Hoch	Hoch	Groß
	Streng vertraulich	Sehr hoch	Sehr hoch	Sehr groß Gefährdung des Unternehmens

Tabelle 2: Übersicht Schutzziele und Schutzklassen

Wie etabliert man Low-Code?

Um Low-Code und Low-Code-Plattformen einzuführen, langfristig erfolgreich zu etablieren und zukunftsfähig tragfähige Lösungen und Lösungslandschaften zu implementieren, bedarf es einem Mehrklang aus Maßnahmen und Regeln. Einzig allein die Bereitstellung einer Plattform kann nicht die Lösung sein. Zu-

nächst muss klar im Unternehmen überprüfbar geregelt sein, ob die Umsetzung mittels Low-Code-Plattform auch tatsächlich möglich ist oder aber ob Unternehmensinteressen der Nutzung gegenüberstehen. Dies kann beispielsweise dann der Fall sein, wenn Prozesse betroffen sind, die komplex und zentral sind. Nutzer von LCDP müssen sich bewusst sein, welches Instrument sie in der Hand halten.

Falsch eingesetzt können im schlimmsten Fall Sicherheitslücken und nicht mehr kontrollierbare Daten- und Funktionsinseln entstehen. Es ist daher unablässig, nur einen begleiteten Low-Code-Einsatz im Unternehmen zu erlauben. Abschließend fassen wir die wichtigsten Aspekte einer guten Low-Code-Governance zusammen. Eine effektive Low-Code-Governance berücksichtigt folgende Aspekte und stellt sicher, dass Low-Code-Entwicklungsaktivitäten sowohl produktiv als auch sicher sind:

- Klare Richtlinien und Standards
- Sicherheit & Compliance
- Dokumentation, Transparenz und klar geregelte Verantwortlichkeiten
- Versionierung und Change Management
- Überwachung und Performance-Optimierung
- Schulungen und Unterstützung
- Low Code ist nicht immer die Antwort – oft hilft nur der ProCoder

ex|Xcellent
solutions

eXXcellent solutions GmbH

Beim Alten Fritz 2
89075 Ulm

info@excellent.de
www.excellent.de



Michael Englbrecht

ist Portfolio Manager für den Bereich SAP sowie Principal Managing Consultant bei der eXXcellent solutions GmbH in Ulm. Dort entwickelt er das Leistungsportfolio der eXXcellent solutions im Bereich SAP weiter. In seiner Rolle als Business Manager ist er für die Umsetzung von Kundenprojekten zuständig und deren langfristigen Betreuung. Zu seinen Schwerpunkten gehört unter anderem die Beratung von Unternehmen bei der Umsetzung innovativer SAP Projekte. Er ist Autor und Co-Autor der Standardwerke zu SAP Fiori, SAPUI5 und SAP Schnittstellenprogrammierung, veröffentlicht im Rheinwerk Verlag (SAP Press).



Ralf Enderle

ist Portfolio Manager für den Bereich Business Applications sowie Senior Software Architekt bei der eXXcellent solutions GmbH. Er ist branchen- und technologieübergreifend unterwegs und unterstützt Projekte in der Konzeption und Weiterentwicklung. Sein Fokus liegt dabei auf der effizienten Entwicklung und dem Schaffen von Synergien über Projekte hinweg. Er ist stetig auf der Suche nach innovativen, soliden und nachhaltigen Lösungen und Konzepten – neue Themen und Trends finden bei ihm immer ein offenes Ohr.