Low-Code als Triebfeder für Innovation in der Verwaltung

Ein Handlungsleitfaden für Auswahl, Einführung und Betrieb von Low-Code-Plattformen



bridging IT
Gruppe
Ein Unternehmen von SPIE

Inhaltsverzeichnis

Einleitung **4**

Low-Code-Plattformen (LCP): Vorteile und Herausforderungen **5**

Chancen von Low-Code in der Verwaltung **6**

Marktanalyse **7**
Die Plattformen im Vergleich 8

Handlungsleitfaden zur Einführung von LCP **10**
Low-Code-Canvas: Unterstützung bei der Auswahl der richtigen Plattform 10
Organisatorische Voraussetzungen für eine erfolgreiche Einführung von LCP 11
Herausforderung Vendor Lock-In 12

Fazit und Ausblick **13**

Über uns *14*

Glossar *15*

Abkürzungsverzeichnis *16*

Quellenverzeichnis *17*

Einleitung

Low-Code und Low-Code-Plattformen, kurz LCP, sind aus technologischer Sicht kein neues Phänomen, gewinnen aber in den letzten Jahren immer mehr an Fahrt. Bei einer Marktgröße von etwa 20 Mrd. Euro im Jahr 2023 wird sich deren Wert bis 2032 auf voraussichtlich 175 Mrd. Euro nahezu verneunfachen[19]. Zum einen haben sich die Plattformen deutlich weiterentwickelt und diversifiziert. Zum anderen versprechen Anbieter, dass man auch ohne Informatikstudium oder -ausbildung schnell und effizient Applikationen plattformübergreifend mit Hilfe von Drag & Drop-Editoren nach dem Bausteinprinzip zusammenstellen und veröffentlichen kann.

Das macht LCPs nicht nur interessant, um dem wachsenden Fachkräftemangel entgegenzuwirken[21], sondern auch die vorhandenen Fachkräfte zu entlasten und strategisch wertschöpfender einzusetzen[15]. Eine entscheidende Rolle spielen dabei sogenannte Citizen Developer, technikinteressierte Domänenexpert:innen[39], die durch ihr erworbenes Wissen im Umgang mit LCP Innovationstreiber sein und das Bindeglied zwischen Fach- und IT-Abteilung bilden können[8].

Im Hinblick auf die deutsche Verwaltung entsteht durch LCPs das Potenzial Schatten-IT zu begrenzen und die Akzeptanz der Belegschaft in Richtung verwaltungsinterner Digitalisierung zu erhöhen, indem diese befähigt wird, Prozesse selbstwirksam zu digitalisieren, ohne vollumfänglich von internen wie externen IT-Dienstleistern abhängig zu sein[8, 15, 17, 30]. Die Herausforderung besteht darin, in der Menge der verfügbaren Plattformen die richtige zu finden, um diese Befähigung zu ermöglichen.

Eine solche Plattform in Unternehmen oder Verwaltungen einfach bereitzustellen, führt jedoch nicht automatisch zu Digitalisierung und Innovationen. Neben technologischen gibt es auch organisatorische Aspekte zu beachten, die für die erfolgreiche Implementierung solcher Plattformen wichtig sind. Dieser Leitfaden gibt in den folgenden Abschnitten einen Überblick über LCP und deren spezifische Vorteile und Herausforderungen. Es wird gezeigt, wie LCP im Verwaltungsumfeld zur Digitalisierung beitragen können. Im weiteren Verlauf werden die wichtigsten Low-Code-Plattformen vorgestellt und verglichen. Zuletzt stellen wir einen Handlungsleitfaden für die erfolgreiche Auswahl und Implementierung einer LCP vor, der den Weg hin zu echter Wertschöpfung ebnen soll.

Autoren



Philip Heinz
philip.heinz@bridging-it.de



Detlef Schumann

Weitere Ansprechperson



Lena Brommer
lena.brommer@bridging-it.de



Low-Code-Plattformen

Vorteile und Herausforderungen

Für die weitere Betrachtung benötigen wir zuerst eine Definition, was wir unter LCP verstehen. Aufgrund der Heterogenität der auf dem Markt befindlichen Plattformen bedienen wir uns bewusst einer weit gefassten Definition von „(…) Low-Code als einen Entwicklungsansatz, der es ermöglicht, digitale Lösungen mit minimalem manuellem Programmieraufwand zu erstellen, und sehen Low-Code-Plattformen als benutzerfreundliche, visuelle Softwareentwicklungsumgebungen, die viele Aspekte der Code-Generierung und Anwendungsentwicklung automatisieren."[8]

Aufbauend auf dieser Definition betrachten wir die Vorteile und Herausforderungen, die mit der Nutzung von LCP und der Einführung von Citizen Development einhergehen können. Wir haben hierfür jeweils eine Auswahl der aus unserer Sicht wichtigsten Punkte zusammengetragen.

Vorteile / Chancen



- Schnellere Entwicklung und Bereitstellung von auf die Bedürfnisse von Fachbereichen zugeschnittenen Applikationen und Prozessen[6, 20]
- Ermöglichung digitaler Innovation durch Mitarbeiter:innen ohne Programmierkenntnisse, aber mit fundiertem Domänenwissen[15]
- Reduzierung der Abhängigkeit von IT-Abteilungen[6] und eine stärkere Konzentration der professionellen Entwickler:innen auf komplexere Aufgaben bezüglich Backend-Services und Integrationen in Umsysteme [8, 15]

Nachteile / Herausforderungen



- Trotz der möglichen Kosteneinsparungen durch den Einsatz von LCP können diese durch andere Kosten bspw. durch Lizenzen oder Schulungen aufgewogen werden[2]
- Entwickler:innen haben in vielen Fällen keinen direkten Zugriff auf den Quellcode der LCP und ihrer bereitgestellten Funktionalitäten, was die Bewertung der vorhandenen Sicherheits- und Schutzmaßnahmen erheblich einschränkt[26]
- Low-Code-Plattformen können bei komplexen oder spezialisierten Softwarelösungen schnell an ihre Grenzen kommen [31]. Darüber hinaus müssen die begrenzte Portabilität von Low-Code-Plattformen [14] berücksichtigt und die nahtlose Kommunikation der neuen Anwendungen mit den bestehenden Systemen sichergestellt werden, um Dateninkonsistenzen und Betriebsunterbrechungen zu vermeiden [9].

Chancen von Low-Code in der Verwaltung

Obwohl das Thema Digitalisierung an Fahrt gewonnen hat, konnte Deutschland im internationalen Vergleich bisher weder mithalten noch die Erwartungen der Bürgerschaft vollends erfüllen[11]. Zwar hat sich in den letzten Jahren dank Onlinezugangs- und Registermodernisierungsgesetz einiges getan, um bürgernahe Verwaltungsleistungen als Onlinelösungen bereitzustellen[12], gleichzeitig wurde die Binnendigitalisierung der Behörden bislang eher vernachlässigt, so dass die „gängigen Verwaltungsabläufe (…) in vielen Fällen noch im Zeichen von Aktenlaufplänen, sequenzieller Bearbeitung und strikt hierarchisch geordneten Arbeitsverhältnissen" stehen[7]. Dieser Medienbruch zwischen für Bürger:innen bereitgestellten Lösungen und innerbehördlichen Abläufen führt insbesondere dazu, dass Verwaltungsmitarbeitende keine Entlastung finden[25]. Derartige Verhältnisse, mitunter gepaart mit fehlender IT-Governance sowie mangelnder Abstimmung zwischen Fach- und IT-Abteilungen, können die Entstehung und Ausweitung von Schatten-IT ermöglichen und zahlreiche Risiken, insbesondere in puncto Datensicherheit und Datenschutz, begünstigen [3, 10, 29, 35].

Zentral verwaltete LCP können durchaus einen Beitrag zur Reduktion solcher Risiken durch Schatten-IT beitragen, sofern diese für die Mitarbeitenden so niederschwellig wie möglich sind[40]. Niederschwelligkeit bezieht sich einerseits auf die Gestaltungsmöglichkeiten der LCP, aber auch die organisatorischen Rahmenbedingungen, unter denen Anwendungen und Prozesse entstehen dürfen. Sind diese zu einschränkend, wird die Adoption erschwert oder gar verhindert[ebenda].

Diese und weitere Erfolgsfaktoren erlauben es, Verwaltungsmitarbeitenden zu Citizen Developern zu werden und die Binnendigitalisierung voranzutreiben. Im Fokus sollte dabei die Realisierung der Verwaltungsanwendungen stehen, die „zu klein für die Entwicklung eines eigenen aufwändigen und kostenintensiven IT-Fachverfahrens [sind], [jedoch] zu wichtig und häufig, um sie bei der Digitalisierung außer Acht zu lassen (…)"[24].



Marktanalyse

Um einen groben Überblick über den heterogenen Markt der LCP zu gewinnen, haben wir fünf Plattformen ausgewählt, die aus unserer Sicht die spezifischen Anforderungen des öffentlichen Sektors, darunter Sicherheit, Skalierbarkeit, Benutzerfreundlichkeit und Integrationsmöglichkeiten, berücksichtigen.

OutSystems erlaubt die schnelle Entwicklung komplexer und skalierbarer Anwendungen. Die Plattform bietet eine leistungsstarke visuelle Entwicklungsumgebung und umfangreiche Integrationsmöglichkeiten, die es Nutzenden ermöglichen, Anwendungen effizient zu erstellen und zu verwalten. Hinzu kommt eine starke Community[28].

Die Lernkurve ist für Anfänger:innen vergleichsweise steil und auch die Kosten sind im Vergleich mit anderen Plattformen relativ hoch.

Mendix bietet eine Low-Code-Plattform, die durch ihre Kollaborationsfunktionen und Flexibilität besticht. Die Plattform ermöglicht es Teams, gemeinsam Anwendungen zu entwickeln. Auch Mendix hat eine große Community und verfügt über einen umfangreichen Marktplatz für Erweiterungen[22].

Analog zu OutSystems ist auch in Mendix die Lernkurve steiler als bei anderen Plattformen und auch die Kosten sind vergleichsweise hoch.

Microsoft Power Platform integriert sich nahtlos in das Microsoft-Ökosystem, bietet starke Sicherheitsfunktionen und ermöglicht die Erstellung und Verteilung benutzerdefinierter Anwendungen mit einer intuitiven Drag & Drop-Oberfläche. Die Plattform ist besonders für Unternehmen geeignet, die bereits Microsoft-Dienste nutzen. Von den hier vorgestellten Plattformen ist Microsoft Power Platform die einzige, die keine reine on-premise-Option bietet[23].

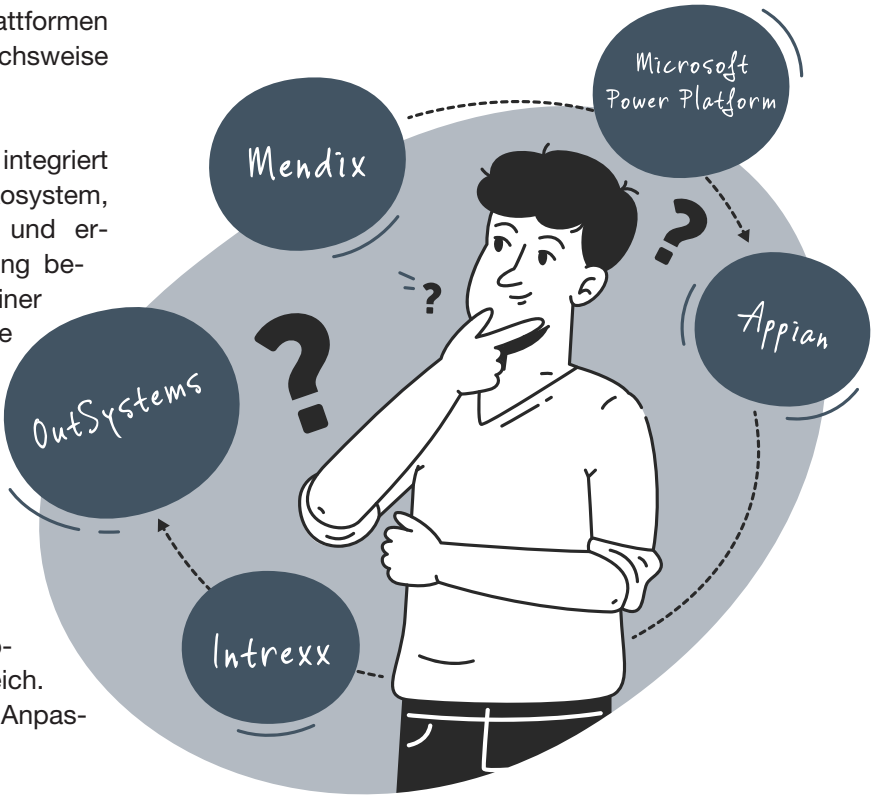
Die Abhängigkeit vom Microsoft-Ökosystem ist Stärke und Schwäche zugleich. Die Plattform bietet begrenztere Anpassungsmöglichkeiten durch die Bereitstellung als Software as a Service (SaaS), durch die Integration in die weitverbreitete Microsoft-365-Umgebung jedoch eine der größten weltweiten Communities.

Appian konzentriert sich auf die Automatisierung von Geschäftsprozessen und bietet eine Plattform mit umfassenden Sicherheitsfunktionen, die die schnelle Anwendungsentwicklung und die Integration verschiedener Systeme und Datenquellen unterstützt[1].

Eine höhere Komplexität in der Anwendung und höhere Kosten bei umfangreicher Nutzung sollten bedacht werden.

Intrexx bietet eine benutzerfreundliche Entwicklungsumgebung zur effizienten Erstellung und Verwaltung von Webanwendungen und Geschäftsprozessen. Die Plattform zeichnet sich durch ihre umfangreichen Integrationsmöglichkeiten und die Unterstützung für verschiedene Datenquellen aus[13].

Die Plattform ist kosteneffizient, ist jedoch weniger bekannt als andere Plattformen und bietet nur eine sehr begrenzte Community-Unterstützung.



Kriterien und Bewertungsstufen

- **Initialer Entwicklungsaufwand[1]:**
  Gering: Laut Hersteller vollständige KI-App-Generierung (Datenmodell, Oberfläche, Logik) aus Text/Dokumenten oder es gibt umfangreiche Starter-Apps; Betriebsumgebung inklusive.
  Mittel: Visuelle Baukästen/Vorlagen vorhanden, aber Datenmodell bzw. Infrastruktur weitgehend manuell.
  Hoch: Installation, Einrichtung oder Programmierung nötig.

- **Integration/Bereitstellung von Services[2]:**
  Umfangreich: REST, plus mindestens ein weiterer Standard (SOAP, OData, Datenbank) und fertige Unternehmenskonnektoren (z. B. SAP, Salesforce).
  Standard: REST-/Datenbankanbindung, kaum fertige Konnektoren.
  Begrenzt: Nur einzelne Integrationswege.

- **Individuelle Anpassung durch Code:**
  Vollständig: Eigener Code in Allzwecksprache samt Fremdbibliotheken in Frontend und Backend.
  Teilweise: Nur eine Ebene oder stark eingeschränkt.
  Keine: Keine Code-Ebene.

- **Laufzeitanalyse:**
  Umfangreich: Protokolle, Messwerte/Dashboards sowie Tracing oder Export an Monitoring-Werkzeuge (APM, OpenTelemetry).
  Standard: Protokolle und Basismetriken.
  Begrenzt: Nur Protokolle.

- **Feature-Umfang für komplexe Anwendungen:**
  Umfangreich: Leader im Gartner Magic Quadrant Enterprise Low-Code 2025 oder Forrester Wave Low-Code Q2 2025.
  Mittel: Dort bewertet, aber kein Leader.
  Begrenzt: Laut Hersteller nur für einfache/abteilungsbezogene Apps.

- **Weiterentwicklung ohne Mehraufwand:**
  Ja: Versionsverwaltung oder Änderungsfolgenanalyse sowie Hersteller-Updates ohne Neubau.
  Eingeschränkt: Teilweise erfüllt oder Neubau bei Generationswechsel.
  Nein: Nicht erfüllt.

- **Testbarkeit der Anwendungen:**
  Umfangreich: Herstellerwerkzeuge für automatisierte Logik- und Oberflächentests.
  Standard: Herstellerwerkzeuge für nur eine Ebene.
  Begrenzt: Nur manuell oder über Drittanbieter.

- **Debugging und Fehlerdiagnose:**
  Umfangreich: Debugger (Haltepunkte, Einzelschritte, Variablen) plus Betriebsdiagnose (Logs/Traces oder Remote-Debugging).
  Standard: Nur eines davon.
  Begrenzt: Nur Logs.

- **Automatisierung durch CI/CD:**
  Umfangreich: Integrierte Deployment-Pipeline und externe CI/CD-Anbindung (z. B. Jenkins, Azure DevOps, GitHub).
  Standard: Nur eines davon.
  Begrenzt: Manuelle Deployments.

- **Skalierbarkeit und Ausfallsicherheit:**
  Sehr gut: Automatische Skalierung oder zonenübergreifende Hochverfügbarkeit plus vertragliche Verfügbarkeit ≥ 99,9 % (tarifabhängig).
  Gut: Skalierung dokumentiert, entsprechende Zusage fehlt.
  Eingeschränkt: Keine Skalierungs-/Hochverfügbarkeitsmechanismen dokumentiert.

- **KI und Automatisierung:**
  Umfangreich: KI-gestützte Entwicklung, KI in Apps (Agenten, GenAI) und Prozessautomatisierung.
  Teilweise: Nicht alle drei belegt.
  Keine: Nichts davon belegt.

- **Kollaboration:**
  Umfangreich: Parallele Entwicklung mit Versionskontrolle/Merging plus Teamfunktionen (Echtzeit-Kollaboration, User Stories).
  Standard: Mehrbenutzerentwicklung mit Versionskontrolle.
  Begrenzt: Einzelbearbeitung/Sperren.

- **Reporting:**
  Umfangreich: Eigenes BI-/Analysewerkzeug integriert oder lizenzierbar.
  Standard: Diagramm-/Report-Komponenten für Apps.
  Begrenzt: Keine entsprechenden Funktionen.

- **Sicherheit und Benutzerverwaltung:**
  Umfangreich: ISO 27001 plus SOC 2 oder BSI C5 sowie Rollenrechte oder Identity-Provider-Anbindung.
  Gut: ISO 27001 und Rollenrechte.
  Begrenzt: Keine Zertifizierung belegt.

- **Barrierefreiheit[3]:**
  Umfangreich: Konformitätsbericht (VPAT/ACR) und dokumentierte Funktionen (ARIA, Fokus, Kontrast, Tastatur).
  Standard: WCAG-bezogene Funktionen/Anleitungen ohne Bericht.
  Begrenzt: Nur Hinweise oder Ankündigungen.

- **Für alle Kriterien gilt:**
  k.Q.: keine belastbare Quelle gefunden, nicht bewertbar.

[1] Bis zur ersten nutzbaren App
[2] Anbindung externer Systeme und Bereitstellung eigener Schnittstellen
[3] Unterstützung für WCAG 2.1/2.2 AA (über EN 301 549 Grundlage der BITV 2.0 für Bundesbehörden). Bewertet wird die Plattform, nicht die fertige App.

Kriterium	OutSystems	Mendix	Power Platform	Appian	Intrexx IX25
Initialer Entwicklungsaufwand	Gering	Gering	Gering	Gering	Mittel
Integration/ Bereitstellung von Services	Umfangreich	Umfangreich	Umfangreich	Umfangreich	Standard
Individuelle Anpassung durch Code	Vollständig	Vollständig	Vollständig	Vollständig	Keine
Laufzeitanalyse	Umfangreich	Umfangreich	Umfangreich	Umfangreich	Umfangreich
Feature-Umfang für komplexe Anwendungen	Umfangreich	Umfangreich	Umfangreich	Umfangreich	k.Q.
Weiterentwicklung ohne Mehraufwand	Ja	Ja	Ja	Ja	Ja
Testbarkeit der Anwendungen	Standard	Standard	Standard	Umfangreich	Begrenzt
Debugging zur Fehlerdiagnose	Umfangreich	Umfangreich	Standard	Standard	Standard
Automatisierung durch CI/CD	Umfangreich	Umfangreich	Umfangreich	Umfangreich	Standard
Skalierbarkeit und Ausfallsicherheit	Sehr gut	Sehr gut	Sehr gut	Sehr gut	Gut
KI und Automatisierung	Umfangreich	Umfangreich	Umfangreich	Umfangreich	Teilweise
Kollaboration	Standard	Umfangreich	Umfangreich	Standard	Umfangreich
Reporting	Standard	Standard	Umfangreich	Umfangreich	Standard
Sicherheit und Benutzerverwaltung	Umfangreich	Umfangreich	Umfangreich	Umfangreich	Gut
Barrierefreiheit	Standard	Standard	Standard	Standard	Standard

Handlungsleitfaden zur Einführung von Low-Code-Plattformen

Low-Code-Canvas

Unterstützung bei der Auswahl der richtigen Plattform

Die Heterogenität der vorhandenen LCP erschwert nicht nur eine einheitliche Definition ihrer Funktionalitäten[8], sondern vor allem die Auswahl der für den Anwendungsfall richtigen Plattform. Vor der Anschaffung einer LCP sollte darüber hinaus gründlich geprüft werden, ob die vorhandene IT-Infrastruktur bereits entsprechende Plattformen bietet.[ebenda]

Richtungsgebend kann sein, die wichtigsten Anwendungsfälle zu identifizieren: Je nach Anwendungsfall „(…) könnten spezialisierte Low-Code-Plattformen besser geeignet sein als allgemeine ‚general-purpose' Lösungen."[ebenda]

Auch die Grenzen der Plattformen spielen bei der Entscheidungsfindung eine wichtige Rolle. Typische Anwendungsfelder von LCP sind „schnelle Prototypenentwicklung, Anwendungen mit begrenzter Komplexität, Automatisierung von Geschäftsprozessen und Anwendungen, die von Fachabteilungen entwickelt und genutzt werden"[15]. Bei komplexen, spezialisierten Anwendungen, die unter Umständen eine tiefgreifende Integration mit Umsystemen erfordern, sollten tradierte Methoden der Softwareentwicklung bevorzugt werden.[2, 31, 34]



Sind diese Punkte geklärt, ist die Evaluation der möglichen LCP der nächste Schritt. Einen Leitfaden bietet der Low-Code-Canvas, der vom Fraunhofer-Institut für offene Kommunikationssysteme entwickelt wurde. Er ermöglicht die systematische Betrachtung von LCP unter verschiedenen Gesichtspunkten, aufgeteilt in vier Themengruppen[18]:

- **Modellierung:** Bezieht sich vor allem auf die Anwendersicht auf eine Plattform, z. B. im Hinblick auf Anwendungs-, Anpassungs- und Gestaltungsmöglichkeiten
- **Plattform:** Umfasst technische Aspekte wie bspw. Betriebsmöglichkeiten (on-premise oder SaaS), Interoperabilität mit anderen Plattformen oder Erweiterbarkeit
- **Qualität:** Betrachtet neben Aspekten wie Lizenzen, Community oder Dokumentation auch nicht-funktionale Anforderungen wie Barrierefreiheit, Datenschutz oder Sicherheit
- **Zielgruppe:** Fokussiert sich auf die Rollen, die bei der Einführung und Nutzung einer LCP benötigt werden, aber auch auf die erforderlichen Rahmenbedingungen

Der ÖFIT-Wegbereiter „Startklar für Low Code" vom Kompetenzzentrum Öffentliche IT ist ein konkreter Startpunkt für die Nutzung des Low-Code-Canvas. Er übersetzt die im Low-Code-Canvas dargelegten Themenkomplexe in Fragenkataloge, um bei der Evaluation der wichtigsten Aspekte bei der Auswahl einer LCP und der Zusammenstellung eines geeigneten Teams zu unterstützen[39].

Organisatorische Voraussetzungen für eine erfolgreiche Einführung von LCP

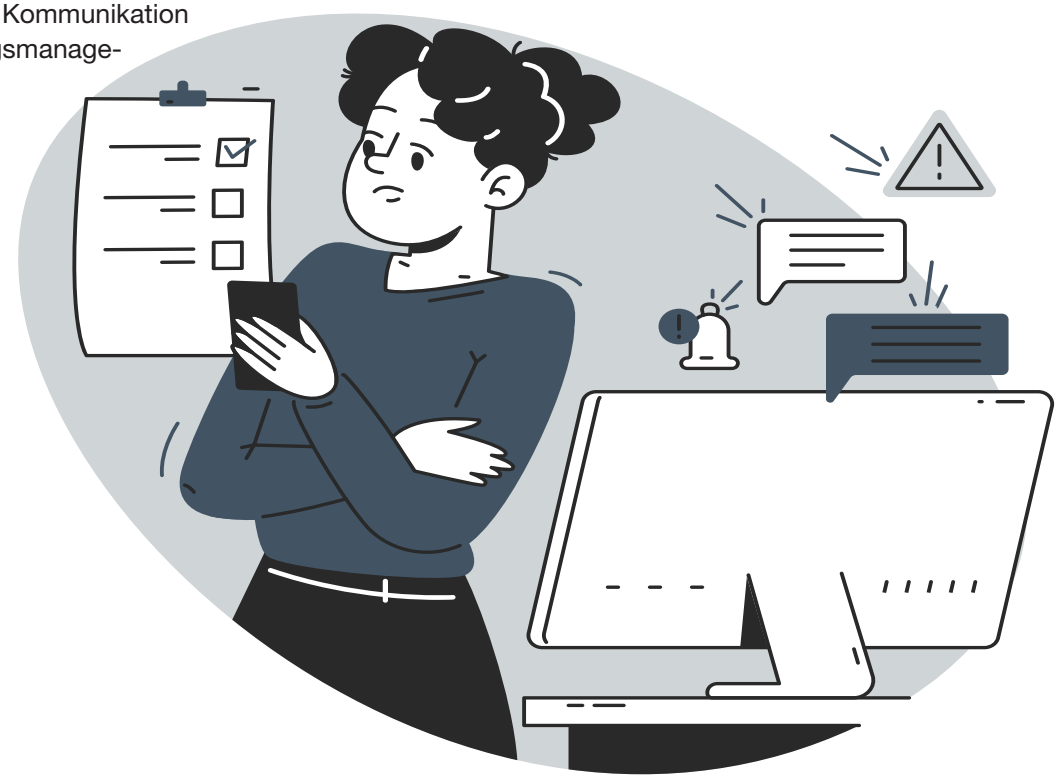
Die Einführung von LCP erfordert neben der technischen Integration in die IT-Landschaft auch organisatorische Veränderungen, um Schatten-IT einzudämmen statt zu befeuern, sowie Datenschutz und -sicherheit zu gewährleisten. Auch ein kultureller Wandel ist vonnöten, um den angehenden Citizen Developern genügend Freiraum zu geben, die neuen Möglichkeiten explorativ zu erschließen und in enger Abstimmung mit der IT-Abteilung innerbehördliche Digitalisierung auf den Weg zu bringen. Die folgenden Punkte geben einen Überblick über Arbeiten, in denen konkrete Handlungsempfehlungen formuliert wurden, um diesen Wandel zu ermöglichen und per Citizen Development eine Digitalisierung von innen heraus zu starten.

Binzer et al. geben in ihrer Arbeit sechs konkrete Handlungsempfehlungen zur effektiven Einführung von LCP und Citizen Development. Dies reicht von der Bewertung der bereits vorhandenen IT-Landschaft, um ungenutzte Potenziale zu erkennen und auszuschöpfen, bis zu Governance-Strategien, die als Leitplanke dienen, ohne zu stark einzuschränken, jedoch Sicherheit und Qualität der entwickelten Lösungen zu gewährleisten. Hervorgehoben werden die Wichtigkeit der Zusammenarbeit zwischen IT- und Fachabteilungen sowie die Faktoren Schulungen, Kommunikation und Erwartungsmanagement[4].

In der Arbeit von ***Elshan et al.*** wird die Wichtigkeit umfassender Schulungen von Citizen Developern hervorgehoben, „(…) die nicht nur die grundlegende Nutzung der Plattformen abdecken, sondern auch sicherheitsrelevante Best Practices vermitteln. Unternehmen sollten sicherstellen, dass diese Schulungen verpflichtend sind und regelmäßig aktualisiert werden, um auf neue Bedrohungen und Entwicklungen in der IT-Sicherheit zu reagieren."[8]

Die Arbeit von ***Leible et al.*** fokussiert sich auf mitarbeitergetriebene Digitalisierung durch Citizen Developer. Es werden Förder- und Hemmfaktoren beleuchtet sowie Handlungsempfehlungen formuliert, um die identifizierten Hemmnisse abzubauen. Auch hier wird die Wichtigkeit einer Governance-Strategie, der abteilungsübergreifenden Kommunikation und der Befähigung der Citizen Developer durch Trainings hervorgehoben.[17]

Einen spannenden Einblick in den digitalen Transformationsprozess durch Low-Code und Citizen Development gibt der internationale Energiekonzern Shell in einer detaillierten Arbeit. Shell beschreibt umfänglich die verschiedenen Phasen der Einführung von Citizen Development, mögliche Stolpersteine und entsprechende Handlungsempfehlungen. Darüber hinaus werden Einblicke in den LCP-Auswahlprozess, verwendete Technologien und vieles mehr gewährt.[6]



Herausforderung Vendor Lock-In

Vendor Lock-In beschreibt die Situation, in der Kunden oder Organisationen gezwungen sind, ein Produkt oder Service unabhängig von dessen Qualität zu nutzen, da ein Wechsel technisch herausfordernd und mitunter kostenintensiv sein kann[38].

Low-Code-Plattformen sind in der Regel proprietär und nicht quelloffen. Zudem erschweren fehlende Standards die Interoperabilität und Portabilität zwischen den Plattformen oder anderen IT-Systemen[27, 32].

Neben diesen Problemen können sich weitere Risiken ergeben, bspw. wenn die Entwicklung einer Plattform stagniert oder sogar komplett eingestellt (wie z. B. AppMaker von Google[5]) wird und man gezwungen ist, auf eine andere Plattform zu migrieren. Daten und selbst entwickelte Applikationen zu portieren kann hier zu einem erheblichen Kraftakt werden[37].

Die Studienlage zum Thema Vendor Lock-In bezüglich Low-Code ist noch recht dünn, wohingegen das Problem in Bezug auf Cloud Computing bereits recht gut untersucht ist. Mögliche Maßnahmen gegen Vendor Lock-In lassen sich gut auf LCP adaptieren, da hier ähnliche Probleme auftreten können.

Diese Maßnahmen können wie folgt aussehen:

- einer der wichtigsten Aspekte ist, in Frage kommende Plattformen in Bezug auf Interoperabilität mit anderen Technologien, die Nutzung von Standards sowie Lizenzmodelle genau zu prüfen[27, 37], um bspw. sicherzustellen, dass die Integration mit bestehenden Systemen reibungslos funktioniert und bezüglich der Kosten keine unangenehmen Überraschungen entstehen

- ein weiterer kritischer Entscheidungsfaktor sollte sein, wie viel Kontrolle man über seine Daten hat, so dass sich diese bei Bedarf mit nachvollziehbarem Aufwand portieren lassen[37]

- es kann sinnvoll sein, je nach Bedarfen und Anwendungsfällen, mehrere Plattformen zu verwenden, sofern sich diese deutlich in ihren Schwerpunkten unterscheiden. In so einer Multiplattform-Architektur sollten die jeweiligen Plattformen ausschließlich über wohldefinierte und entwicklungsfähige APIs miteinander kommunizieren, so dass ein Plattformwechsel leichter möglich ist[33]



Fazit und Ausblick

Wenn bis zum Jahr 2030 mehr als ein Drittel der im öffentlichen Sektor Beschäftigten in Rente gehen, wird sich der bereits heute spürbare Fachkräftemangel deutlich verschärfen[36]. Dies bedeutet neben der Reduzierung von personellen Kapazitäten nicht zuletzt eine Abwanderung von Fachwissen. Die gleichzeitig wenig ausgebaute innerbehördliche Digitalisierung verschärft diese Lage und wird den Leistungsdruck auf Verwaltungsmitarbeitende sukzessive erhöhen. Low-Code-Plattformen bieten hier einen potenziellen Ausweg: Bei der Auswahl der passenden Plattform können Fachexpert:innen mit entsprechender Unterstützung zu Citizen Developern werden, die ihre eigenen Prozesse digitalisieren. So fließt das kritische Domänenwissen direkt in die Applikationen und automatisierten Prozesse ein und kann im Rahmen einer Innovationen fördernden Kultur zu einer echten Entlastung werden und den Fachkräftemangel zumindest abmildern. Professionelle Entwickler:innen können unterstützen und sich gleichzeitig komplexeren Aufgaben widmen.

Die richtige Low-Code-Plattform zu finden erfordert hierfür eine ganzheitliche Betrachtung der auf dem Markt befindlichen Plattformen wie auch die genaue Ermittlung der eigenen Bedarfe, um die wünschenswerten Effizienzsteigerungen zu erreichen. Neben den technischen sind auch organisatorische Voraussetzungen zu schaffen und kontinuierlich weiterzuentwickeln.

Wir haben fünf Plattformen vorgestellt und miteinander verglichen. Für einen holistischen Blick auf die benötigte Technologie, die eigenen Bedarfe und Ansprüche bieten der Low-Code-Canvas und die darauf aufbauenden Arbeitsmaterialien des Kompetenzzentrums Öffentliche IT eine gute Unterstützung bei der Auswahl der richtigen Plattform. Für die notwendigen organisatorischen Veränderungen haben wir eine Auswahl an Arbeiten zusammengestellt, die hilfreiche Handlungsempfehlungen formulieren. Zum Umgang mit einem befürchteten Vendor Lock-In haben wir Strategien präsentiert, die zur Vermeidung beitragen können.

Der Low-Code-Markt wird sich in den nächsten Jahren weiter diversifizieren und vor allem die jüngsten und künftig zu erwartenden Entwicklungen generativer künstlicher Intelligenzen werden diesem Segment deutlich ihren Stempel aufdrücken. Haben Low-Code-Plattformen bereits zu einer Demokratisierung der IT beigetragen, so werden KI-Werkzeuge diese Entwicklung weiter voranbringen. Für Citizen Developer werden sich somit neue Chancen für einen schnellen Einstieg in LCP und die Verwirklichung von Ideen eröffnen. Für die Verwaltung eröffnen sich hierdurch spannende Zukunftsaussichten.



Über uns

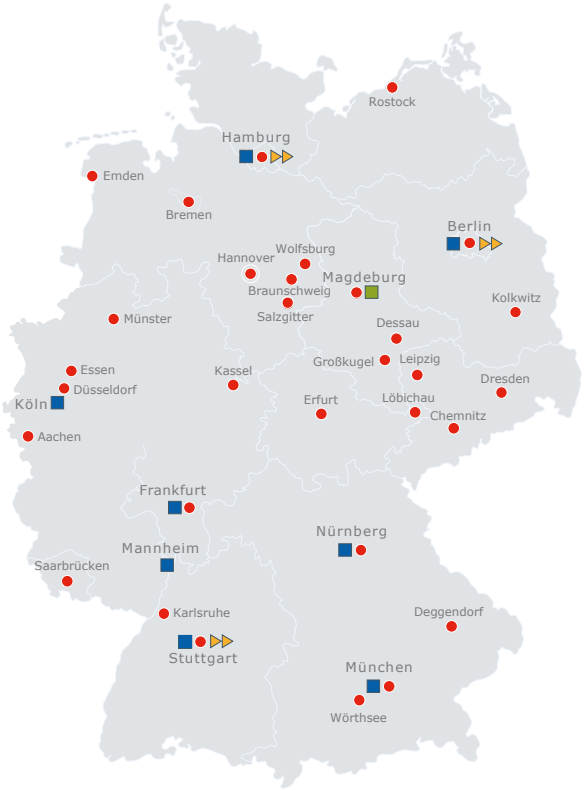
Über die bridgingIT-Gruppe

Die BridgingIT GmbH wurde 2008 gegründet und bildet gemeinsam mit den Tochterunternehmen fastahead GmbH & Co. KG und CraftingIT GmbH die bridgingIT-Gruppe. Mit über 700 Mitarbeitenden an 9 Standorten in Deutschland unterstützt die Gruppe Unternehmen verschiedenster Branchen bei Veränderungs- und Digitalisierungsprojekten.

Als unabhängige Technologie- und Managementberatung bietet die bridgingIT-Gruppe ein umfassendes End-to-End-Portfolio – von der strategischen Beratung über die technologische Umsetzung bis zum Betrieb. Sie verbindet Business- und IT-Kompetenz, um digitale Transformation ganzheitlich, effizient und praxisorientiert zu gestalten. BridgingIT steht für technologische Lösungen und Prozessoptimierung, fastahead für digitale Geschäftsmodelle und craftingIT für individuelle IT-Services.

Ende 2023 fusionierte die bridgingIT-Gruppe mit der SPIE Germany Switzerland Austria GmbH, die mit 75,1 % der Stimmrechtsanteile als Mehrheitsgesellschafter einstieg. Die Dienstleistungen und Lösungen bleiben unverändert, was die anhaltende Stabilität und das Engagement für unsere Kunden bekräftigt.

Mehr zur bridgingIT-Gruppe und unseren Projekten unter: **www.bridging-it-gruppe.de**



bridging IT
Ein Unternehmen von SPIE

crafting IT
Ein Unternehmen von SPIE

fastahead
Ein Unternehmen von SPIE

SPIE ICS*

*Information & Communication Services

Schauen Sie vorbei und folgen Sie uns!

Glossar

Automatisierung: Automatisierung in Bezug auf Software und Geschäftsprozesse bezeichnet den Einsatz von Technologie, um wiederkehrende Aufgaben und Abläufe ohne menschliches Eingreifen effizient und fehlerfrei auszuführen
..... **7, 8, 9, 10**

Backend-Services: Serverseitige Dienste und Anwendungen, die die Logik, Datenverarbeitung und Datenverwaltung einer Softwarelösung bereitstellen und typischerweise über APIs mit anderen Systemen kommunizieren
..... **5**

Best Practices: Bezeichnet bewährte und allgemein anerkannte Methoden oder Verfahren, die sich als besonders effektiv und effizient erwiesen haben und daher als Standard für die Durchführung bestimmter Aufgaben oder Prozesse gelten
..... **11**

CD: Continuous Deployment (CD) ist ebenso wie Continuous Integration eine Softwareentwicklungspraxis, bei der Codeänderungen automatisch nach erfolgreichen Tests in die Produktionsumgebung überführt werden, um neue Funktionen und Updates kontinuierlich und ohne manuelle Eingriffe bereitzustellen
..... **8, 9**

CI: Continuous Integration (CI) ist eine Softwareentwicklungspraxis, bei der Codeänderungen regelmäßig und automatisch in ein gemeinsames Repository integriert und durch automatisierte Builds und Tests überprüft werden, um Fehler frühzeitig zu erkennen und die Qualität des Codes zu gewährleisten
..... **8, 9**

Citizen Developer: Mitarbeitende ohne formale Programmierausbildung, die mithilfe von Low-Code- oder No-Code-Plattformen eigenständig Anwendungen entwickeln, um spezifische Geschäftsanforderungen zu erfüllen
..... **4, 6, 11, 13**

Community: In Bezug auf Software bezeichnet der Begriff eine Gruppe von Entwickler:innen, Nutzer:innen und Enthusiast:innen, die zusammenarbeiten, um Softwareprojekte zu entwickeln, zu verbessern und zu unterstützen, oft durch den Austausch von Code, Feedback und Ressourcen
..... **7, 10**

Debugging: Der Prozess des Identifizierens, Analysierens und Behebens von Fehlern oder Bugs in einem Softwareprogramm, um dessen korrekte Funktion sicherzustellen
..... **8, 9**

Drag & Drop-Editor: Ein benutzerfreundliches Werkzeug, das es ermöglicht, Elemente durch Ziehen und Ablegen mit Maus oder Trackpad auf einer Oberfläche zu platzieren und so Inhalte oder Layouts ohne Programmierkenntnisse zu erstellen und anzupassen
..... **4**

general-purpose: Anwendungen oder Systeme, die für eine Vielzahl von Aufgaben und Anwendungsbereichen entwickelt wurden und nicht auf eine spezifische Funktion oder Branche beschränkt sind
..... **10**

Governance: Bezeichnet die Strukturen, Prozesse und Regeln, durch die eine Organisation oder ein System gesteuert und kontrolliert wird, um Ziele zu erreichen, Verantwortlichkeiten zu definieren und die Einhaltung von Richtlinien zu gewährleisten
..... **6, 11**

Onlinezugangsgesetz: Das Onlinezugangsgesetz (OZG) verpflichtet Bund, Länder und Kommunen in Deutschland dazu, ihre Verwaltungsleistungen bis Ende 2022 auch online über Verwaltungsportale anzubieten, um den Zugang zu öffentlichen Dienstleistungen zu erleichtern und zu digitalisieren
..... **6**

on-premise: Bezeichnet die Installation und den Betrieb von Software oder IT-Infrastruktur innerhalb der eigenen Räumlichkeiten eines Unternehmens, im Gegensatz zur Nutzung von Cloud-Diensten, die extern gehostet werden
..... **7, 10**

proprietär: Software, Technologien oder Systeme, die von einem Unternehmen entwickelt und kontrolliert werden, wobei der Quellcode oder die technischen Spezifikationen nicht öffentlich zugänglich sind und deren Nutzung, Modifikation und Verbreitung durch Lizenzen eingeschränkt ist
..... **12**

Registermodernisierungsgesetz: Das Registermodernisierungsgesetz (RegMoG) zielt darauf ab, die Registerführung in Deutschland zu modernisieren, indem es eine einheitliche und effiziente Datenhaltung sowie den Austausch von Daten zwischen verschiedenen Registern der öffentlichen Verwaltung ermöglicht
..... **6**

Schatten-IT: Bezeichnet die Nutzung von IT-Systemen, -Geräten, -Software und -Diensten innerhalb einer Organisation ohne das Wissen oder die Zustimmung der zentralen IT-Abteilung, was oft zu Sicherheitsrisiken und Compliance-Problemen führen kann
..... **4, 6, 11**

Software as a Service: Ein Bereitstellungsmodell, bei dem Softwareanwendungen über das Internet als Dienstleistung angeboten werden, so dass Nutzer auf die Software zugreifen können, ohne sie lokal installieren oder warten zu müssen
..... **7, 10**

Abkürzungsverzeichnis

CD ..... **Continuous Deployment**

CI ..... **Continuous Integration**

IT ..... **Informationstechnik**

KI ..... **Künstliche Intelligenz**

LCP ..... **Low-Code-Plattform**

ÖFIT ..... **Kompetenzzentrum Öffentliche IT**

SaaS ..... **Software as a Service**

Quellenverzeichnis

[1] „Appian". Appian. Web. 09.10.2024. https://appian.com/
[2] Baumgarten, Christoph, Rainer Endl, and Silvan Stich. „Professionelle Softwareentwicklung mit Low Code optimieren–eine Fallstudie." HMD Praxis der Wirtschaftsinformatik (2024): 1-22.
[3] Behrens, Sandy. „Shadow systems: The good, the bad and the ugly." Communications of the ACM 52.2 (2009): 124-129.
[4] Binzer, Björn, and Till J. Winkler. „Die vier Phasen von Citizen Development-Initiativen: Treiber, Herausforderungen und Handlungsempfehlungen." HMD Praxis der Wirtschaftsinformatik (2024): 1-23.
[5] Bock, Alexander C., and Ulrich Frank. „Low-code platform." Business & Information Systems Engineering 63 (2021): 733-740.
[6] Carroll, Noel, and Mary Maher. „How Shell Fueled Digital Transformation by Establishing DIY Software Development." MIS Q. Executive 22.2 (2023): 3.
[7] „Digitalisierung in Deutschland – Lehren aus der Corona-Krise". Bundesministerium für Wirtschaft und Klimaschutz. Web. 03.10.2024. https://www.bmwk.de/Redaktion/DE/Publikationen/Ministerium/Veroeffentlichung-Wissenschaftlicher-Beirat/gutachten-digitalisierung-in-deutschland.pdf
[8] Elshan, Edona, and Björn Binzer. „Mehr als ein Trend?: Wie Low-Code die digitale Transformation unterstützt." HMD Praxis der Wirtschaftsinformatik (2024): 1-18.
[9] Elshan, Edona, Ernestine Dickhaut, and Philipp Alexander Ebel. „An investigation of why low code platforms provide answers and new challenges." (2023).
[10] György, Andreas, et al. „Exploring the shadows: IT governance approaches to user-driven innovation." (2012).
[11] Halsbenning, Sebastian. „Digitalisierung öffentlicher Dienstleistungen: Herausforderungen und Erfolgsfaktoren der OZG-Umsetzung in der Kommunalverwaltung." HMD Praxis der Wirtschaftsinformatik 58.5 (2021): 1038-1053.
[12] Handke, Stefan. „Digitalisierung der öffentlichen Verwaltung." (2021).
[13] „Intrexx". Intrexx GmbH. Web. 09.10.2024. https://www.intrexx.com/de/
[14] Käss, Sebastian, Susanne Strahringer, and Markus Westner. „Drivers and inhibitors of low code development platform adoption." 2022 IEEE 24th Conference on Business Informatics (CBI). Vol. 1. IEEE, 2022.
[15] Krejci, Désirée, Satu Iho, and Stéphanie Missonier. „Innovating with employees: an exploratory study of idea development on low-code development platforms." ECIS. 2021.
[16] Lebens, Mary, et al. „Rise of the citizen developer." Muma Business Review 5.12 (2021): 101-111.
[17] Leible, Stephan, et al. „Citizen Developer als Katalysatoren für die Entwicklung mitarbeitergetriebener Innovationen." HMD Praxis der Wirtschaftsinformatik (2024): 1-27.
[18] „Low-Code-Canvas – Gestaltungsaspekte von Low-Code-Plattformen". Fraunhofer FOKUS. Web. 26.09.2024. https://cdn0.scrvt.com/fokus/1e1537b8a98ef0f9/02f7bf97c774/DPS_Low_Code_Canvas_Faltblatt_A4_v08_Screen-1.pdf
[19] „Low Code Development Platform Market". Straits Research. Web. 02.10.2024. https://straitsresearch.com/report/low-code-development-platform-market
[20] Luo, Yajing, et al. „Characteristics and challenges of low-code development: the practitioners' perspective." Proceedings of the 15th ACM/IEEE international symposium on empirical software engineering and measurement (ESEM). 2021.
[21] „Mangel an IT-Fachkräften droht sich dramatisch zu verschärfen". Bitkom e.V. Web. 01.10.2024. https://www.bitkom.org/Presse/Presseinformation/Mangel-an-IT-Fachkraeften-droht-sich-zu-verschaerfen

[22] „Mendix". Mendix Technology B.V. Web. 09.10.2024. https://www.mendix.com/
[23] „Microsoft Power Platform". Microsoft. Web. 24.01.2025. https://www.microsoft.com/de-de/power-platform
[24] „MODUL-F: Low-Code-Baukasten zur Verwaltungsdigitalisierung". Fraunhofer FOKUS. Web. 03.10.2024. https://www.fokus.fraunhofer.de/de/dps/projekte/MODUL-F
[25] Müller, Torsten, and Bernd Heinrich Peper. „Die Automatisierung von Prozessen und Entscheidungen in der Verwaltung." Verwaltung, eGovernment und Digitalisierung: Grundlagen, Konzepte und Anwendungsfälle (2019): 241-250.
[26] Onoufriou, Alkisti. Safety Risks and Security Threats in Low-code Software Development. BS thesis. University of Twente, 2024.
[27] Opara-Martins, Justice, Reza Sahandi, and Feng Tian. „Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective." Journal of Cloud Computing 5 (2016): 1-18.
[28] „OutSystems". OutSystems. Web. 09.10.2024. https://www.outsystems.com/
[29] Panko, Raymond R., and Daniel N. Port. „End user computing: The dark matter (and dark energy) of corporate IT." Journal of Organizational and End User Computing (JOEUC) 25.3 (2013): 1-19.
[30] Prinz, Niculin, et al. „Citizen Development als Treiber der digitalen Transformation– Aktuelle Ansätze bei der Adoption von Low-Code Development Plattformen." HMD Praxis der Wirtschaftsinformatik (2023): 1-21.
[31] Prinz, Niculin, et al. „Two Perspectives of Low-Code Development Platform Challenges– An Exploratory Study." Pacific Asia Conference on Information Systems (PACIS 2022), July 5-9, 2022, Taipei-Sydney, Virtual Conference. Vol. 235. Association for Information Systems (AIS)-eLibrary, 2022.
[32] Sahay, Apurvanand, et al. „Supporting the understanding and comparison of low-code development platforms." 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, 2020.
[33] Setälä, Manu, Pekka Abrahamsson, and Tommi Mikkonen. „Elements of sustainability for public sector software–mosaic enterprise architecture, macroservices, and low-code." International Conference on Software Business. Cham: Springer International Publishing, 2021.
[34] Siemon, Dominik, and Damian Kedziora. „From Accountant to Software Developer– Transforming Employees with Robotic Process Automation (RPA)." (2023).
[35] Silic, Mario, and Andrea Back. „Shadow IT–A view from behind the curtain." Computers & Security 45 (2014): 274-283.
[36] Stern, Sebastian, et al. „Die Besten, bitte: Wie der öffentlichen Sektor als Arbeitgeber punkten kann." (2019).
[37] „Vendor lock-in: Understanding risks and how to avoid it". OutSystems. Web. 08.10.2024. https://www.outsystems.com/tech-hub/app-dev/vendor-lock-in/
[38] Vilpponen, Hannu, Mika Grundström, and Pekka Abrahamsson. „Exploring the critical success factors in social and health care information systems project procurement." Recent Developments in Engineering Research Vol. 8 (2020).
[39] „Wegbereiter: Startklar für Low Code". Kompetenzzentrum Öffentliche IT. Web. 29.09.2024. https://www.oeffentliche-it.de/-/wegbereiter-startklar-low-code
[40] Weiß, Erstprüfer Prof Dr Peter, and Zweitprüfer Prof Dr Frank Morelli. „Potenzialanalyse bei der Softwareentwicklung mit Low Code Plattformen in der öffentlichen Verwaltung."

Sie haben noch Fragen zum Thema?
Schreiben Sie uns:

lowcode@bridging-it.de

Impressum

Herausgeber
BridgingIT GmbH
N7, 5-6
68161 Mannheim
Deutschland

Geschäftsleitung
Simon Dillmann
Dr. Jochen Ruben (COO)

Kontakt
info@bridging-it.de

Registergericht
Amtsgericht Mannheim,
HRB 703853

USt-IdNr.
DE814938761

© 2026 BridgingIT GmbH

