

# Time to Market als entscheidender Faktor

18 Juni 2020 | **Business Innovation** | [Marc Faeh](#), [Simon Schaefer](#)

**Lesezeit:** 6 Minutes

**Schnell sein und eine Idee zu einem marktreifen Produkt bringen war nie wichtiger als heute. Nebst einer guten Idee und einer überzeugenden User Experience macht der Faktor Zeit oft den Unterschied, ob ein Produkt wie eine App oder ein Kundenportal im Markt durchstartet.**

Häufig steht die Frage im Raum, warum die IT und deren Lieferanten so lange brauchen, um selbst vermeintlich einfache Dinge umzusetzen. Schaut man sich ein klassisches Projekt genauer an, muss man zu Beginn im Besten Fall ein bis zwei Monate in die technische Basis investieren. Darunter fallen Dinge, die man als Benutzer kaum wahrnimmt aber selbstverständlich voraussetzt: Ein Datenmodell, eine Login-Möglichkeit, die IT -Infrastruktur, die Anbindung von Datenquellen oder die Ausgabe auf mobilen Geräten.

Genau dort setzen Low-code Plattformen an: Sie stellen einen Entwicklungs-Baukasten zur Verfügung, um rasch modell-getrieben eine Lösung zu entwickeln. Dies ermöglicht es auch Mitarbeitern aus der Fachabteilung ohne Programmierkenntnisse aktiv an der Entwicklung einer Software Applikation mitzuarbeiten. Typische, oft gebrauchte Funktionen wie eine Workflow-Engine oder ein Login-Verfahren aber auch die Infrastruktur stehen ab dem ersten Projekttag bereit.

## Was ist Low-Code?

Low-Code ist ein Werkzeugkasten, welcher einer Unternehmung hilft, rasch und unkompliziert Produkte im Software Umfeld zu entwickeln. Es gibt verschiedene Produkte auf dem Markt, wobei die folgenden Merkmale eine umfassende Low-Code Plattform ausmachen:

- **Visual Development Tools:** Visuelle, drag-and-drop Modelle werden benutzt, um Datenmodelle, Prozesse oder Oberflächen zu gestalten. Dadurch können auch Technik Laien aktiv an Applikationen mitarbeiten und kleinere Anpassungen direkt selbst vornehmen.
- **Application Store:** Eine Community oder ein Marktplatz, wo Plattform-Benutzer Plug-Ins, Komponenten oder Integrationen finden und auch selbst publizieren können.
- **Full Application Lifecycle (SDLC) Support:** Die meisten Low-Code-Plattformen decken den gesamten Lebenszyklus ab. Das bedeutet, dass auch Business Analyse, Produkt-Management, Testing oder Betrieb Teil der Plattform-Fähigkeiten sind.

## Projekte dreimal schneller - geht das?

Die Low-Code Anbieter versprechen Verbesserungen, welche jedoch mit gewisser Vorsicht zu sehen sind. Wie immer gilt auch hier die Devise: Ein eingespieltes Team mit einem klaren Ziel

und den notwendigen Skills wird durch ein Low-Code Baukasten nicht überflüssig, sondern noch wichtiger als sonst, um die sich bietenden Tempo-Chancen auch tatsächlich zu nutzen. Die Projektlaufzeit kann dabei ohne Weiteres auf einen Drittel der normal zu veranschlagenden Dauer verkürzt werden. Gewisse Hersteller versprechen noch grössere Produktivitätssteigerungen. Dies mag bei Idealbedingungen möglich sein. Die Praxis zeigt jedoch, dass die meisten Unternehmen Zeit brauchen, Ergebnisse zu prüfen, Feedback zu geben und Entscheidungen zu treffen. Die Anbindung an die bestehende Architektur ist zudem auch bei hervorragenden Integrationswerkzeugen nach wie vor mit Entwicklungs- und Test-Aufwänden verbunden.

### Praxisbeispiele

- **Portfolioanalyse für Immobilien-Investoren:** Im Zentrum des Projekts steht die Vision, das Know-How des Auftraggebers in eine moderne und flexible Webapplikation umzusetzen. Während der gesamten Laufzeit des Projekts werden laufend neue Ideen einbezogen und umgesetzt. Eine enge Zusammenarbeit und die Verfügbarkeit der Testumgebung ab Start ermöglichen es, bereits viel mehr umzusetzen, als ursprünglich geplant war.
- **Interaktionsdatenbank für Wirkstoffe (reguliertes Software-Projekt):** Eine bewährte Interaktionsdatenbank für Wirkstoffe für medizinisches Fachpersonal wird für die Zukunft in eine moderne Webapplikation überführt. Auch hier fließen neue Ideen und Optimierungen und Features noch während der Entwicklung in das Projekt ein. Die resultierende Webapplikation ist State-of-the-art, wird in Spitälern, Kliniken und Arztpraxen eingesetzt und zeigt, dass Low-Code auch im regulierten Umfeld funktioniert.
- **Reales Prototyping bei einer Versicherung:** Um die Machbarkeit und den Nutzen einer neuen Idee direkt umzusetzen und interne Stakeholder zu überzeugen, setzt ein kleines Team eine lauffähige und erlebbare Webapplikation in kurzer Zeit um. Während andere noch Konzepte und Anträge schreiben, validiert das Team die Idee bereits mit verschiedenen Abteilungen und lässt Feedback direkt einfließen.

### Mit Low-Code zum Erfolg

Was gilt es zu beachten beim Einsatz von Low-Code?

- **Business Case gut planen:** Der Einsatz von Low-Code muss im Rahmen des Business Cases geprüft werden. Der Nutzen wächst über die Zeit und kommt besonders bei einer vollen Roadmap zum Tragen. Zu beachten sind auch die Lizenzmodelle, um böse Überraschungen nach der ersten Produktivitätseuphorie zu vermeiden.
- **Schnelle Prototypen & Feedback-Zyklen:** Mit Low-Code kann innerhalb weniger Tage ein Prototyp gebaut und lauffähig in die Cloud gebracht werden. Dadurch sind schnelle Feedback Zyklen und Entscheidungen möglich. Im Unterschied zu klickbaren Prototypen sogar mit echten Daten und tatsächlich nutzbar.

- **Gute Basisqualität durch Reuse:** Indem man auf einer etablierten Low-Code-Plattform aufsetzt, sind viele Fehler schon gemacht und korrigiert worden. Dadurch kann der Fokus auf die Funktionalität, den Benutzer und kundenspezifische Aspekte gelegt werden. Im App-Store stehen etliche Komponenten einer wachsenden Community von Dritten zur Verfügung. Allein dadurch konnten wir mehrmals in unseren Projekten Zeit und Kosten sparen.
- **Transparente agile Zusammenarbeit:** Die aus der agilen Softwareentwicklung bekannte Vorgehensweise können Sie 1:1 weiterführen. Der Produktivitätsgewinn bringt jedoch noch kürzere Feedbackzyklen, was insbesondere das Business freuen dürfte. Durch die Plattform-Integration von der Idee bis zum Betrieb ist am Schluss einer Phase keine aufwändige Betriebsübergabe notwendig.
- **Handwerk bleibt Handwerk:** Low-Code Plattformen bieten eine Menge aber sie ersetzen nicht die Erfahrung eines eingespielten Teams und eines Architekten. Die Fähigkeit, den Geschäftsnutzen zu verstehen, eine Vision zu entwickeln und in brauchbare Software umzusetzen bleibt zentral. Low-Code macht diese Teams jedoch schneller und erlaubt Abkürzungen, die in der traditionellen Software Entwicklung weniger möglich sind.
- **Überführung in Betrieb:** Der Betrieb ist in klassischen Projekten ein oft mühsamer Prozess und wird oft unterschätzt. Mit Low-Code sind wir bereits zu Beginn mit einem Bein im Betriebsmodus. Alles was ein Betrieb benötigt ist schon da. Dies macht die die Überführung in den Betrieb unkompliziert und schnell und ermöglicht, sich rasch der weiteren Produktentwicklung zu widmen.
- **Letzte Meile:** Die letzten 20 Prozent einer Produktentwicklung sind oftmals geprägt von hoher Individualität und technischen Besonderheiten. Hier sinkt die Produktivität auf ein «normales» Level. Im Gegensatz zu Projekten mit Standardsoftware gelangten wir technisch jedoch nie an Grenzen oder Einschränkungen.

### **Fazit**

Die Entscheidung zwischen Individualentwicklung oder Low-Code ist nicht immer einfach und hat durchaus langfristige Implikationen, da man sich an einen Hersteller bindet ähnlich wie beim Einsatz von Standardsoftware. Um neues zu schaffen, eine Vision Realität werden zu lassen und zu validieren ist Low-Code sehr geeignet. Die Kosten verschieben sich teilweise hin von Initialkosten zu operativen Kosten. Dafür erstellt man damit skalierbare und leicht erweiterbare Produkte, die nicht beim ersten Anstieg der Nutzerbasis bereits eine Nachfolgelösung verlangen.