

Fähigkeiten und Vorgehen von Low Code Ansätzen



[1]

“52% der IT-Vorhaben haben zumindest teilweise nicht die Wünsche und Anforderungen der Auftraggeber erfüllt. 19% der Projekte sind ein Totalausfall und wurden abgebrochen. Nur 29% der untersuchten Projekte waren total erfolgreich.”

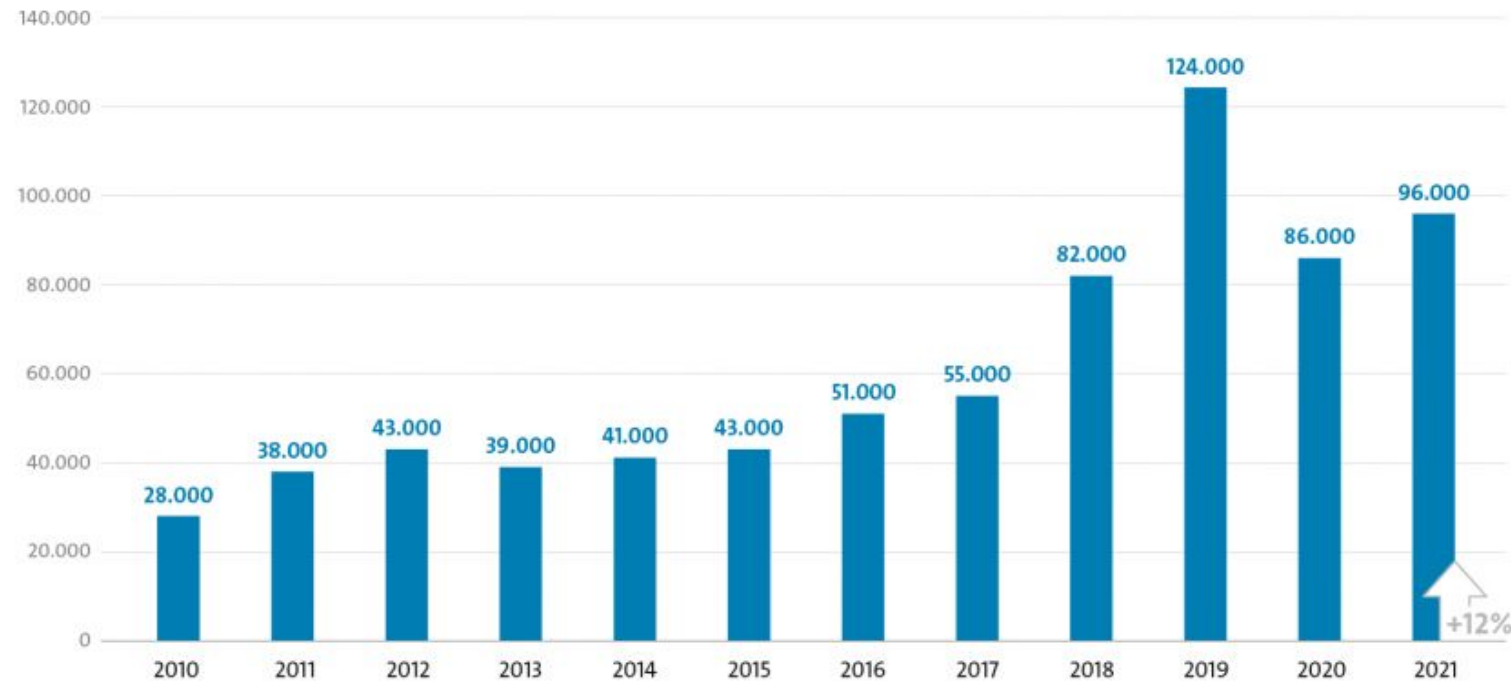
[2] „Chaos Report“ der Standish Group, USA, 2015



[3]

96.000 unbesetzte Stellen für IT-Fachkräfte

Anzahl zu besetzender IT-Stellen in der deutschen Gesamtwirtschaft



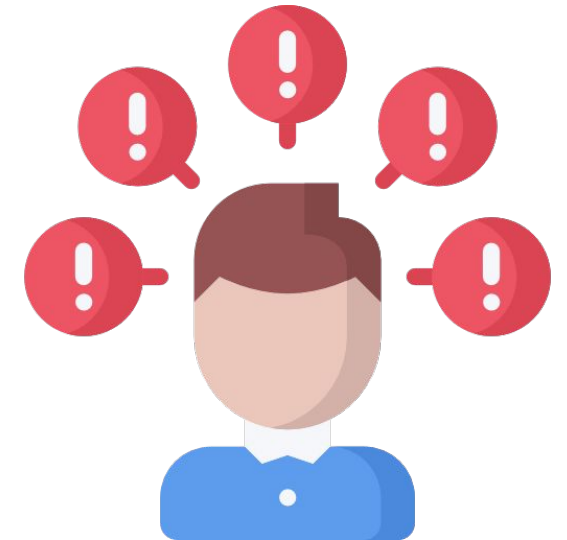
Basis: Unternehmen ab 3 Beschäftigte in Deutschland
Quelle: Bitkom Research

bitkom



Problempunkte

- unzureichende Definition der Anforderungen / Nutzer zu wenig in Entwicklung einbezogen
- viel “Boilerplate”-Code → wenig Zeit zum Arbeiten an wichtigen Problemen
- nach Entwicklung: hoher Ressourcenaufwand für Wartung von Software durch hohe Komplexität



[3]

Gliederung

1. Generationen von Programmiersprachen
2. Was ist Low Code?
3. Vergleich
4. Einschränkungen
5. Fazit
6. Ausblick

Generationen von Programmiersprachen

1 GL. **Maschinensprachen**

2 GL. **Assembler Sprache**

3 GL. **Prozedurale Programmierung,
Objektorientierte Programmierung**

4 GL. **Höherer Abstraktionsgrad**

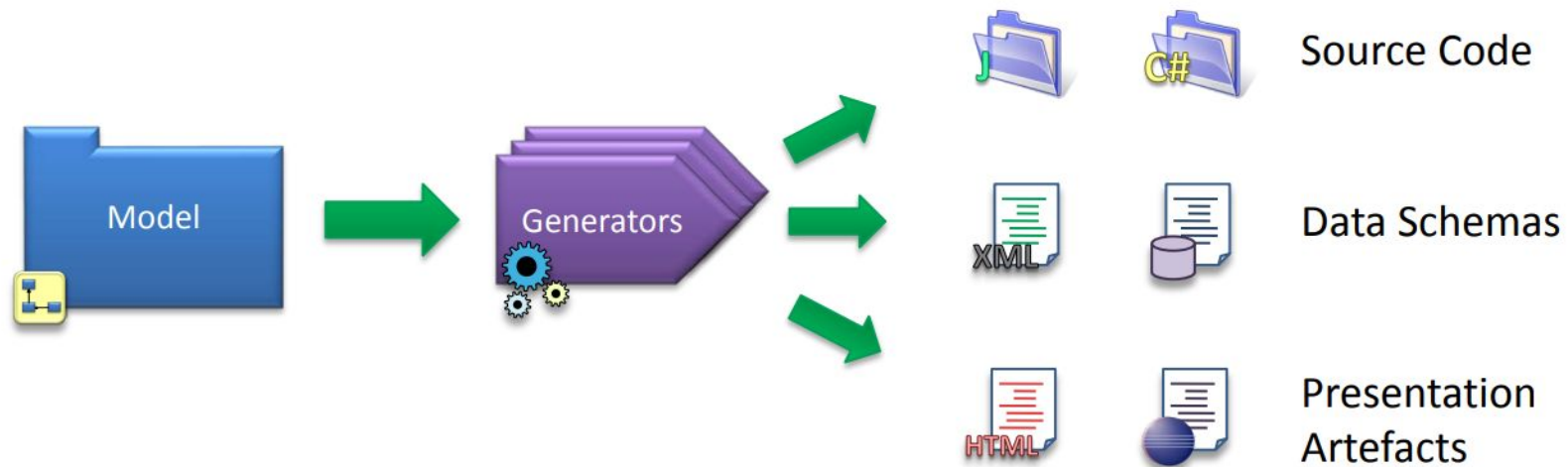


[3]

Model Driven Development

Modellbasierte Software Entwicklung

Modellgetriebene Software Entwicklung



[5]

Was ist Low- Code?

Modellierung über eine visuelle Oberfläche

vorgefertigte Code Bausteine

basierend auf Workflows und Datenmodellen

Ziele von Low-code

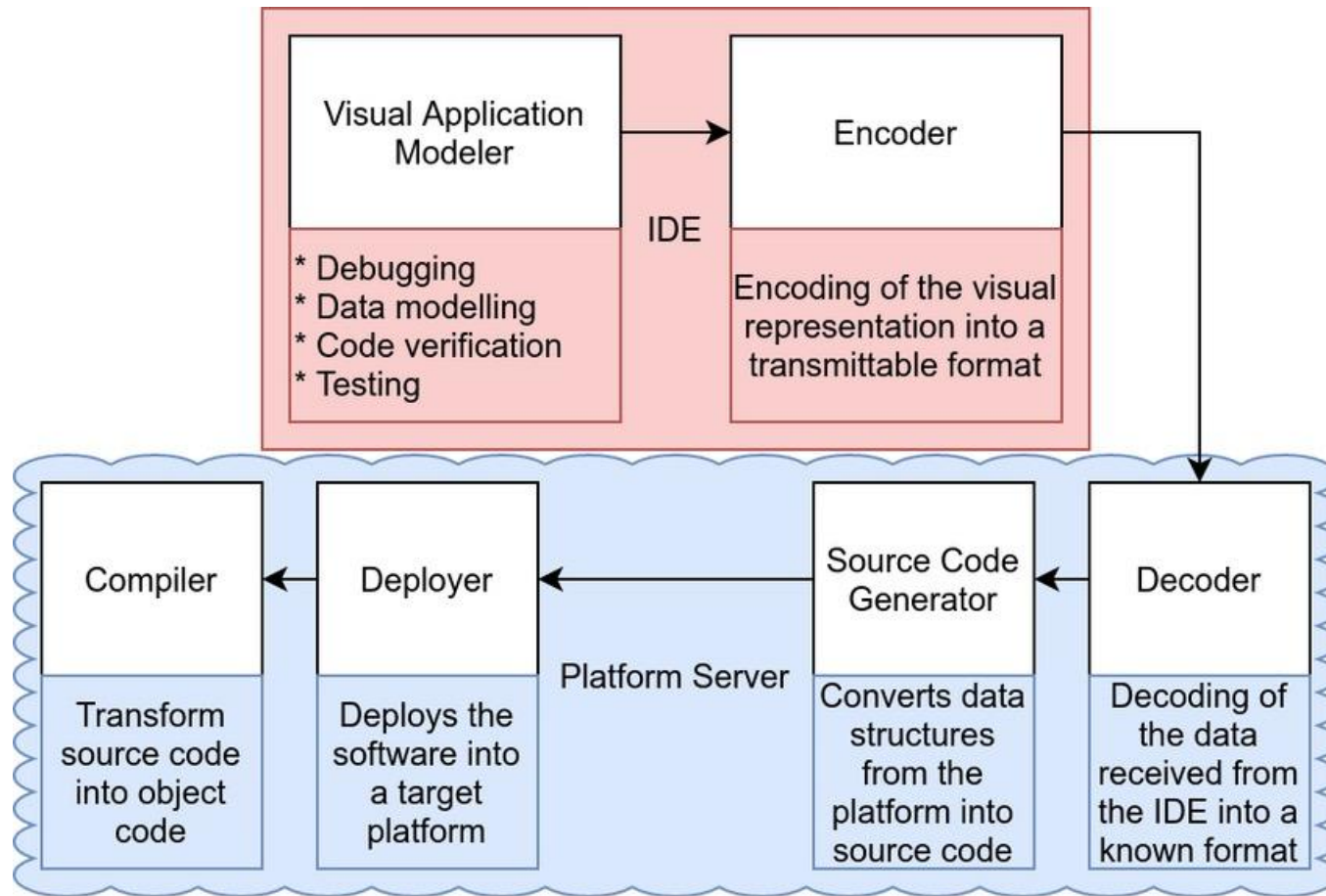
kurze Entwicklungszeit / Schnelligkeit

Weniger Komplexität

Kosteneinsparung

Wenig technische Anforderungen

Architecture von Low-code



[6]

Low-Code zur Traditionellen Programmierung

Vergleich

Unser Beispiel

- Versicherungsanwendung
- vier verschiedene Fälle
- jeweils andere Felder
- Speicherung in einer Datenbank



[7]

What's happened?

Accident

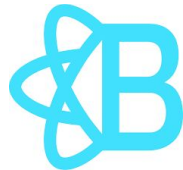
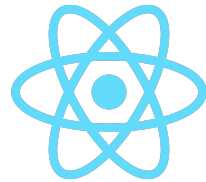
Fire

Theft

Other

Verwendete Werkzeuge

Programmierung



von links: [8], [9], [10], [11], [12]

Low Code



[13]

Magic Quadrant

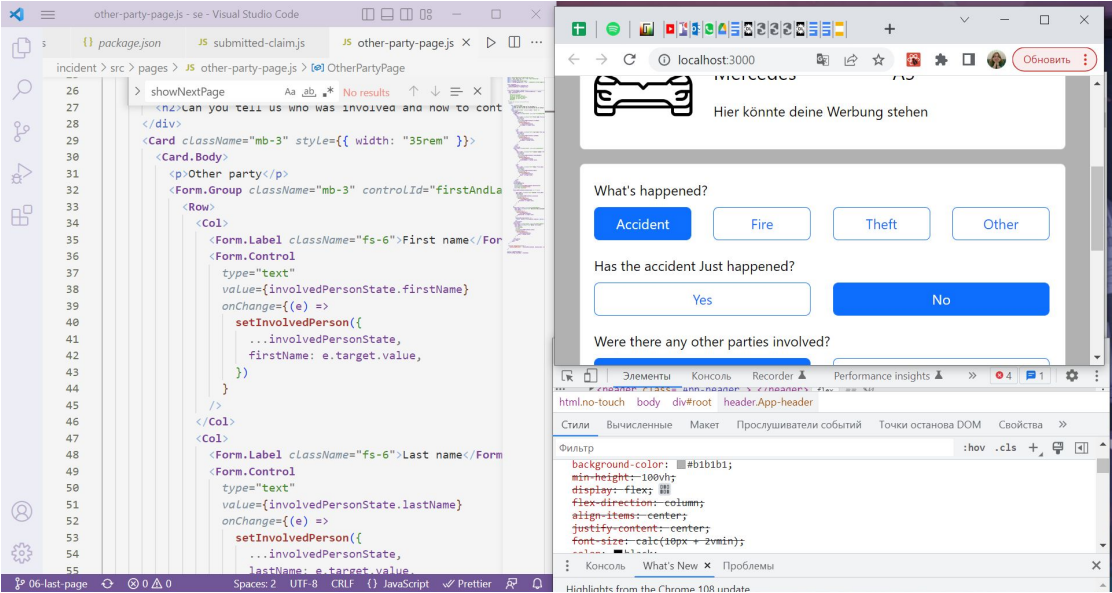


Source: Gartner (August 2021)

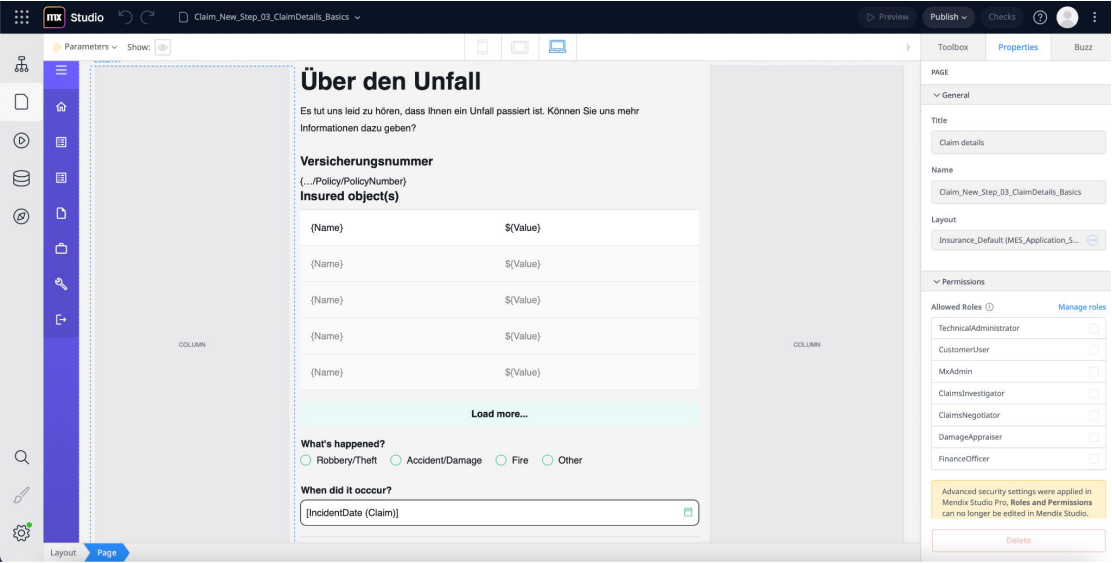
die Fähigkeit zur Ausführung
die Vollständigkeit der Vision

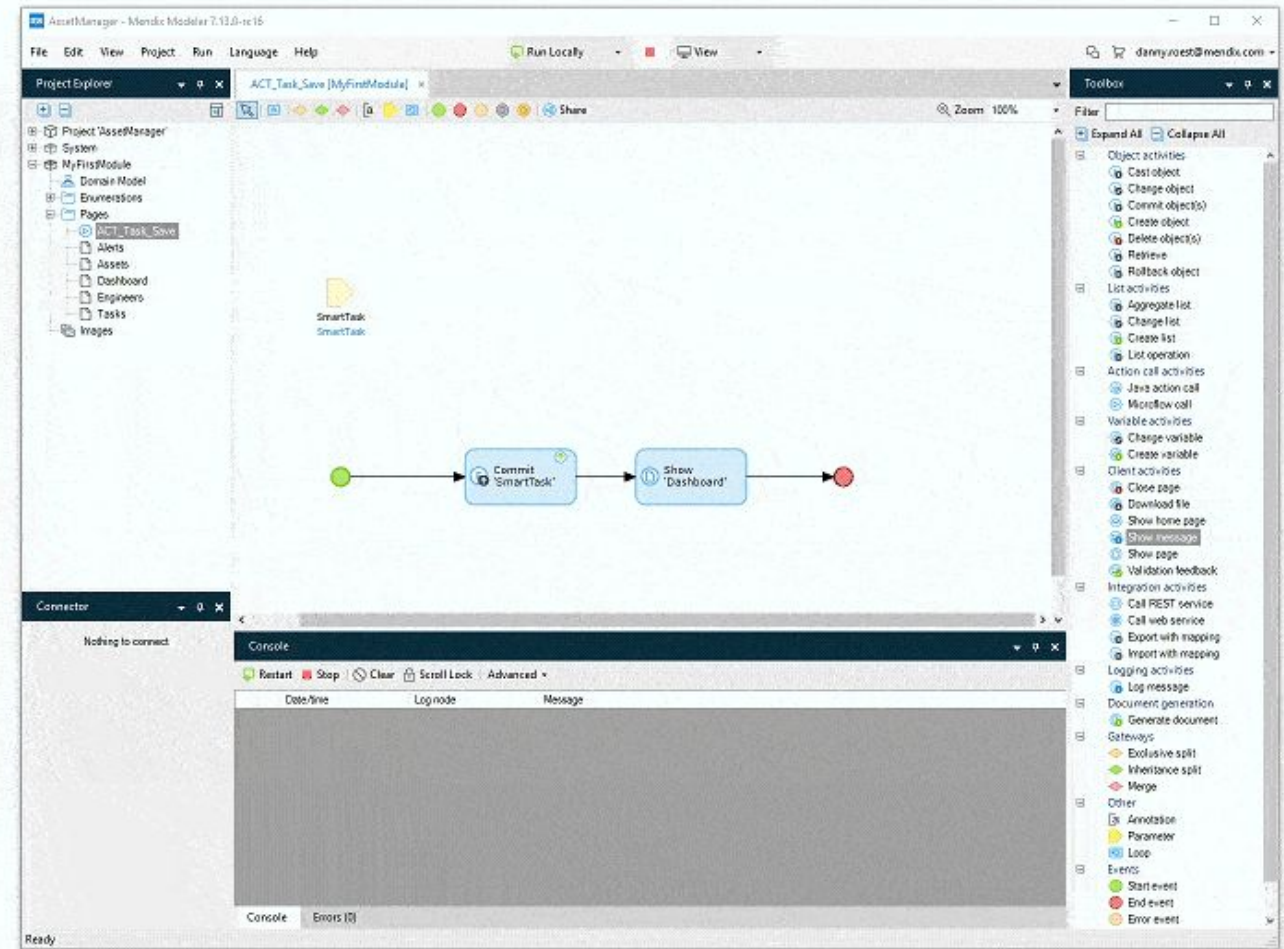
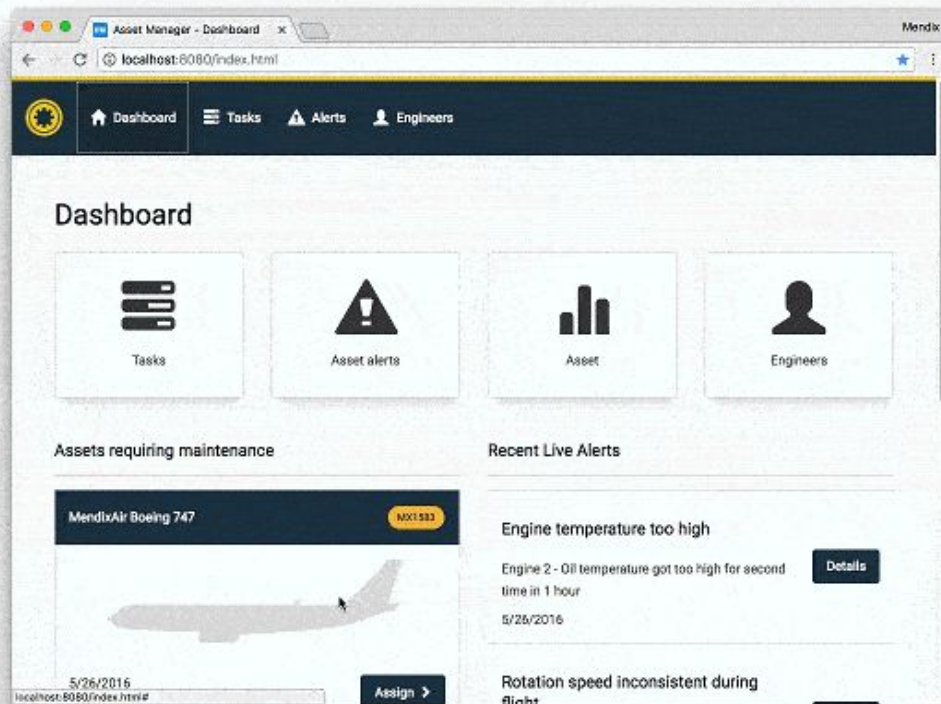
GUI

Programmierung



Low Code

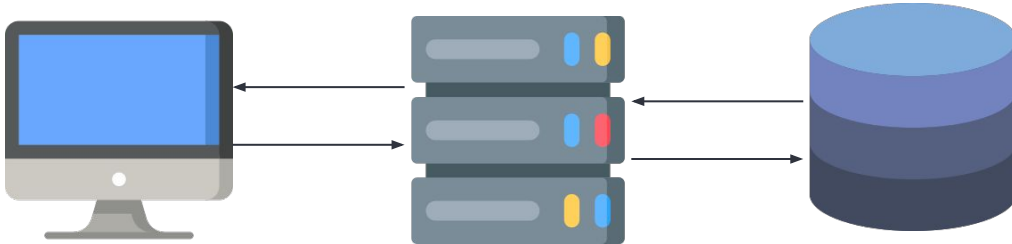




Logik

Programmierung

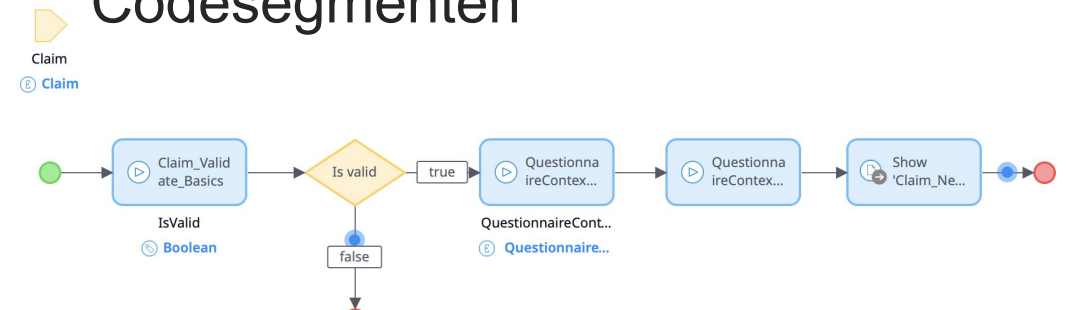
- Client-Server Architecture
- Der Entwickler durchdenkt die Logik der Anwendung



[3]

Low Code

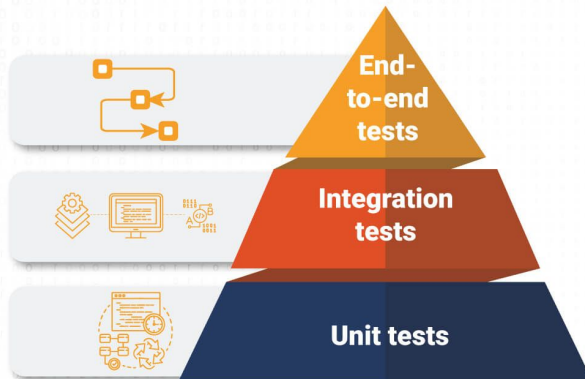
- Funktionen über Flows, Microflows und Datenmodelle
- Nutzung von vorgefertigten und selbst geschriebenen Codesegmenten



Qualitätssicherung

Programmierung

- manuelle Tests oder/und
- die Entwicklung automatisierter Tests sind erforderlich



[16]

Low Code

- alle Komponenten sind vorgetestet
- Tests werden automatisch im Hintergrund durchgeführt
- Anbindung von Selenium oder JUnit möglich



[17], [18]

Wiederverwendbarkeit

Programmierung

- Erstellung von Komponenten/Klassen und eigenen Bibliotheken
- leichte Änderbarkeit



[3]

Low Code

- Speicherung von eigenen Komponenten möglich
- Wiederverwendbarkeit Plattform intern



[3]

Zeit

Konventionell (ReactJS)	Aufgabenteil	Low Code (Mendix)
1 h	Einarbeitung	10 h
3,5 h	Aufsetzen des Projekts	0 h
40 h	Entwicklungszeit	15h
44,5 h	Gesamt	25 h

Vergleiche in Literatur

Table 6: Operations average for all experiments

Operation	Java Swing	Low Code	JavaScript
Code lines	55.50	38.00	66.50
Execution runtime	473.49 ms	1288.60 ms	35.20 ms
Development time	34.00 min	20.50 min	22.50 min
Operations runtime	5.12 ms	0.54 ms	0.310 ms

[???

Grenzen von Low Code

- vor allem für nicht innovative Projekte geeignet
- begrenzte Integrationsmöglichkeiten
- Wahl des Tools muss auf Projekt abgestimmt sein



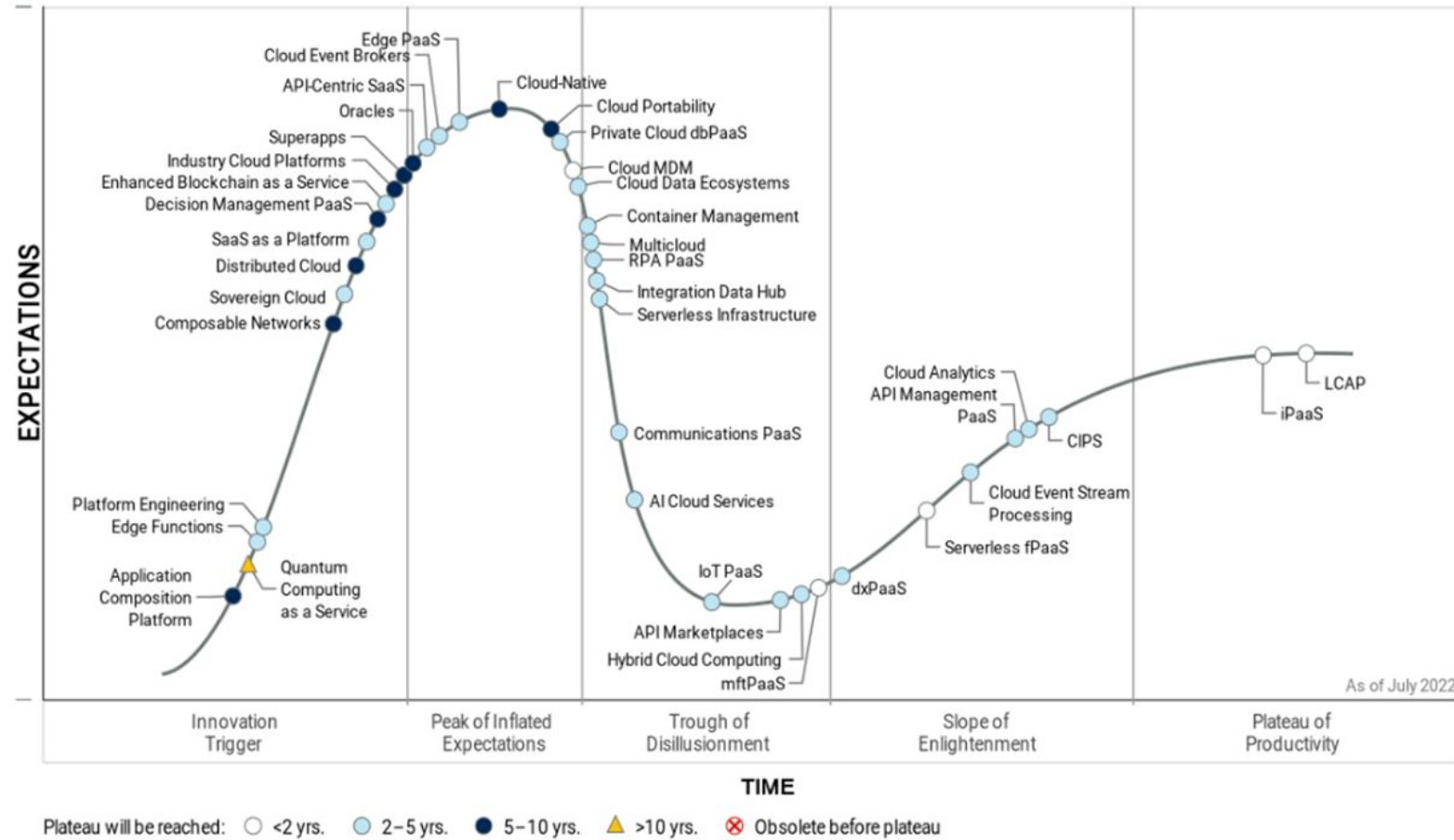
[3]

Fazit

- schnelle Auf- und Umsetzung von Projekten
- für komplexe Anwendungen eher ungeeignet
- große Abhängigkeit von Tools



Ausblick



[??]

Quellen

- [1] https://www.planetcrust.com/why-low-code-is-the-best-for-building-software-solutions?utm_campaign=blog
- [2] https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
- [3] <https://www.flaticon.com/de/>
- [4] <https://www.bitkom.org/Presse/Presseinformation/IT-Fachkraefteluecke-wird-groesser>
- [5] https://wiki.eclipse.org/images/8/8d/Using_MDD_Eclipse_Technology_to_implement_SOA.pdf
- [6] https://www.researchgate.net/publication/354862325_OLP-A_RESTful_Open_Low-Code_Platform
- [7] <https://volvocarautoleader.ru/kuzovnoy-remont>
- [8] https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Node.js_logo.svg
- [9] <https://ru.wikipedia.org/wiki/JavaScript>
- [10] <https://ru.wikipedia.org/wiki/MySQL>
- [11] <https://reactjs.org/>
- [12] <https://react-bootstrap.github.io/>
- [13] https://commons.wikimedia.org/wiki/File:Mendix_logo.svg
- [14] <https://www.gartner.com/doc/reprints?id=1-2C8VSOAH&ct=230113&st=sb>
- [15] <https://www.mendix.com/blog/mendix-7-13-a-bakers-dozen-of-developer-updates/>
- [16] <https://www.headspin.io/blog/the-testing-pyramid-simplified-for-one-and-all>
- [17] <https://www.selenium.dev/>
- [18] <https://ru.wikipedia.org/wiki/JUnit>
- [19] Calçada, André, and Jorge Bernardino. "Experimental Evaluation of Low Code Development, Java Swing and JavaScript Programming." *International Database Engineered Applications Symposium*, September 22, 2022. <https://doi.org/10.1145/3548785.3548792>.
- [20] <https://www.gartner.com/en/newsroom/press-releases/2022-08-04-cloud-platform-hc-press-release>

Diskussion

Habt ihr schon Erfahrungen mit Low Code Entwicklung gesammelt?

Wird Low Code eurer Meinung nach in der Zukunft traditionelles Programmieren ersetzen?