

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	Abgrenzung und Vergleich zu No Code und High Code	6
2.3	Use Cases	7
2.3.1	Anwendungsfall: MVPs	7
2.3.2	Anwendungsfall: Modernisierungen	7
2.3.3	Anwendungsfall: Smart Process Apps	7
2.3.4	Anwendungsfall: neue Plattformen	8
2.4	Übersicht von Low Code Development Plattformen in der Praxis	8
3	Nachteile von LCDP	9
3.1	steile Lernkurve	9
3.2	geringe Skalierbarkeit	9
3.3	geringe Interoperabilität	9
3.4	geringere Flexibilität und Anpassbarkeit	9
3.5	geringe Teststandards	10
3.6	Sicherheitsrisiken	10
4	Vorteile und Potenziale von LCDP	10
4.1	Schnittstellenbereitstellung	10
4.2	Zusammenarbeit von Entwicklern und Fachkundigen	10

4.3	Beschleunigter Entwicklungsprozess	11
4.4	Reduzierte Kosten	12
4.5	hohe Sicherheitsstandards und geringe Ausfallzeiten	12
4.6	Einhaltung von Konventionen und Best Practices	13
4.7	Schwierigkeiten mit komplexer Geschäftslogik	13
4.8	gesteigerte Flexibilität	13
4.9	Cloud-based deployment	13
5	Fazit und Zukunftsausblick	13

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 war oftmals die Lösung auf diese. Corona-Testzentren brauchten 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

In meiner Hausarbeit befasse ich mich 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 befasst sich diese Hausarbeit anfänglich mit einer Einführung in die Thematik, um dem Leser einen leichten Einstieg zu ermöglichen. Im Laufe der Arbeit soll sich für den Leser herauskristallisieren, ob das Entwickeln mit Low Code Development Platforms 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-Platforms abgrenzen?
- (iii) Welche Use Cases für Low Code Development Platforms gibt es?
- (iv) Was für Low Code Development Plattformen gibt es in der Praxis?
- (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) XX

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 Begriffen 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 für 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 und das Lebenszyklusmanagement sowie das Plattformmanagement [DiRuscio.2022].

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 war der 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 Abgrenzung und Vergleich zu No Code und High Code

High-Code [Alamin.2023]

No-Code-Entwicklungsplattform (NCDP) ist ein verwandter Begriff, der für Plattformen verwendet wird, 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

Nicht alle Geschäftsprobleme sind für LC geeignet. Einfache, geschäftsorientierte Projekte sind jedoch ein guter Ausgangspunkt. Bevor sich jedoch 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.

2.3.1 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].

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].

2.3.2 Anwendungsfall: Modernisierungen

Modernisierungen: Legacy-Migrationsanwendungen zielen darauf ab, Anwendungen zu ändern, die keine neuen Prozesse unterstützen oder die richtige Benutzererfahrung bieten können.

2.3.3 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.

2.3.4 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

2.4 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.

Some of the most popular low-code platforms are Appian (2022), Google App Maker (2020), Microsoft Powerapps (2022), and Salesforce Lightning (2022) [Alamin.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.

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

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 fehlen-

der Standards erfordern einige von ihnen umfangreiche Codierung, um neue Funktionen hinzuzufügen, die die architektonischen und gestalterischen Vorgaben der zu erweiternden 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 Sicherheitsrisiken

[Guerra.2020]

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.

4.1 Schnittstellenbereitstellung

Lorem ipsum

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 Developern (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]

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 beschleunigt die Entwicklung und erfüllt den Bedarf der Unternehmen an Geschwindigkeit. Vierundachtzig Prozent der Unternehmen haben eine Low-Code Entwicklungsplattform oder -werkzeug eingeführt. Diese Firmen sind erfolgreich in ihren Bemühungen bestehende IT-Fähigkeiten zu verbessern, Produkte und Dienstleistungen zu erneuern und agiler zu werden - all das ermöglicht durch eine schnellere Markteinführung durch LCDPs [EmmaVanPelt.2019].

Geschäftsanwendungen werden schnell generiert und bereitgestellt mit geringerem Aufwand für Codierung, Implementierung, Installation und Konfiguration [Kaiser.2021].

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 Anwendungen (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

Wie in 4.3 beschrieben, wird durch LCDPs ein beschleunigter Entwicklungsprozess erreicht. Aber das ist nicht der einzige Faktor der zur Kosteneinsparung beiträgt.

4.5 hohe Sicherheitsstandards und geringe Ausfallzeiten

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

Low-Code-Plattformen können höchste Unternehmensanforderungen erfüllen. Unternehmen mit einer geringsten Toleranz gegenüber Ausfallzeiten und Datenverlusten und dazu Anforderungen an 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]

4.6 Einhaltung von Konventionen und Best Practices

4.7 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]

4.8 gesteigerte Flexibilität

Low-Code-Plattformen mit Top-Anwendungen 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-DevDelivery vorweisen kann. Und 49 % sagen, dass Low-Code die besten Möglichkeiten zur Prozessautomatisierung bietet. [EmmaVanPelt.2019]

4.9 Cloud-based 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 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.

Rückblickend auf die vorhergehenden Kapitel kann man XXX feststellen.

Die Verwendung von Low-Code in Unternehmen wird wahrscheinlich zunehmen, da die Plattformen Unterstützung für komplexe Geschäftslogik verbessern. Heute würden 30 % der Unternehmen custom coding nutzen um komplexe Logik zu erstellen,

doch dieser Anteil schrumpft auf 16 %, wenn sie in die Zukunft denken. Stattdessen würden sie es vorziehen, Low-Code für komplexe Geschäftslogik verwenden (30 %). Dies zeigt uns, dass Unternehmen zwar Vorbehalte gegenüber Low-Code für komplexe Logik haben, aber auch ein Auge auf die zukünftigen Möglichkeiten von Low-Code und die Möglichkeit in diese Richtung zu gehen, haben. [EmmaVanPelt.2019]