

Hier sollte ich das Abstract rein schreiben

Werden sich Low-Code Development Platforms durchsetzen?

Sonja Klein

10.01.2023

Inhalt

1	Einleitung	3
1.1	Relevanz und Ziel der Arbeit	3
1.2	Methodik	5
1.3	Aufbau der Arbeit	5
2	Entstehung von LCDP	5
2.1	Entstehung	5
2.2	Definition und Abgrenzung zu Low Code Development Platforms und High Code Development Platforms	5
2.3	Beispiele	6
2.4	Use Cases	6
3	Nachteile von LCDP	6
3.1	-	6
4	Potenziale und Vorteile von LCDP	6
4.1	-	6
5	Fazit und Ausblick	6
6	Referenzen	7

1 Einleitung

Low Code Development Plattformen sind benutzerfreundliche Umgebungen, die vor allem bei der Entwicklung mobiler Anwendungen zunehmend an Bedeutung gewinnen. In jüngster Zeit werden sie zunehmend von großen IT-Unternehmen eingeführt und gefördert, um die Entwicklung von Softwareanwendungen zu beschleunigen und den typischen Zeitdruck in verschiedenen Bereichen zu bewältigen. Bis zu einem gewissen Grad könnte Low-Code-Engineering als ein Synonym und eine Weiterentwicklung des traditionellen Model-Driven-Engineering (MDE) betrachtet werden. Bei Low-Code-Entwicklungsplattformen erstellen die Benutzer "Modelle", um die Schlüssellogik zu konzipieren, und der Code wird automatisch generiert. [3]

1.1 Relevanz und Ziel der Arbeit

Die Informationstechnologie (IT) ist heute ein wesentlicher Treiber der digitalen Transformation und damit ein wichtiger Faktor für den Erfolg einer Organisation. Infolgedessen steigt der Bedarf an neuen, innovativen und umfassenden Softwarelösungen und automatisierten Arbeitsabläufen, die Unternehmen pünktlich, budgetgerecht und in hoher Qualität liefern müssen (Richardson und Rymer 2014). Trotzdem gab es im Jahr 2019 124.000 offene IT-Stellen in der deutschen Wirtschaft, so Bitkom Research (2019), was einen Anstieg von 51 % im Vergleich zum Vorjahr bedeutet. Fast ein Drittel davon waren Softwareentwickler (Bitkom Research 2019). Internationale Studien zum Mangel an IT-Fachkräften zeigen ähnliche Ergebnisse (Cushing 2019; Harvey Nash und KPMG 2018). Um diesem Mangel entgegenzuwirken und den steigenden Bedarf an IT-Lösungen zu befriedigen, sollte eine zeitgemäße Software- und Prozessentwicklung einfach und effizient sein und auch weniger qualifizierte Mitarbeiter (in Bezug auf ihre Programmierkenntnisse) an IT-Entwicklungsaufgaben beteiligen (Richardson und Rymer 2014).

Diese befähigten Mitarbeiter werden als Bürgerentwickler bezeichnet und sind hauptsächlich entweder Power-User, Entwickler in einer Fachabteilung oder reguläre Mitarbeiter in der Fachabteilung (McKendrick 2017). IT-Entwicklungsplattformen im Unternehmen helfen ihnen, Geschäftsanwendungen oder Workflows unabhängig von der IT-Abteilung des Unternehmens zu entwickeln (Rollings 2012). Um einen Hypernamen für diese Plattformen zu etablieren, prägte Forrester Research (Richardson und Rymer 2014) erstmals den Begriff "Low-Code Development Platform" (LCDP) im Jahr 2014. Die Autoren charakterisieren LCDPs als eine enorme Reduzierung von Handcodierung, als schnellere Bereitstellung von Anwendungen mit Hilfe von visuellen Tools und als die Fähigkeit, Daten effektiv aufzubereiten, um mehrstufige Workflows zu erstellen. Eine weitere Veröffentlichung (Tisi et al. 2019) definiert sie als Software-Entwicklungsplattform in der Cloud, die ein Platform-as-a-Service (PaaS)-Modell bietet, mit dem Nutzer schlüsselfertige betriebliche Anwendungen mit deklarativen Sprachen, dynamischen grafischen Benutzeroberflächen (UI) und visuellen Diagrammen.

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 Challenges 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.

LC/NC gewinnt 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. [1]

Mit dem Aufkommen der digitalen Transformation und der Verlagerung der Arbeit in die Ferne mussten viele Unternehmen schnelle Veränderungen vornehmen, um den neuen Anforderungen ihrer Mitarbeiter gerecht zu werden. Und da immer mehr Menschen von zu Hause aus arbeiten, müssen Unternehmen weiterhin lernen, wie sie mobile Benutzer besser unterstützen und schnell neue Anwendungen entwickeln können, damit ihre Mitarbeiter produktiv, kooperativ und effizient bleiben.

Eine wichtige Möglichkeit, wie Ihr Unternehmen seine Mitarbeiter unterstützen kann, ist die Low-Code-Entwicklung. Anstatt sich stark auf die Programmierung zu verlassen, vereinfachen Low-Code-Plattformen die Anwendungsentwicklung mit Techniken wie Drag-and-Drop-Funktionalität und visueller Anleitung. Das bedeutet, dass jeder in Ihrem Unternehmen, unabhängig von seinen technischen Kenntnissen oder Fähigkeiten, Anwendungen erstellen kann, so dass das Unternehmen einige Aufgaben von der IT-Abteilung übernehmen kann.

Im Gegensatz zu professionellen Entwicklern kennen sich solche "Citizen Developer" vielleicht nicht so gut mit der manuellen Programmierung aus und haben in der Regel auch keine formale Ausbildung in der Programmierung, aber sie können dennoch Anwendungen mit Low-Code-Plattformen erstellen. Da Low-Code den Prozess der Anwendungserstellung vereinfacht, müssen Bürgerentwickler keine Programmierexperten sein, um effiziente Anwendungen zu erstellen. Durch den Einsatz von Bürgerentwicklern werden auch Ihre IT- und Entwicklungsressourcen entlastet, sodass sie sich auf komplexere Projekte konzentrieren können.

Low-Code-Plattformen helfen Ihrem Unternehmen und Ihren Entwicklern auch, den wachsenden Bedarf an erstklassigen internen Workflow-Anwendungen, zeitsparenden Automatisierungen, besseren Kundenerlebnissen und nahtlosen Integrationen zu decken. Außerdem können Ihre professionellen Entwickler damit Anwendungen schneller erstellen und müssen nicht Zeile für Zeile Code schreiben.

1.2 Methodik

Zu Beginn wurde ein erster Eindruck über die Quellenlage geschaffen. In den online-Bibliotheken der Hochschule für Technik, Wirtschaft und Gestaltung Konstanz und in der Universität Konstanz zum Thema Low Code nicht viel zu finden war, musste auf andere Quellen ausgewichen werden.

Wiso-net lieferte ebenfalls keine relevanten Ergebnisse. Google Scholar und AISnet.org brachten mehr Informationen.

1.3 Aufbau der Arbeit

In diesem Artikel wird zu Beginn der Begriff Low Code Development Plattformen definiert. Als nächstes wird er von den Begriffen No-Code Development Plattformen und High Code Development Plattformen abgegrenzt.

2 Entstehung von LCDP

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales...

Die Mainstream-Hochsprachen der Programmierung (High-Level im Vergleich zu Assembler Sprachen und Maschinencode) haben sich seit dem Aufkommen der Sprache Fortran der Sprache Fortran vor gut einem halben Jahrhundert dramatisch weiterentwickelt, wobei Hunderte von Sprachen wurden seitdem entwickelt wurden.[Margaria.2021]

2.1 Entstehung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante... ???

2.2 Definition und Abgrenzung zu Low Code Development Platforms und High Code Development Platforms

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales...

2.3 Beispiele

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante.

2.4 Use Cases

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante.

3 Nachteile von LCDP

However, Vincent et al. (2019) consider these NCDPs primarily as a marketing and positioning statement and part of the LCDP market. Despite the described benefits of LCDPs, studies show that these platforms create new technical and social challenges (McKendrick 2017; OutSystems 2019). To find out whether current research covers these challenges, we review and classify the current state of research regarding LCDPs and propose possible future research fields by identifying research gaps. <— das war auch alles von Niculin

3.1 -

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante.

4 Potenziale und Vorteile von LCDP

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante

4.1 -

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante.

5 Fazit und Ausblick

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilissem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante... ???

6 Referenzen

- [1] Amy Glasscock. *The need for speed. Why State CIOs are Turning to Low-Code and No-Code Software Development*. Ed. by NASCIO. 2021. URL: <https://uidl.naswa.org/handle/20.500.11941/4392>.
- [2] Alexander C. Bock and Ulrich Frank. “Low-Code Platform”. In: *Business & Information Systems Engineering* 63.6 (2021). PII: 726, pp. 733–740. ISSN: 2363-7005. DOI: 10.1007/s12599-021-00726-8.
- [3] Yi Wang et al. “The Necessity of Low-code Engineering for Industrial Software Development: A Case Study and Reflections”. In: *2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. 2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (Wuhan, China). IEEE, 2021, pp. 415–420. ISBN: 978-1-6654-2603-9. DOI: 10.1109/ISSREW53611.2021.00112.
- [4] Yi Wang et al. “The Necessity of Low-code Engineering for Industrial Software Development: A Case Study and Reflections”. In: *2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2021, pp. 415–420. ISBN: 978-1-6654-2603-9. DOI: 10.1109/ISSREW53611.2021.00112.