

Low-Code-Entwicklung

Herzog Max, Loidl Patrick

February 25, 2025

Agenda

▶ Was ist Low-Code?

Agenda

- ▶ Was ist Low-Code?
- ▶ Low-Code vs Traditionelle Entwicklungsmethoden

Agenda

- ▶ Was ist Low-Code?
- ▶ Low-Code vs Traditionelle Entwicklungsmethoden
- ▶ Vor- und Nachteile

Agenda

- ▶ Was ist Low-Code?
- ▶ Low-Code vs Traditionelle Entwicklungsmethoden
- ▶ Vor- und Nachteile
- ▶ Mendix

Agenda

- ▶ Was ist Low-Code?
- ▶ Low-Code vs Traditionelle Entwicklungsmethoden
- ▶ Vor- und Nachteile
- ▶ Mendix
- ▶ Low-Code am Beispiel von Mendix

Agenda

- ▶ Was ist Low-Code?
- ▶ Low-Code vs Traditionelle Entwicklungsmethoden
- ▶ Vor- und Nachteile
- ▶ Mendix
- ▶ Low-Code am Beispiel von Mendix
- ▶ Vorteile für Unternehmen

Was ist Low-Code?

- ▶ Methode zur Softwareentwicklung

Was ist Low-Code?

- ▶ Methode zur Softwareentwicklung
- ▶ Anwendungen mit minimalem Code

Was ist Low-Code?

- ▶ Methode zur Softwareentwicklung
- ▶ Anwendungen mit minimalem Code
- ▶ Visuelle Entwicklungsumgebung

Was ist Low-Code?

- ▶ Methode zur Softwareentwicklung
- ▶ Anwendungen mit minimalem Code
- ▶ Visuelle Entwicklungsumgebung
- ▶ Bietet vorgefertigte Bausteine

Was ist Low-Code?

- ▶ Methode zur Softwareentwicklung
- ▶ Anwendungen mit minimalem Code
- ▶ Visuelle Entwicklungsumgebung
- ▶ Bietet vorgefertigte Bausteine
- ▶ Entwicklungsprozesse beschleunigen

Low-Code vs Traditionelle Entwicklungsmethoden

Low-Code:

- ▶ Schnelle App-Entwicklung

Low-Code vs Traditionelle Entwicklungsmethoden

Low-Code:

- ▶ Schnelle App-Entwicklung
- ▶ Grafische
Benutzeroberfläche

Low-Code vs Traditionelle Entwicklungsmethoden

Low-Code:

- ▶ Schnelle App-Entwicklung
- ▶ Grafische Benutzeroberfläche
- ▶ Reduzierte Komplexität

Traditioneller Code:

- ▶ Kenntnisse in den jeweiligen Programmiersprachen

Low-Code vs Traditionelle Entwicklungsmethoden

Low-Code:

- ▶ Schnelle App-Entwicklung
- ▶ Grafische Benutzeroberfläche
- ▶ Reduzierte Komplexität

Traditioneller Code:

- ▶ Kenntnisse in den jeweiligen Programmiersprachen
- ▶ Manuellen Code schreiben

Low-Code vs Traditionelle Entwicklungsmethoden

Low-Code:

- ▶ Schnelle App-Entwicklung
- ▶ Grafische Benutzeroberfläche
- ▶ Reduzierte Komplexität

Traditioneller Code:

- ▶ Kenntnisse in den jeweiligen Programmiersprachen
- ▶ Manuellen Code schreiben
- ▶ Flexible Softwaregestaltung

Low-Code vs Traditionelle Entwicklungsmethoden

Low-Code:

- ▶ Schnelle App-Entwicklung
- ▶ Grafische Benutzeroberfläche
- ▶ Reduzierte Komplexität

Traditioneller Code:

- ▶ Kenntnisse in den jeweiligen Programmiersprachen
- ▶ Manuellen Code schreiben
- ▶ Flexible Softwaregestaltung
- ▶ Längere Entwicklungszeiten

Merkmale von Low-Code

▶ Visuelle Entwicklungsumgebung

Merkmale von Low-Code

- ▶ Visuelle Entwicklungsumgebung
- ▶ Einfache Integration und Anpassung

Merkmale von Low-Code

- ▶ Visuelle Entwicklungsumgebung
- ▶ Einfache Integration und Anpassung
- ▶ Schnelle Entwicklung und Bereitstellung

Merkmale von Low-Code

- ▶ Visuelle Entwicklungsumgebung
- ▶ Einfache Integration und Anpassung
- ▶ Schnelle Entwicklung und Bereitstellung
- ▶ Breites Spektrum an Nutzern

Vorteile von Low-Code

- ▶ Schnelle Entwicklung

Vorteile von Low-Code

- ▶ Schnelle Entwicklung
- ▶ Benutzerfreundlichkeit

Vorteile von Low-Code

- ▶ Schnelle Entwicklung
- ▶ Benutzerfreundlichkeit
- ▶ Kosteneffizienz

Vorteile von Low-Code

- ▶ Schnelle Entwicklung
- ▶ Benutzerfreundlichkeit
- ▶ Kosteneffizienz
- ▶ Wartung

Nachteile von Low-Code

- ▶ Begrenzte Anpassungsfähigkeit

Nachteile von Low-Code

- ▶ Begrenzte Anpassungsfähigkeit
- ▶ Leistungsbeschränkungen

Nachteile von Low-Code

- ▶ Begrenzte Anpassungsfähigkeit
- ▶ Leistungsbeschränkungen
- ▶ Abhängigkeit von Plattformanbietern

Nachteile von Low-Code

- ▶ Begrenzte Anpassungsfähigkeit
- ▶ Leistungsbeschränkungen
- ▶ Abhängigkeit von Plattformanbietern
- ▶ Sicherheitsbedenken

Anwendungsbeispiele für Low-Code

- ▶ **Gesundheitswesen:** Schnellere Entwicklung von Patientenmanagement-Systemen

Anwendungsbeispiele für Low-Code

- ▶ **Gesundheitswesen:** Schnellere Entwicklung von Patientenmanagement-Systemen
- ▶ **Finanzsektor:** Anpassung von Banking-Anwendungen und Integrationen

Anwendungsbeispiele für Low-Code

- ▶ **Gesundheitswesen:** Schnellere Entwicklung von Patientenmanagement-Systemen
- ▶ **Finanzsektor:** Anpassung von Banking-Anwendungen und Integrationen
- ▶ **E-Commerce:** Anpassung von Online-Shops und Inventarsystemen

Anwendungsbeispiele für Low-Code

- ▶ **Gesundheitswesen:** Schnellere Entwicklung von Patientenmanagement-Systemen
- ▶ **Finanzsektor:** Anpassung von Banking-Anwendungen und Integrationen
- ▶ **E-Commerce:** Anpassung von Online-Shops und Inventarsystemen
- ▶ **Öffentlicher Sektor:** Optimierung von Verwaltungsprozessen und Formularen

Anwendungsbeispiele für Low-Code

- ▶ **Gesundheitswesen:** Schnellere Entwicklung von Patientenmanagement-Systemen
- ▶ **Finanzsektor:** Anpassung von Banking-Anwendungen und Integrationen
- ▶ **E-Commerce:** Anpassung von Online-Shops und Inventarsystemen
- ▶ **Öffentlicher Sektor:** Optimierung von Verwaltungsprozessen und Formularen
- ▶ **Bildung:** Entwicklung von Lernplattformen und Verwaltungssystemen

Mendix



- ▶ **Marktführende Low-Code Plattform**
- ▶ 2005 in den **Niederlanden** gegründet
- ▶ 2018 von Siemens für **730 Millionen €** gekauft

Mendix



- ▶ **Marktführende Low-Code Plattform**
- ▶ 2005 in den **Niederlanden** gegründet
- ▶ 2018 von Siemens für **730 Millionen €** gekauft
- ▶ **Mendix aktuell:**
 - ▶ mehr als 300.000 **aktive Entwickler**
 - ▶ über 4.000 **Kunden**
 - ▶ ca. 50 Millionen **Endnutzer**
 - ▶ mehr als 200.000 **entwickelte Apps**

Domain Modell

- ▶ Visuelle Darstellung der Datenstruktur

Domain Modell

- ▶ Visuelle Darstellung der Datenstruktur
- ▶ Zeigt Beziehungen zwischen Datenobjekten

Domain Modell

- ▶ Visuelle Darstellung der Datenstruktur
- ▶ Zeigt Beziehungen zwischen Datenobjekten
- ▶ Unterstützt datenbasierte Entwicklung

Assoziationen

- ▶ Beziehungen zwischen Entitäten

Assoziationen

- ▶ Beziehungen zwischen Entitäten
- ▶ Beispiel: Kunde - Bestellungen

Assoziationen

- ▶ Beziehungen zwischen Entitäten
- ▶ Beispiel: Kunde - Bestellungen
- ▶ Effiziente Datenverwaltung

Grafische Benutzeroberfläche (GUI)

- ▶ Intuitive visuelle Oberfläche

Grafische Benutzeroberfläche (GUI)

- ▶ Intuitive visuelle Oberfläche
- ▶ Elemente per Drag-and-Drop gestalten

Grafische Benutzeroberfläche (GUI)

- ▶ Intuitive visuelle Oberfläche
- ▶ Elemente per Drag-and-Drop gestalten
- ▶ Keine manuelle Programmierung nötig

Widgets

- ▶ Vorprogrammierte, wiederverwendbare Bausteine

Widgets

- ▶ Vorprogrammierte, wiederverwendbare Bausteine
- ▶ Beispiele: Buttons, Data-Views, Textboxes

Widgets

- ▶ Vorprogrammierte, wiederverwendbare Bausteine
- ▶ Beispiele: Buttons, Data-Views, Textboxes
- ▶ Erleichtert Entwicklung durch Wiederverwendung

Microflows

▶ Visuelle Workflows für Logik

Microflows

- ▶ Visuelle Workflows für Logik
- ▶ Automatisieren von Prozessen

Microflows

- ▶ Visuelle Workflows für Logik
- ▶ Automatisieren von Prozessen
- ▶ Beispiele: Datenabruf, Berechnungen

Vorteile für Unternehmen

Beschleunigte Anwendungsentwicklung:

- ▶ Schnelle Reaktion auf Marktanforderungen

Vorteile für Unternehmen

Beschleunigte Anwendungsentwicklung:

- ▶ Schnelle Reaktion auf Marktanforderungen
- ▶ Agile Produktentwicklung

Vorteile für Unternehmen

Beschleunigte Anwendungsentwicklung:

- ▶ Schnelle Reaktion auf Marktanforderungen
- ▶ Agile Produktentwicklung
- ▶ Drag-and-Drop Development spart Zeit und Kosten

Effiziente Ressourcennutzung:

- ▶ Schnelles Coding für unerfahrene Mitarbeiter

Vorteile für Unternehmen

Beschleunigte Anwendungsentwicklung:

- ▶ Schnelle Reaktion auf Marktanforderungen
- ▶ Agile Produktentwicklung
- ▶ Drag-and-Drop Development spart Zeit und Kosten

Effiziente Ressourcennutzung:

- ▶ Schnelles Coding für unerfahrene Mitarbeiter
- ▶ Automatisierung von Routineaufgaben

Vorteile für Unternehmen

Beschleunigte Anwendungsentwicklung:

- ▶ Schnelle Reaktion auf Marktanforderungen
- ▶ Agile Produktentwicklung
- ▶ Drag-and-Drop Development spart Zeit und Kosten

Effiziente Ressourcennutzung:

- ▶ Schnelles Coding für unerfahrene Mitarbeiter
- ▶ Automatisierung von Routineaufgaben
- ▶ Mehr Zeit für strategische Aufgaben