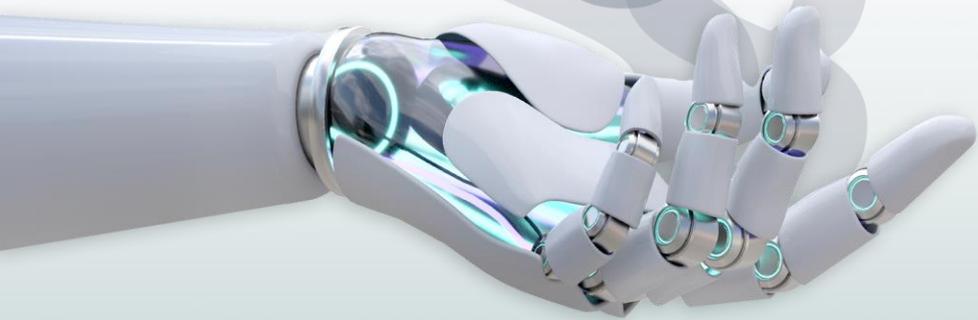


READY

FOR THE FUTURE



IDSt-TaxTechToks

Kompakt, clever, innovativ



Low Code / No Code für steuerliche Anwendungen

Wird der Steuerexperte zum Softwareentwickler?

Uli Weber

Capgemini, 24. Oktober 2024

Wir stellen uns vor

Moderator

Dietmar Gegusch
Deloitte



- Steuerpolitik
- Learning und Knowledge Management
- Verantwortlich für IDSt-Akademie
- Stellv. Vorsitzender FA VIII

Referent

Uli Weber
Capgemini



Relevante Erfahrungen:

- Anforderungsmanagement und Business Analyse
- Nutzung von Low Code Anwendungen in der Softwareentwicklung
- Modellbasierte Softwareentwicklung
- Steuerliche Digitalisierungsprojekte

Low Code / No Code für steuerliche Anwendungen

IDSt Glossar – Was ist Low Code / No Code?

Low Code / No Code

Unter Low Code- oder No Code-Lösungen werden **Software-Entwicklungswerkzeuge** bzw. Plattformen verstanden, welche es dem Nutzer ermöglichen, ohne Code, also ohne die Verwendung von Programmiersprachen, Lösungen zu erstellen. Die „Programmierung“ erfolgt über eine graphische Benutzeroberfläche, die es per „**Drag-and-Drop**“ ermöglicht Funktionsbausteine zusammen zu setzen. Die eigentliche Codierung läuft im Hintergrund. Im Unterschied zu No Code-Lösungen erfordern Low Code-Lösungen noch ein gewisses Maß an klassischem Programmieraufwand. Vornehmlich werden Low Code-Umgebungen auch dazu genutzt, fehlende Funktionen nachträglich hinzuzufügen. No Code-Lösungen bieten auch Mitarbeitenden ohne Programmierwissen die Möglichkeit, individuelle Lösungen zu erstellen, was sich gerade für die **Steuerabteilung** als durchaus reizvoll darstellen kann.

Übergeordnete Einsatzgebiete	Konkrete steuerliche Anwendungsfälle (beispielhaft)
<ul style="list-style-type: none"> ▪ Software-Entwicklung ohne Programmierkenntnisse ▪ Codierung im Hintergrund 	<ul style="list-style-type: none"> ▪ Erstellung steuerlicher Expertensysteme in Form von Entscheidungsbäumen etwa im Bereich der Umsatzsteuer ▪ Erstellung von Workflows für die Steuerabteilung zur Einholung wiederkehrender Informationen von Tochtergesellschaften oder aus anderen Abteilungen ▪ Erstellung von Chatbots für wiederkehrende steuerliche Fragen („Steuerlicher Self Service“) ▪ Nutzung von vordefinierten „Bausteinen“ zur Digitalisierung von Arbeitsschritten im Steuerumfeld, für die eine Standardsoftware (noch) nicht zur Verfügung steht

Low Code / No Code für steuerliche Anwendungen

Antwort auf die Frage an die KI (MS Copilot) – „Was ist Low Code und was ist No Code?“

Low-Code und No-Code sind Ansätze zur Softwareentwicklung, die darauf abzielen, den Programmieraufwand zu reduzieren und die Entwicklung von Anwendungen zu beschleunigen.

Low-Code

Definition: Low-Code-Plattformen bieten eine visuelle Entwicklungsumgebung, in der Benutzer Anwendungen durch Drag-and-Drop-Komponenten und visuelle Modellierung erstellen können. Es ist jedoch möglich, bei Bedarf eigenen Code hinzuzufügen.

Zielgruppe: Diese Plattformen richten sich hauptsächlich an **professionelle Entwickler**, die durch die visuelle Unterstützung schneller arbeiten können, aber auch an technisch versierte Geschäftsanwender.

Flexibilität: Low-Code bietet mehr Flexibilität und Anpassungsmöglichkeiten, da Entwickler bei Bedarf Code einfügen können.

No-Code

Definition: No-Code-Plattformen ermöglichen es Benutzern, Anwendungen vollständig ohne Programmierung zu erstellen. Dies geschieht durch eine rein visuelle Schnittstelle mit vorgefertigten Bausteinen.

Zielgruppe: Diese Plattformen sind für **Nicht-Entwickler oder Citizen Developer** gedacht, die keine Programmierkenntnisse haben.

Einschränkungen: No-Code-Plattformen bieten weniger Flexibilität und Anpassungsmöglichkeiten im Vergleich zu Low-Code, da sie auf vorgefertigte Module und Bausteine beschränkt sind.

Gemeinsamkeiten und Unterschiede

Gemeinsamkeiten: Beide Ansätze zielen darauf ab, die Softwareentwicklung zu vereinfachen und zu beschleunigen, indem sie visuelle Werkzeuge und Drag-and-Drop-Funktionalitäten nutzen.

Unterschiede: Der **Hauptunterschied liegt in der Zielgruppe und der Flexibilität**. Low-Code erfordert einige Programmierkenntnisse und bietet mehr Anpassungsmöglichkeiten, während No-Code für Benutzer ohne Programmierkenntnisse gedacht ist und weniger flexibel ist.

Low Code / No Code für steuerliche Anwendungen

(Sehr allgemeine klarstellende/ergänzende) Unterschiede von Low Code / No Code

Low Code

- Adressaten sind Entwickler und Nicht-Entwickler
- Limitierung durch bereitgestellte Features des Anbieters
- Anwendung kann durch klassische Entwicklung ergänzt und so Limitierung durchbrochen werden
- Möglichkeit der Einbindung in die bestehende IT-Landschaft durch Programmierschnittstellen (API)

No Code

- Adressaten sind Nicht-Entwickler
- Limitierung durch bereitgestellte Features des Anbieters
- Herausforderung bei der Einbindung in die bestehende IT-Landschaft

Low Code / No Code für steuerliche Anwendungen

Argumente **für** Low Code / No Code

Allgemein

- Schnelle(re) Softwareentwicklung und Bereitstellung von Applikationen
- Erhöhung von Prozesseffizienzen
- Erstellung von hochgradig auf Kundenbedürfnisse abgestimmte Automationsworkflows (RPA, BPA)
- Kostenreduktion (?)
- Einsatz von Fachexperten (Nicht-IT-Experten / „Citizen Developer“) um IT-Fachkräftemangel zu begegnen
- Steigerung der Nutzerfreundlichkeit von Applikationen
- Agiles Projektvorgehen
- Förderung von IT-Verständnis der Fachexperten

Steuerkontext

- Schnelle(re) Reaktion auf (geänderte) Gesetzgebung, Rechtsprechung oder Verwaltungsauffassung
- Steuerexperten als „Citizen Developer“
- Beherrschung von steuerlichen Komplexitäten, die mit klassischer Softwareentwicklung nicht mehr beherrschbar ist

Low Code / No Code für steuerliche Anwendungen

Argumente **gegen** Low Code / No Code

Allgemein

- IT-Sicherheitsrisiken
- Schatten-IT und unkoordiniertes Vorgehen
- Hoher Wartungsaufwand (?)
- Datenschutz
- Fehlende Akzeptanz durch IT-Experten
- Konzept des „Citizen Developer“ setzt sich nicht durch

Steuerkontext

- Steuerexperten sind keine „Citizen Developer“ (?)

Low Code / No Code für steuerliche Anwendungen

Weitere Punkte beim Einsatz von Low Code / No Code

- Low Code und No Code Entwicklung ist Softwareentwicklung
- Klassische Aufgaben (über die (Erst)Entwicklung hinaus) muss begegnet werden:
 - Qualitätssicherung
 - Testbarkeit
 - Wartung
 - Betrieb (Cloud, Hybrid Cloud oder On Premise)
 - Weiterentwicklung
 - Datenschutz
 - Mandantenfähigkeit
 - Integration in die Systemlandschaft
- Analyse der Anwendungsfälle und Geschäftsanforderungen, die die Auswahl der einzusetzenden Low-Code-Plattformen bestimmen

Low Code / No Code für steuerliche Anwendungen

Antwort auf die Frage an die KI (MS Copilot) – „Wie kann KI hier zum Einsatz kommen?“ (Folgefrage)

Künstliche Intelligenz (KI) kann in Low-Code- und No-Code-Plattformen auf verschiedene Weise integriert werden, um die Entwicklung von Anwendungen zu vereinfachen und zu beschleunigen. Hier sind einige Beispiele:

1. Automatisierung und Effizienzsteigerung

Automatisierte Code-Generierung: KI kann dabei helfen, **Code automatisch zu generieren**, basierend auf den Anforderungen des Benutzers. Dies reduziert den manuellen Aufwand und beschleunigt die Entwicklungsprozesse.

Vorhersageanalysen: KI-Modelle können genutzt werden, um Vorhersagen zu treffen und Analysen durchzuführen, die in die Anwendungen integriert werden können.

2. Benutzerfreundlichkeit und Zugänglichkeit

Sprach- und Bilderkennung: KI-Funktionen wie Sprach- und Bilderkennung können in Anwendungen eingebaut werden, um die Benutzerfreundlichkeit zu erhöhen. Dies ermöglicht es Benutzern, Anwendungen durch Sprachbefehle oder Bilder zu steuern.

Chatbots und virtuelle Assistenten: KI-gesteuerte Chatbots können in Anwendungen integriert werden, um Benutzern bei Fragen und Problemen zu helfen.

3. Unterstützung für Citizen Developer

Intelligente Empfehlungen: KI kann **Empfehlungen geben, wie bestimmte Funktionen oder Module in einer Anwendung implementiert werden sollten**, basierend auf Best Practices und vorherigen Projekten.

Fehlererkennung und -behebung: KI kann dabei helfen, **Fehler im Code zu erkennen und Vorschläge zur Behebung zu machen**, was besonders für Benutzer ohne tiefgehende Programmierkenntnisse hilfreich ist.

4. Generative KI

Autogenerierung von Lösungen: Generative KI kann verwendet werden, um **automatisch Lösungsvorschläge für bestimmte Probleme zu erstellen**, was die Kreativität und Effizienz der Benutzer steigert.

Die Integration von KI in Low-Code- und No-Code-Plattformen eröffnet viele Möglichkeiten und macht die Entwicklung von Anwendungen zugänglicher und effizienter.

Low Code / No Code für steuerliche Anwendungen

Einsatz von KI

In der Entwicklung

- Codegenerierung (Durchbrechung der Limitierung durch bereitgestellte Features des Anbieters)
- Ausführung der „Citizen Developer Programmierung“ (Entwurfserstellung und -verbesserung) – „Citizen Developer“ wird zum Prompt Engineer
- Handlungs- und Umsetzungsempfehlungen für den „Citizen Developer“
- Fehlererkennung und Korrekturvorschläge (Qualitätssicherung und Testen)

In der Applikation

- Vorhersage- und Analysefeatures
- Sprach- und Bilderkennungsfeatures
- Chat Bots und virtuelle Assistenten

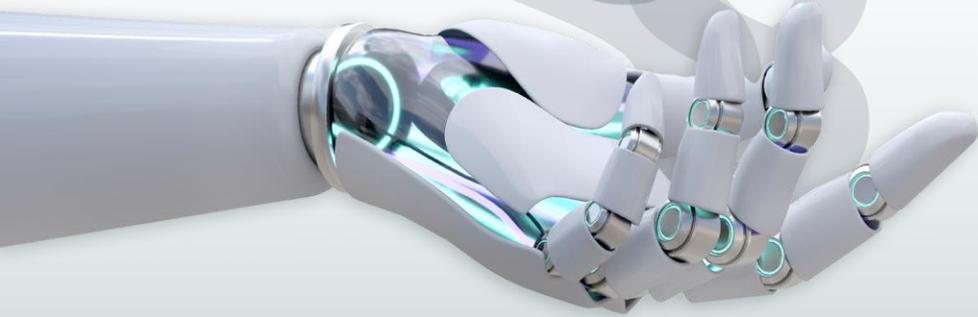
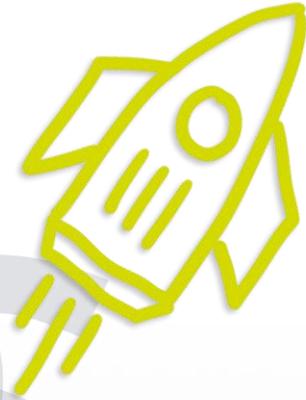
Low Code / No Code für steuerliche Anwendungen

Ausgewählte Literatur

- Krebs, Robles, Standt: Low Code before it was cool, .inf, Herbst 2024, abrufbar unter: <https://inf.gi.de/07/low-code-before-it-was-cool> (zuletzt abgerufen am: 23.10.2024).
- Endres, Grabmair, Hinrichs, Jennrich: Steuergesetze der Zukunft – digitaltauglich formuliert und digital anwendbar, REthinking Tax 2/2024, 68.
- Sultanow, Dyballa, Sebrak: Low Code, Ein Praxisleitfaden für den öffentlichen Sektor, Capgemini PoV, abrufbar unter: <https://www.capgemini.com/de-de/insights/research/low-code-ein-praxisleitfaden-fuer-den-oeffentlichen-sektor/> (zuletzt abgerufen am: 23.10.2024).
- CIO, CSO und Computerwoche: Studie No-Code/Low-Code 2023, abrufbar unter: <https://www.escriba.de/studie-no-code-low-code-2023/> (zuletzt abgerufen am: 23.10.2024).
- Gartner, Newsroom vom 13.12.2022, abrufbar unter: <https://www.gartner.com/en/newsroom/press-releases/2022-12-13-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-20-percent-in-2023> (zuletzt abgerufen am: 23.10.2024).
- Hüsing, Liekenbrock, Weber: Low-Code-Plattform-Anwendungen zur Transformation des Finanz- und Steuerbereichs, DB 42/2022, 2439.
- Fachausschuss VII des IDSt, Glossar Tax Technology, abrufbar unter: <https://idst.tax/wp-content/uploads/2022/11/IDSt-Glossar-TaxTech.pdf> (zuletzt abgerufen am: 23.10.2024).
- Bräutigam: Einsatz von Low-Code-Plattformen – Vom Versuch, Unstrukturiertes zu strukturieren, am Beispiel von Power Automate, REthinking Tax, 1/2022, 4.
- Weber: Digitale Souveränität des steuerlichen Beraters – Potenzial der Enterprise-Low-Code-Plattform A12 in der Steuerfunktion zur Umsetzung von digital-integrierten Lösungen, REthinking Tax, 2/2021, 17.
- *Ergänzung: Erfolgreiche Erprobung des Einsatzes von Low Code in zwei steuerlichen Master-Seminaren:* <https://taxtech.blog/2021/06/25/lowcoding-tax-apps-casestudy-an-der-uni-siegen/> und <https://insights.mgm-tp.com/de/digitale-steuerlehre-und-it-in-steuerpraxis-kooperation-mit-der-tu-chemnitz/>.

READY

FOR THE FUTURE



Vielen Dank!

Veranstaltungen

Donnerstag, 23.05.2024 - 12:30 Uhr bis 13:00 Uhr

Robotics Process Automation im Steuerbereich



Donnerstag, 20.06.2024 - 12:30 Uhr bis 13:00 Uhr

Process Mining für steuerliche Prozesse



Donnerstag, 25.07.2024 - 12:30 Uhr bis 13:00 Uhr

Natural Language Processing im Steuerbereich



Donnerstag, 19.09.2024 - 12:30 Uhr bis 13:00 Uhr

Schnittstellen / API



Donnerstag, 24.10.2024 - 12:30 Uhr bis 13:00 Uhr

Low Code / No Code für steuerliche Anwendungen



Donnerstag, 21.11.2024 - 12:30 Uhr bis 13:00 Uhr

Blockchain im Besteuerungsprozess



Donnerstag, 19.12.2024 - 12:30 Uhr bis 13:00 Uhr

Cloud (Computing) im Steuerbereich



Weitere kostenfreie Webcasts:

