

Werden sich Low-Code Development Platforms durchsetzen?

Sonja Klein

25.01.2023

Inhalt

1	Einleitung	3
1.1	Relevanz und Ziel der Arbeit	3
1.2	Methodik	4
1.3	Aufbau der Arbeit	4
2	Was sind LCDP?	5
2.1	Aufkommen und Definition	5
2.2	Wichtige Begriffe	7
2.2.1	Abgrenzung zu High-Code	7
2.2.2	Abgrenzung zu No-Code	7
2.3	Use Cases	8
2.4	Anwendungsfall: ERP und CRM	8
2.4.1	Anwendungsfall: MVPs	8
2.4.2	Anwendungsfall: Modernisierungen und Mobile-First	8
2.4.3	Anwendungsfall: Smart Process Apps	9
2.4.4	Anwendungsfall: neue Plattformen	9
2.5	Übersicht von Low Code Development Plattformen in der Praxis	9
3	Nachteile von LCDP	10
3.1	Steile Lernkurve	11
3.2	Geringe Skalierbarkeit	11
3.3	Geringe Interoperabilität und eine enge Bindung an einen Anbieter . .	11
3.4	Geringere Flexibilität und Anpassbarkeit	11
3.5	Geringe Teststandards	12
3.6	Auswahlschwierigkeiten	12

4	Vorteile und Potenziale von LCDP	12
4.1	Schnittstellenbereitstellung	13
4.2	Zusammenarbeit von Entwicklern und Fachkundigen	13
4.3	Beschleunigter Entwicklungsprozess	14
4.4	Reduzierte Kosten	15
4.5	Hohe Sicherheitsstandards und geringe Ausfallzeiten	15
4.6	Schwierigkeiten mit komplexer Geschäftslogik	16
4.7	Gesteigerte Flexibilität	16
4.8	Cloud-basiertes Deployment	17
5	Marktforschungen und Prognosen	17
6	Fazit und Zukunftsausblick	19

1 Einleitung

Low Code erlebte einen Aufschwung während der Pandemie. Neben Teams und Zoom wuchs auch die Relevanz von Low-Code Development Plattformen. Die außergewöhnlichen Umstände brachten noch nie dagewesene und dringende Herausforderungen mit sich und Low-Code-Lösungen waren oftmals die Antwort auf diese. Corona-Testzentren brauchten beispielsweise innerhalb kürzester Zeit Terminportale und mit Low-Code konnten diese in bis zu zwei Tagen entwickelt werden [AmyGlasscock.2021].

LC/NC gewann also immer mehr an Bedeutung. Das wird auch in einer staatlichen Umfrage von NASCIO deutlich. 2020 und 2021 wurden CIOs gefragt, welche neue Technologie ihrer Meinung nach in den nächsten 3 bis 5 Jahren die größte Bedeutung haben wird. Während 2020 LC/NC noch auf dem zweiten Platz mit 33%, direkt hinter AI (Künstlicher Intelligenz, RPA, ...) mit 61% lag, teilte LC/NC sich 2021 schon den ersten Platz mit 31%. AI machte in diesem Jahr nur noch 30% aus [AmyGlasscock.2021].

1.1 Relevanz und Ziel der Arbeit

Diese Hausarbeit befasst sich deshalb mit folgender Fragestellung:
Werden sich Low-Code Development Platforms durchsetzen?

Das Gebiet der Low Code Entwicklung ist vergleichsweise noch recht neu. Aus diesem Grund bietet diese Hausarbeit zu Beginn eine Einführung in die Thematik. Dies ermöglicht dem Leser einen leichten Einstieg. Im Laufe der Arbeit soll sich für den Leser herauskristallisieren, ob das Entwickeln mit Low Code Development Plattformen sich in der Zukunft durchsetzen wird.

Anhand der Hauptfragestellung können folgende Teilfragen abgeleitet werden:

- (i) Was sind Low Code Development Platforms?
- (ii) Wie lässt sich der Begriff von den Begriffen High-Code- und No-Code-Development-Plattformen abgrenzen?
- (iii) Welche Use Cases für Low Code Development Platforms gibt es?
- (iv) Welche Low Code Development Plattformen werden in der Praxis genutzt?
- (v) Was sind die Nachteile des Entwickelns mit Low Code Development Platforms?
- (vi) Was sind die Vorteile des Entwickeln mit Low Code Development Platforms?
- (vii) In welchem Ausmaß wird die Nutzung von LCDPs in der Zukunft prognostiziert?

Das Ziel dieser Hausarbeit ist es, dem Leser diese Fragen zu beantworten.

1.2 Methodik

Zur Beantwortung der Fragestellung wird die Methodik der Literaturrecherche verwendet. Hierbei liegt der Fokus auf folgenden Begrifflichkeiten: "Low Code" und "Low Code Development Platforms", "No Code" und "No Code Development Platforms". Um ein allgemeines Bild über die Thematik zu schaffen, waren erste Anlaufstellen für Quellen das WorldWideWeb und YouTube.

Auf Grundlage dessen wurde nach passender Literatur gesucht. In den online-Bibliotheken der Hochschule für Technik, Wirtschaft und Gestaltung Konstanz und in der Universität Konstanz (KonSearch) zum Thema Low Code war die Suche nur wenige relevante Treffer. Die Vorwärts- und Rückwärtssuche in Google Scholar, SpringerLink und AISnet.org brachten jedoch einige relevante Quellen.

Außerdem wurde während der Ausarbeitung mit weiteren Phrasen wie "Low Code Security", "Low Code for Legacy Code", "Low Code for MVP", etc. gesucht.

Neben wissenschaftlichen Veröffentlichungen wurde außerdem auch auf Internetdokumente zurückgegriffen. Dabei wurde auf die Auswahl der richtigen Quelle geachtet.

1.3 Aufbau der Arbeit

Die vorliegende Arbeit ist in fünf Kapitel untergliedert. Im aktuellen Kapitel wird dem Leser zunächst die Struktur, der Ablauf, das Verfahren sowie die Wichtigkeit dieser Hausarbeit erläutert.

Im zweiten Kapitel wird durch eine kurze Beschreibung des Aufkommens von Low Code Development Platforms an das Thema heran geführt. Anschließend wird mit der Erklärung von Use Cases und darauf folgend Beispielen von Low Code Development Plattformen ein Verständnis der aktuellen Situation geschaffen.

Das dritte Kapitel informiert dem Leser über mögliche Nachteile die das Entwickeln mit diesen Plattformen mit sich bringt. Die Vorteile des Entwickelns mit Hilfe von Low Code Plattformen werden im vierten Kapitel beleuchtet.

Den Abschluss der Arbeit umfasst ein Fazit, welches die Beantwortung der Leitfrage "Werden sich Low-Code Development Platforms durchsetzen?" als Aufgabe hat.

2 Was sind LCDP?

In diesem Kapitel wird zunächst kurz auf das Aufkommen von Low Code Entwicklungsplattformen eingegangen. Daraufhin wird der Begriff von den Termini High-Code- und No-Code-Development-Plattformen abgegrenzt. Um dem Leser einen Einblick über die aktuelle Situation zu schaffen werden als nächstes einige Use Cases für diese Plattformen erklärt und darauf hin Real-Welt Beispiele genannt.

2.1 Aufkommen und Definition

In den vergangenen Jahrzehnten gab es einige Branchentrends, die darauf abzielten, die Menge an handgeschriebenem Code zu reduzieren, der für die Erstellung von Software erforderlich ist [DiRuscio.2022].

Die erstmalige Verwendung des Begriffs Low-Code geht auf das Marktanalyseunternehmen Forrester im Jahr 2014 zurück, bei dem Low-Code-Entwicklungsplattformen (LCDP) als Plattformen, die eine schnelle Bereitstellung von Geschäftsanwendungen mit einem Minimum an manueller Programmierung und minimalen Vorabinvestitionen in Einrichtung, Schulung und Bereitstellung ermöglichen, definiert wurden. Im Jahr 2016 unterteilte Forrester die erfolgreichen Anwendungsbereiche der LCDPs in vier spezifische Anwendungsszenarien, d.h. Datenbank, Anfragebearbeitung, Prozess und Mobile-First [DiRuscio.2022].

Diese Definition entwickelte sich weiter und 2017 legte Forrester eine detailliertere Version vor, in der LCDPs als Produkte und/oder Cloud-Dienste für die Anwendungsentwicklung, die visuelle, deklarative Techniken anstelle von Programmierung verwenden und den Kunden zu geringen oder gar keinen Kosten in Form von Geld und Schulungszeit zu Beginn zur Verfügung stehen, wobei die Kosten im Verhältnis zum Geschäftswert der Plattformen steigen, beschrieben werden. Der Schwerpunkt liegt hier auf visuellen Schnittstellen und deklarativen Techniken. Die Fokussierung auf den Begriff "Plattform" wird als ein wichtiger Aspekt hervorgehoben, der diese Lösungen von der vorherigen Generation deklarativer Werkzeuge unterscheidet: LCDPs sind in erster Linie Plattformen, mit Funktionen für die Anwendungsbereitstellung, Lebenszyklusmanagement sowie Plattformmanagement [DiRuscio.2022].

Das Konzept der Bürgerentwickler geht auf ein Buch von James Martin aus dem Jahr 1982 mit dem Titel "Application Development Without Programmers" zurück. Es brachte eine Programmiersprache der vierten Generation hervor, die computergestützte Softwareentwicklungswerkzeuge verwendet und einer Person ohne Programmierkennt-

nisse, einem Bürgerentwickler, das Entwickeln ermöglichte. Diese Bewegung scheiterte jedoch, weil sie zu viel versprach und zu wenig hielt. Die Tools unterstützten zum Beispiel keine Best Practices. Versionskontrolle, Tests, Bereitstellung, Dokumentation und andere Best Practices für die Entwicklung mussten manuell durchgeführt werden. Aber da Low-Code- und No-Code-Plattformen heutzutage zunehmend zugänglicher werden und die Zusammenarbeit einfacher wird, ist dieses Konzept nun gängige Praxis in der Entwicklung mit LCDPs [KevinShuler.2023].

Gartner identifizierte 2016 außerdem ein ähnliches Segment, welches als Low-Code Application Platform (LCAP) bezeichnet wird. Insbesondere wurden Enterprise LCAPs vorgestellt, die darauf abzielen, Anwendungen der Unternehmensklasse zu erstellen, die eine hohe Leistung, Skalierbarkeit, Hochverfügbarkeit, Disaster Recovery, Sicherheit, SLAs, Ressourcennutzungstracking, technischen Support durch den Anbieter und API-Zugang zu und von lokalen und Cloud-Diensten erfordern [DiRuscio.2022].

Das Jahr 2017 markiert den Beginn einer Reihe von Übernahmen von LCDP-Anbietern. Appian startete im Mai 2017 einen Börsengang und erreichte 2018 eine Marktbewertung von fast 2 Milliarden US-Dollar. Im Juli 2018 erhielt OutSystems Investitionen in Höhe von 360 Millionen US-Dollar. Im August 2018 kündigte Siemens die Übernahme von Mendix für 730 Millionen US-Dollar an. Im Jahr 2017 schätzte Forrester die globale Marktgröße für LCDPs auf 3,8 Milliarden US-Dollar [DiRuscio.2022].

Forrester führt außerdem regelmäßig Umfragen durch, bei welchen Entwickler zur Nutzung von LCDPs befragt werden. 2018 gaben 23% der Entwickler an, Low-Code-Plattformen zu nutzen, und weitere 22% planten, dies innerhalb eines Jahres zu tun. Im Jahr 2019 nutzten 37% der Entwickler Low-Code-Produkte oder planten, sie zu nutzen [DiRuscio.2022].

Im Jahr 2021 boten die meisten großen Cloud-Anbieter LCDPs innerhalb ihrer Cloud-basierten Lösungen an. Microsoft war einer der ersten, die den Trend aufgriffen und im November 2016 Power Apps LCDP veröffentlichte. Im Januar 2020 übernahm Google den LCDP-Anbieter AppSheet und machte ihn zu seinem Flaggschiff unter den Low-Code-Lösungen. Im Juni 2020 veröffentlichte Amazon Honeycode, eine LCDP für die Entwicklung von Web- und Mobilanwendungen [DiRuscio.2022].

2.2 Wichtige Begriffe

Um sich in der Thematik der Low Code Entwicklung mühelos informieren zu können, ist es wichtig im Zusammenhang mit der Low Code Entwicklung genutzte Begriffe zu verstehen. Das folgende Unterkapitel bietet deshalb eine kurze Übersicht über diese Begriffe.

2.2.1 Abgrenzung zu High-Code

High-Code oder traditionelle Entwicklung beinhaltet manuelle Programmierung in mehreren Sprachen und mit verschiedenen Technologien. Es können beliebige Lösung mit individuellen Anforderungen erstellt werden, dafür werden jedoch in hohem Maße professionelle Entwicklerressourcen benötigt [Mendix.2023].

High-Code-Development ist zeitintensiv und kostspielig. Es werden spezialisierte Fähigkeiten mit einem kleinen Talentpool benötigt. Außerdem ist eine separate Entwicklung für Mobilgeräte, Web und verschiedene Betriebssysteme/Geräte erforderlich, was sich durch die meisten LCDPs vermeiden lässt [Mendix.2023].

2.2.2 Abgrenzung zu No-Code

Der Begriff "No-Code-Entwicklungsplattform", oder auch "NCDP" wird für Plattformen verwendet, die durch visuelle Sprachen, grafische Benutzeroberflächen und Konfiguration die Notwendigkeit der Programmierung eliminieren. Während der Begriff im Marketing weit verbreitet ist, lehnen es Marktanalysefirmen derzeit ab, ihn zur Identifizierung eines klaren Marktsegments zu verwenden [DiRuscio.2022].

Im Gegensatz zu dem Entwickeln mit LCDPs ist es bei NCDPs jedem ohne jegliche Programmierkenntnisse möglich zu entwickeln. Das reduziert das benötigte Know-How für ein Projekt und somit auch die Kosten [Microsoft.2023].

Auf der anderen Seite können starre Templates bedeuten, dass die Anpassung der Anwendung und die Flexibilität eingeschränkt werden. Und, App-Entwickler ohne professionelle Entwicklungserfahrung könnten wichtige Aspekte der Benutzerfreundlichkeit übersehen [Microsoft.2023].

Low-Code- und No-Code-Plattformen können für unterschiedliche Zwecke nützlich sein. Eine Low-Code-Plattform wird meist von IT-Fachleuten mit einigen Programmierkenntnissen verwendet, um benutzerdefinierte Anwendungen zu erstellen, während No-Code-Plattformen in der Regel Geschäftsanwendern ohne Programmierkenntnisse ermöglichen, ihre eigenen Entwicklungsanforderungen zu erfüllen [Microsoft.2023].

2.3 Use Cases

Low-Code-Plattformen können zwar für die Entwicklung einfacher Automatisierungen und Integrationen verwendet werden, aber auch für die Entwicklung komplexer Anwendungen auf Unternehmensebene kann auf LCDPs zurück gegriffen werden [KevinShuler.2023].

Aber nicht alle Geschäftsprobleme sind für LC geeignet. Bevor sich für den Low Code Ansatz entschieden wird, sollte geprüft werden, ob es für den eigenen Anwendungsfall eine geeignete LCDP gibt. In diesem Kapitel werden mögliche Use Cases für Low Code Entwicklungsplattformen erläutert. Dadurch wird dem Leser nicht nur eine bessere Vorstellung dieses Themas geschaffen sondern auch die Vielzahl der Anwendungsgebiete verdeutlicht.

2.4 Anwendungsfall: ERP und CRM

Das wohl bekannteste Beispiel für die Entwicklung mit LCDPs: es kann alles von einem benutzerdefinierten ERP bis zu einem benutzerdefinierten CRM erstellt werden. Und Unternehmen arbeiten oft mit Supportpartnern oder Anbietern zusammen, um Beschaffung, Personalwesen und andere komplexe Arbeitsabläufe zu automatisieren [KevinShuler.2023].

2.4.1 Anwendungsfall: MVPs

Ob eine Anwendung erfolgreich sein wird, ist erst möglich zu wissen, wenn sie tatsächlich gestartet wird. Das bedeutet aber nicht dass dafür immer in einer umfassende Entwicklung investiert werden muss [OleksiiGlib.2022].

Ein MVP (Minimum Viable Product) ist eine erste Version eines Produkts mit seinen Kernfunktionen, die in der Regel dazu dient, eine Hypothese zu überprüfen. So kann festgestellt werden, ob es einen Markt für das Produkt gibt und was verbessert werden sollte, um mehr Nutzer zu gewinnen [OleksiiGlib.2022].

Low Code Development Plattformen ermöglichen die rasche Entwicklung eines MVP. Mit Low-Code-MVP-Entwicklungsansätzen können also Konzepte validiert werden, ohne in qualifizierte und teure Entwicklungsteams und einen fortschrittlichen technischen Stack zu investieren [OleksiiGlib.2022].

2.4.2 Anwendungsfall: Modernisierungen und Mobile-First

Altsysteme verursachen in der Regel mehr Ärger (und Kosten) als sie wert sind. Mehr als 75% des IT-Budgets der Bundesregierung in Höhe von 80 Milliarden Dollar werden für die Wartung von Altsystemen aufgewendet, und viele davon sind über 25 Jahre alt. Legacy-Systeme behindern Unternehmen, indem sie einen Großteil ihres IT-Budgets

verschlingen und sie an veraltete Plattformen binden, die zu wenig leisten und die Reaktionsfähigkeit des Unternehmens einschränken [KevinShuler.2023].

And Ricoh replaced their legacy systems, resulting in a 253% ROI in 7 month [KevinShuler.2023].

2.4.3 Anwendungsfall: Smart Process Apps

Smart Process Apps: Operational efficiency apps sind Anwendungen mit dem Ziel durch die Automatisierung manueller Prozesse Kosten zu senken. Die North Carolina State Uni bspw. nutzt eine LCDP, um eine App zur Kursanmeldung zu erstellen, die 500.000 Anmeldungen für Nicht-Kreditkurse pro Jahr ermöglicht.

2.4.4 Anwendungsfall: neue Plattformen

neue Plattformen: wie Augmented Reality, Virtual Reality oder dialogorientierte Schnittstellen. Einige Multiexperience DPs (MXDPs) ermöglichen die Entw. für diese über NC/LC

2.5 Übersicht von Low Code Development Plattformen in der Praxis

LCDPs unterstützen die Entwicklung von Zielumgebungen, die Web-only oder auch nativ sein können. So können sie sowohl Desktop- als auch Mobilgeräte nativ unterstützen und sich in bestehende Arbeitsabläufe integrieren, die mit beliebigen Software-as-a-Service (SaaS)-Anwendungen entwickelt wurden, darunter Zapier, Amazon AppFlow und Trello. Appian ist eine der langlebigsten LCDP, während Amazon Honeycode und Google AppSheet zu neueren Ansätzen gehören. [DiRuscio.2022]

Einige der Merkmale, die bestehende LCDPs unterscheiden beruhen auf der User Experience der hochentwickelten grafischen Benutzeroberflächen, die Werkzeuge und Widgets bereitstellen, mit denen Entwickler die gewünschten Anwendungen konzipieren können. Drag-and-Drop-Möglichkeiten, fortgeschrittene Berichtsfunktionen, Entscheidungsmaschinen zur Modellierung komplexer Logik und Formular-building-Tools sind nur Beispiele für Funktionen im Frontend von LCDPs. [DiRuscio.2022]

Außerdem können manche LCDPs die Entwicklung durch Live-Kollaborations-Tools unterstützen, um geografisch verteilten Entwicklern gemeinsames, kollaboratives Arbeiten an denselben Anwendungen zu ermöglichen. [DiRuscio.2022]

Ein weiteres Unterscheidungsmerkmal aktueller LCDPs bezieht sich auf die unterstützte Anwendungsdomäne. Diese sollte im Mittelpunkt des Interesses stehen. Node-RED beispielsweise unterstützt in erster Linie die Entwicklung von IoT-Anwendungen. Andere Plattformen unterstützen die Entwicklung von Chatbots, während die Mehrzahl der bestehenden LCDPs darauf abzielen, universell einsetzbar zu sein und die Entwicklung beliebiger datenintensiver Anwendungen zu unterstützen. [DiRuscio.2022]

LCDPs können Benutzern auch vordefinierte Artefakte zur Verfügung stellen, die als Ausgangspunkte verwendet werden können. Salesforce App Cloud enthält beispielsweise den umfangreichen AppExchange-Marktplatz, der aus vorgefertigten Anwendungen und Komponenten, wiederverwendbaren Objekten und Elementen, Drag-and-Drop-Process-Builder und integriertem Kanban-Board besteht. [DiRuscio.2022]

Bei der Betrachtung der typischen Schritte einer Anwendungsentwicklung mit LCDPs kann man in jedem Schritt außerdem unterschiedliche Ausführungen in den verschiedenen Plattformen sehen.

Dies sind die 12 besten Low-Code-Plattformen, die es derzeit gibt:

1. Zoho Creator
2. Appian
3. PowerApps
4. Mendix
5. Outsystems
6. AppMaker
7. Quickbase
8. Verfolgen über
9. Salesforce App Cloud
10. Kissflow
11. Workato
12. Pipefy

Es gibt jedoch mehr als 220 Low-Code-Plattformen auf dem Markt und mehrere, die maschinelles Lernen nutzen [KevinShuler.2023].

3 Nachteile von LCDP

Viele Unternehmen zögern noch den Low Code Ansatz in größere Projekte zu integrieren. In diesem Kapitel werden unterschiedliche Nachteile von Low Code Development Platforms genauer beleuchtet um die Gründe für dafür aufzuzeigen.

Der Outsystems State of Applications Development Report aus dem Jahr 2019 nennt mehrere Gründe, warum Unternehmen Low-Code derzeit nicht nutzen oder planen, es zu nutzen. Das Mangelndes Wissen über Low-Code macht mit 43% nahezu der Hälfte aller Unternehmen sorgen. Besorgnis über die Bindung an eine Plattform oder einen Anbieter macht mit 37% ebenfalls vielen Unternehmen Sorgen. Der Glauben, dass Low-Code die eigenen Anforderungen nicht erfüllen kann trifft bei 32% zu. Bedenken hinsichtlich der Skalierbarkeit von Low-Code-Anwendungen ist bei 28%

der Unternehmen ein Problem und Bedenken bezüglich der Sicherheit von Low-Code-Anwendungen bei 25% [KevinShuler.2023].

3.1 Steile Lernkurve

Die meisten Plattformen haben weniger intuitive grafische Oberflächen. Bei einigen von ihnen sind die Drag-and-Drop-Funktionen begrenzt, und sie bieten nicht genügend Lehrmaterial, einschließlich Beispielanwendungen und Online-Tutorials, um die Plattform zu erlernen. Für die Nutzung einiger Plattformen sind nach wie vor Kenntnisse in der Softwareentwicklung erforderlich, was die Akzeptanz bei bürgerlichen Entwicklern, die eigentlich die Hauptzielgruppe dieser Plattformen und Produkte sein sollten, einschränkt [Alamin.2023].

3.2 Geringe Skalierbarkeit

Low-Code-Plattformen sollten vorzugsweise Cloud-basiert und in der Lage sein, intensive Berechnungen auszuführen und Big Data zu verwalten, die mit hoher Geschwindigkeit, Vielfalt und Volumen entstehen. Aufgrund des Mangels an offenen Standards ist es jedoch sehr schwierig, die Skalierbarkeit der Plattformen zu bewerten, zu erforschen und zur Skalierbarkeit dieser Plattformen beizutragen [Alamin.2023].

3.3 Geringe Interoperabilität und eine enge Bindung an einen Anbieter

Interoperabilität bezeichnet die Interaktion und den Austausch von Informationen und Artefakten zwischen verschiedenen Low-Code-Plattformen, z.B. zur gemeinsamen Nutzung von Architekturdesign, Implementierung oder entwickelter Dienste. Dies ist unerlässlich, um Probleme im Zusammenhang mit der Herstellerabhängigkeit zu verringern. Leider sind die meisten Low-Code-Plattformen proprietär und geschlossen. Es fehlt an Standards in diesem Bereich, was die Entwicklung und Zusammenarbeit zwischen verschiedenen Ingenieuren und Entwicklern erschwert. So können sie nicht voneinander lernen und die Wiederverwendung von bereits definierten Architekturen, Artefakten und Implementierungen wird weiterhin behindert [Alamin.2023].

3.4 Geringere Flexibilität und Anpassbarkeit

Die Möglichkeit, neue Funktionen hinzuzufügen, die von der genutzten Plattform nicht angeboten werden, ist bei LCDPs oft schwer oder sogar unmöglich. Aufgrund fehlender Standards erfordern einige von ihnen umfangreiche Codierung, um neue Funktionen hinzuzufügen, die die architektonischen und gestalterischen Vorgaben der zu erweitern-den Plattform einhalten [Alamin.2023].

3.5 Geringe Teststandards

Es gibt keinen allgemeinen Rahmen, der die Low-Code-Testfunktionen unterstützt, der von LCDP-Entwicklern für den Aufbau der Testkomponente ihrer geplanten LCDPs verwendet werden kann. Dies ist problematisch, denn das Fehlen eines solchen Rahmens hat dazu geführt, dass Abhängigkeiten bestehender LCDPs von technischen Testwerkzeugen Dritter Tools entstanden sind, die für bürgerliche Entwickler nicht nutzbar sind. Hinzu kommt, einige kommerzielle LCDPs zwar neue Low-Code-Testing-Frameworks vorschlagen, diese aber nicht alle Low-Code-Testing-Merkmale erfüllen, nicht für andere LCDPs wiederverwendbar sind, und bzw. oder ihre Ressourcen nicht öffentlich zugänglich sind [Khorram.2020].

3.6 Auswahlsschwierigkeiten

Das Verstehen und Bewerten der LCDP, die für das zu lösende Problem genutzt werden soll sind schwierige Aufgaben, denn Entscheidungsträger müssen zwischen Hunderten von heterogenen Plattformen wählen, die ohne spezielle Unterstützung schwer zu bewerten sind. Bis 2022 gab es mehr als 400 LCSD-Plattformen, die von fast allen großen Unternehmen wie Google (2020) und Salesforce (2022) angeboten werden. Eine falsche Auswahl von LCDP kann jedoch zu einer Verschwendung von Zeit und Ressourcen führen [Alamin.2023].

4 Vorteile und Potenziale von LCDP

Während wir nun mit den Nachteilen von Low Code Entwicklungsplattformen die Gründe für das Zögern einiger Unternehmen betrachtet haben, stellen wir dem nun im folgenden Kapitel ihre Vorteile und Potenziale gegenüber.

Obwohl die Verteilung des Nutzens eher uneinheitlich ist, gibt es eine deutliche Kluft zwischen den drei wichtigsten Vorteilen und dem Rest. Die drei wichtigsten Vorteile sind verbesserte Kapitalrendite (RoI) (19,9%), Abstraktion (16,36%) und GUI-basierte Entwicklung (12,7%) [Bucaioni.2022].

Hinter diesen gibt es eine Gruppe von vier Vorteilen: Automatisierung (8,1%), Interoperabilität (6,3%), Benutzerfreundlichkeit und Flexibilität (jeweils 5,4%) [Bucaioni.2022].

Schließlich haben wir eine weitere Gruppe von 6 Vorteilen: Datenschutz oder Sicherheit und Anpassbarkeit (je 4,5%), Wartbarkeit (3,6%), Wiederverwendbarkeit, Offenheit und Digitalisierung von Geschäftsprozessen (je 2,7%) [Bucaioni.2022].

4.1 Schnittstellenbereitstellung

LCDPs unterstützen das sichere Integrieren von Daten und Logik aus beliebigen Quellen, Systemen oder Diensten - einschließlich wichtiger Altsysteme. Anwendungen können mithilfe vorkonfigurierter APIs und Konnektoren erstellt werden und so das Erstellen und Verwalten von Systemen in größerem Umfang ermöglichen, das gemeinsame Nutzen von Daten über Projekte und Teams hinweg unterstützen und eine schnellere Erstellung wiederverwendbarer Komponenten und Microservices mit nahtlosem Zugriff auf Unternehmensdaten bieten [Mendix.2023].

4.2 Zusammenarbeit von Entwicklern und Fachkundigen

Software durchdringt mittlerweile alle Aspekte unseres Lebens. Dadurch ist die Nachfrage nach Softwareentwicklern größer als das Angebot an entsprechend qualifizierten Fachleuten, und die Lücke wird immer größer. Darüber hinaus fühlen sich hochqualifizierte Softwareentwickler von intellektuell anspruchsvollen (und finanziell lohnenden) Softwaresystemen angezogen und nicht von alltäglichen Anwendungen. Dadurch entsteht eine wachsende Lücke für Geschäftsanwendungen, die effektiver wären als gemeinsam genutzte Tabellenkalkulationen, aber zu teuer sind, um sie manuell zu implementieren und zu warten.

Fünfundsechzig Prozent befragter Unternehmen nennen mangelnde technische Fähigkeiten oder Kenntnisse als Herausforderung in der digitalen Transformation [EmmaVanPelt.2019].

Die durchschnittlichen Computerkenntnisse haben sich in den letzten 40 Jahren jedoch dramatisch verbessert. Die Grundlagen der Programmierung werden in vielen Ländern im Rahmen der Sekundarschule gelehrt, und die neuen Generationen von Fachleuten sind Digital Natives. Während die meisten Fachexperten eine umfangreiche Ausbildung benötigen, um einen Teil der Komplexität eines CASE-Tools zu beherrschen, das vor 40 Jahren auf den Markt kam, verfügt eine wachsende Zahl heutiger Fachexperten über umfangreiche Erfahrungen im Umgang mit Computern und nicht-trivialer Software und benötigt viel weniger Ausbildung, um eine LCDP zur Implementierung maßgeschneiderter Anwendungen zu verwenden[DiRuscio.2022].

Auch die Medien, über die die Nutzer lernen haben sich in letzter Zeit stark verändert. Vor ein paar Jahrzehnten waren die primären Lernmedien für Anwendungsentwicklung Bücher, die von Technologieexperten geschrieben wurden. Dies hat sich mit dem Wachstum des Internets und insbesondere durch Video-Sharing-Dienste wie YouTube, die es einfacher machen aktuelles Schulungsmaterial für unterschiedliche Zielgruppen bereitzustellen, dramatisch verändert. Dies ermöglicht es Citizen Devel-

opern (dt. Bürgerentwickler) ihr eigenes Schulungsmaterial (z. B. Walk-Throughs, Screencasts) zu entwickeln und weiterzugeben, anstatt als passive Konsumenten zu agieren [DiRuscio.2022].

Low-Code-Plattformen werden in der Regel an professionelle Entwickler vermarktet, erfordern aber keine Programmierkenntnisse. Mit vielen Low-Code-Plattformen ist es möglich, eine Anwendung zu erstellen oder einen Geschäftsprozess zu automatisieren und Daten zu integrieren, ohne dass ein einziges Mal programmiert werden muss [MichelleGardner.2022]. Laut einer Studie von Gartner aus dem Jahr 2021 werden bis 2025 mehr als die Hälfte der Low-Code-Nutzer keine Informatiker sein. Die Einfachheit der LCDPs ermöglicht es Unternehmen und Fachleuten aus der Industrie, die keine technischen Kenntnisse haben, an der Entwicklung mitzuwirken. Dadurch können diese außerdem ihr Fachwissen in das Produkt einfließen lassen [OleksiiGlib.2022].

In einer Umfrage von 2018 fand Mendix heraus, dass 25% ihrer Low-Code-Entwicklergemeinschaft keine Erfahrung mit Code haben. Stattdessen haben 40% der Nutzer einen Hintergrund in der Wirtschaft. 70% der Entwickler ohne Hintergrunderfahrung in Low-Code lernten innerhalb eines Monats, wie man Anwendungen erstellt. Und 28% lernten es in weniger als zwei Wochen [KevinShuler.2023].

4.3 Beschleunigter Entwicklungsprozess

Vierundsiebzig Prozent befragter Unternehmen nennen die Unfähigkeit so schnell das Softwareprodukt zu liefern wie es das Unternehmen braucht als Herausforderung in der digitalen Transformation [EmmaVanPelt.2019].

Low-Code kann die Anwendungsentwicklung um den Faktor 10 beschleunigen, d. h. Entwickler können neue Anwendungen innerhalb von Tagen oder Wochen statt Monaten konzipieren, erstellen, testen und bereitstellen [KevinShuler.2023].

Die IDC stellte fest, dass Low-Code den Softwareentwicklungszyklus bei neuen Anwendungen um 62% und bei der Hinzufügung neuer Funktionen um 72% verkürzt, einen zusätzlichen Jahresumsatz von 19,8 Millionen US-Dollar erzielt und die Produktivität um 123% gesteigert hat [KevinShuler.2023].

Low-Code- und No-Code-Plattformen können den Zeitaufwand für die Erstellung benutzerdefinierter Anwendungen im Vergleich zur herkömmlichen Anwendungsentwicklung, die auf einer Programmiersprache basiert, um 50 bis 90% reduzieren [KevinShuler.2023].

Moderne LCDPs können beispielsweise nicht nur Code generieren, sondern die erzeugten Softwaresysteme auch auf skalierbaren cloudbasierten Infrastrukturen einsetzen und sie den Nutzern weltweit über webbasierte Schnittstellen sofort zur Verfügung stellen. Dies reduziert die Zeit und den Aufwand für die Freigabe von An-

wendungen (und Updates) und erhöht die Attraktivität von LCDPs als Medium für eine schnelle Anwendungsentwicklung und -bereitstellung [DiRuscio.2022].

Eine weitere Verbesserung: da im Grunde jeder in einem Unternehmen eine Rolle bei der Low-Code-Entwicklung spielen kann, können auch Nicht-Programmierer den Prozess beschleunigen, anstatt darauf zu warten, dass diejenigen mit Programmierkenntnissen Zeit haben, sich der App zu widmen [Microsoft.2023].

Low-Code-Plattformen optimieren außerdem die App-Entwicklung und machen die Teams letztlich produktiver. Entwickler tauschen ein wenig Programmierflexibilität gegen mehr Effizienz ein, aber Low-Code-Plattformen geben ihnen Zeit, damit sie sich auf Projekte konzentrieren können, die eine umfangreichere Programmierung erfordern [Microsoft.2023].

4.4 Reduzierte Kosten

Durch die Verkürzung des Entwicklungszyklus aus zeitlicher Sicht werden auch die Kosten, unabhängig davon, ob die App vom Unternehmen selbst oder von externen Entwicklern entwickelt wird, reduziert [Sanchis.2020b].

Aber das ist nicht der einzige Faktor der zur Kosteneinsparung beiträgt. Wie in 2.4.1 bereits behandelt, können mit LCDPs auch MVPs erstellt werden, was zur Minimierung von instabilen oder inkonsistenten Anforderungen führt. Die Verwendung von Low-Code bedeutet, dass Entwickler schnell minimale lebensfähige Produkte erstellen können, um Ideen und Kundenanforderungen zu validieren, bevor sie Ressourcen für Features und Funktionalitäten verschwenden, die der Kunde möglicherweise gar nicht, oder nicht in der Form, möchte [Sanchis.2020b].

4.5 Hohe Sicherheitsstandards und geringe Ausfallzeiten

Einundsechzig Prozent befragter Unternehmen nennen Sicherheitsanforderungen als Herausforderung in der digitalen Transformation [EmmaVanPelt.2019].

Viele Unternehmen haben eine große Ansammlung von Anwendungen, zwischen denen sie Daten mit Tabellenkalkulationen austauschen, manchmal auch per Email. Dadurch sind diese Daten einem unnötigen Risiko ausgesetzt. Eine E-Mail oder ein Passwort genügen, um Ihre Kundendaten zu gefährden. Mit Low-Code wird die Cyber-sicherheit einer unternehmensweiten, cloudbasierten Plattform bereitgestellt. Und es gibt Teams, die diese Sicherheitsmaßnahmen regelmäßig aktualisieren, um sicherzustellen, dass alle ihre Kunden die Vorschriften einhalten [KevinShuler.2023].

Low-Code-Plattformen können außerdem oft auch hohe Unternehmensanforderungen im Bezug auf Sicherheit erfüllen. Unternehmen mit einer geringsten Toleranz

gegenüber Ausfallzeiten und Datenverlusten kombiniert mit Anforderungen wie kontinuierliche Audits und unabhängige Sicherheitszertifizierungen, betreiben ihre Top-Anwendungen am ehesten auf Low-Code. Ihre Befürwortung von Low-Code beweist, dass unternehmenstaugliche Low-Code-Lösungen bereits auf dem Markt verfügbar sind [EmmaVanPelt.2019].

Und auch Datenschutz durch eine gesteigerte Privatsphäre spielt eine Rolle. Da durch LCDPs ohne eine tiefgreifende Expertise in der Programmierung entwickelt werden kann, können Unternehmen für die Entwicklung mehr und mehr auf eigene Mitarbeiter zurückgreifen und müssen diese Aufgabe nicht an Dritte auslagern. Das erhöht die Vertraulichkeit im Umgang mit eigenen Daten [Sanchis.2020b].

4.6 Schwierigkeiten mit komplexer Geschäftslogik

Unternehmen werden sich Low-Code zuwenden, um komplexe Geschäftslogik zu erstellen. Während viele Firmen heute benutzerdefinierten Code verwenden, um Anwendungen für komplexe Geschäftslogik zu nutzen, sind sie bestrebt, auf dem Erfolg aufzubauen, den die Low-Code-Entwicklung in anderen Bereichen des Unternehmens gebracht hat. In Zukunft werden Unternehmen wahrscheinlich eher Low-Code als benutzerdefinierten Code einsetzen, um geschäftskritischen Anwendungen auszuführen [EmmaVanPelt.2019].

4.7 Gesteigerte Flexibilität

Ist eine Anwendung flexibel anpassbar, so kann die Anwendung immer den aktuellen Anforderungen entsprechend angepasst werden. Ist eine Anwendung flexibel anpassbar, so kann die Anwendung immer den aktuellen Anforderungen entsprechend angepasst werden.

Low-Code-Plattformen erweitern Flexibilität, Geschwindigkeit und Automatisierung. Vierundsechzig Prozent der Unternehmen, die Low-Code für Top-Anwendungen nutzen, nennen als Grund für diese Entscheidung weil es die flexibelste Option ist. Mehr als die Hälfte sagen, dass sie Low-Code verwenden, weil es die schnellste Speed-of-Delivery vorweisen kann. Und 49% sagen, dass Low-Code die besten Möglichkeiten zur Prozessautomatisierung bietet [EmmaVanPelt.2019].

Die Wartungsphase einer Software ist ebenfalls von entscheidender Bedeutung um eine permanente Übereinstimmung zwischen dem von der App angebotenen und den geschäftlichen Anforderungen zu gewährleisten. Da Low-Code-Entwicklungsplattformen im Wesentlichen daraus bestehen, wenig Code anzubieten, muss auch wenig Code

gewartet werden, was die Flexibilität steigert [Sanchis.2020b].

4.8 Cloud-basiertes Deployment

Moderne LCDPs können nicht nur Code generieren, sondern die erstellten Softwaresysteme auch auf skalierbaren cloudbasierten Infrastrukturen bereitstellen und sie den Nutzern weltweit über webbasierte Schnittstellen sofort zugänglich machen. Dies kann den Zeit- und Arbeitsaufwand für die Freigabe von Anwendungen (und Updates) für Nutzer drastisch verkürzen und die Attraktivität von LCDPs als Medium für die schnelle Anwendungsentwicklung und -bereitstellung erhöhen [DiRuscio.2022].

5 Marktforschungen und Prognosen

Um unsere Leitfrage näher zu kommen, werden im folgenden Kapitel auch aktuelle Marktforschungen und Prognosen näher betrachtet. Unsere Leitfrage setzt eine Vermutung über die Zukunft voraus. Die aktuelle Lage in der Industrie und Prognosen von Marktforschungsinstituten bilden die beste Grundlage um Aussagen über die Zukunft treffen zu können.

Der globale Markt für Low-Code wird bis 2027 auf rund 65 Milliarden US-Dollar geschätzt. Außerdem wird erwartet, dass er bis 2030 187 Milliarden Dollar erreichen wird. Das entspricht einer Wachstumsrate von 31,1% für den Zeitraum 2020-2030. Im Jahr 2019 waren es 10,3 Milliarden US-Dollar. COVID 19 triebte diese Entwicklung aufgrund immer mehr Remote-Arbeitsumgebungen, der Notwendigkeit für Unternehmen, agil zu sein, und dem Streben nach schlanken, effizienteren Geschäftssystemen voran [KevinShuler.2023]. Bis 2025 werden Unternehmen 70% ihrer neuen Anwendungen mit Low-Code- oder No-Code-Plattformen entwickeln (Im Jahr 2020 waren es noch weniger als 25%) [KevinShuler.2023].

Bis 2024 werden die meisten Unternehmen (80%) Richtlinien für Bürgerentwickler eingeführt haben. Dies wird der Versuch sein, die IT-Abteilungen zu entlasten und gleichzeitig die Arbeitsabläufe zu verbessern, da Unternehmen in zunehmend unberechenbaren Märkten arbeiten. Derzeit geben 72% der IT-Führungskräfte an, dass ihr Ticket-Backlog strategische Projekte behindert. Tatsächlich verbringen sie einen Großteil ihrer Zeit mit der Wartung von Altsystemen. 26% der Führungskräfte halten Low-Code-Plattformen für die wichtigste Investition in die Automatisierung (seit der Pandemie von 10% angestiegen) [KevinShuler.2023].

Gartner geht davon aus, dass es bis 2023 in großen Unternehmen viermal mehr Bürgerentwickler als professionelle Entwickler geben wird. LCDPs werden innerhalb von 2 bis 5 Jahren den Mainstream erreichen [KevinShuler.2023].

In einem Bericht von 451 Research und Filemaker, Inc. aus dem Jahr 2017 gaben 82% der Unternehmen an, dass die App-Entwicklung außerhalb der IT-Abteilung wichtig ist. In diesem Bericht wurde festgestellt, dass fast 60% der Individualsoftware außerhalb der IT-Abteilung entwickelt wurden. Laut Gartner könnte diese Zahl dank Low-Code bis 2024 auf 70% steigen.

Forrester schätzt, dass bis Ende 2021 75% der gesamten Unternehmenssoftware mit Low-Code entwickelt werden wird. Bis 2023 werden Unternehmen mehr als 500 Millionen Apps in der Cloud entwickeln (das entspricht der Anzahl der in den letzten 40 Jahren entwickelten Apps) [KevinShuler.2023].

Bis 2024 werden 75% der Unternehmen mindestens vier Low-Code-Plattformen für die Entwicklung von Geschäftsanwendungen nutzen. Angetrieben durch die Pandemie werden über 50% der mittleren und großen Unternehmen eine Low-Code- oder No-Code-Plattform für die Anwendungsentwicklung nutzen [KevinShuler.2023].

Laut der Gartner-Umfrage haben 4% der Befragten derzeit Bürgerentwicklungsinitiativen eingerichtet. Und 20% der Unternehmen, die noch keine solchen Initiativen haben, sind dabei, solche zu entwickeln oder zu evaluieren. Mit zunehmender Verbreitung dieser Systeme und dem Bekanntwerden ihrer Vorteile bei Entwicklern und IT-Führungskräften wird der Druck auf die Unternehmen steigen, diese Systeme einzuführen [KevinShuler.2023].

47% der Entwickler gaben an, dass sie keinen Zugang zu den Tools haben, die sie benötigen, um Anwendungen schnell genug zu erstellen, um Termine einzuhalten. Und 41% der Entwickler wünschen sich, dass mehr als die Hälfte der Anwendungen in ihrem Unternehmen auf Low-Code-Plattformen erstellt werden [KevinShuler.2023].

Während 4 von 5 Unternehmen in den USA derzeit Low-Code verwenden, sind es immer noch 20%, die dies nicht tun. Weltweit liegt diese Zahl bei 23% [KevinShuler.2023].

Laut Appian haben 31% der Unternehmen, die derzeit Low-Code verwenden, keine ihrer hochwertigsten Anwendungen damit entwickelt oder bereitgestellt [KevinShuler.2023].

Microsoft informiert mit einer Umfrage zum Low Code Productivity Impacts Trend Index, welche von einem unabhängigen Forschungsunternehmen, Edelman Data x Intelligence, unter 322 Nutzern von Low Code/No Code-Plattformen oder -Apps, 300 potenziellen Nutzern von Low Code/No Code-Plattformen oder -Apps, 150 Entscheidungsträgern in Unternehmen, die Low Code/No Code-Fähigkeiten einstellen möchten, und 150 Entscheidungsträgern in Unternehmen, die bereits Low Code/No Code-Plattformen oder -Apps in ihrem Unternehmen nutzen, durchgeführt wurde [Microsoft.2022]. Die Studie beantwortet die Frage warum, wenn Low-Code- und No-Code-Plattformen ein so großes Potenzial für die Unternehmenskultur haben, sie dann nicht schon heute häufiger eingesetzt werden. Eine überwältigende Mehrheit der Entscheidungsträger in Unternehmen und Nutzer (71 Prozent bzw. 76 Prozent) verweist auf ein mangelndes

Bewusstsein für potenzielle Anwendungsfälle. Auch Bedenken hinsichtlich der Cybersicherheit sowie der Kosten und des Schulungsaufwands für die Mitarbeiter, welche den Nutzen dieser Plattformen oder Anwendungen maximieren sollen, stehen an erster Stelle. Wie bei jeder Unternehmensinvestition wollen die Führungskräfte sicher sein, dass sie eine sinnvolle Rendite erzielen werden [Microsoft.2022].

Einmal eingeführt, sprechen die Ergebnisse schnell für sich selbst und setzen einen Innovationszyklus in Gang - doch die Sicherung der Anfangsinvestitionen erweist sich als Hindernis. Entscheidungsträger dessen wissen am ehesten, wie der Entscheidungsprozess in ihrem Unternehmen abläuft; sie spielen eine Schlüsselrolle, wenn es darum geht, mit den Anwendern in Kontakt zu treten, um die notwendigen Beweise für die Investition des Unternehmens zu erbringen. Angesichts der eindeutigen kulturellen Vorteile und der Erfolgsgeschichten der Anwender haben BDMS ein überzeugendes Argument für die Einführung, das sie der Führungsebene vorlegen können [Microsoft.2022].

6 Fazit und Zukunftsausblick

Um diese Hausarbeit abzurunden, fasst dieser Abschnitt nun die zuvor gewonnen Informationen und Erkenntnisse zusammen. Anhand dessen wird dann eine Beantwortung der Leitfrag "Werden sich Low-Code Development Platforms durchsetzen?" geboten.

Innerhalb der vorherigen Kapitel wurden detailreich einige Nach- und einige Vorteile von Low Code Development Plattformen erläutert. Während die Nachteile

Anhand der nun genannten Informationen lässt sich annehmen dass LCDPs sich in der Zukunft durchsetzen werden.