LOW-CODE-PLATTFORMEN

CODEN OHNE CODE

Agile und schnelle Umsetzung von Softwareprojekten – das versprechen Low-Code-Plattformen. Doch was müssen Unternehmen berücksichtigen, wenn sie den manuellen Programmieraufwand mit entsprechenden Entwicklertools reduzieren wollen? Eine Orientierungshilfe.

TEXT **PETER SCHRÖDER**

Wenn GitHub-Mitgründer Chris Wanstrath davon spricht, dass die Zukunft des Codens eine sei, in der gar nicht mehr programmiert werde, dann hört die Fachwelt hin. Die gesamte Entwicklung von Software werde irgendwann automatisiert ablaufen, manueller Code-Aufwand immer weniger gebraucht, prognostizierte er in seiner Keynote auf der Entwicklerkonferenz GitHub Universe 2017.

Heute gibt es bereits sogenannte No-Code- und Low-Code-Plattformen (NCD/LCD), die es Entwicklern ermöglichen, mit immer weniger manuell geschriebenem Code Software und Applikationen zu entwickeln. Im Gegensatz zu sogenannten No-Code-Plattformen, bei denen das Coden komplett durch Fertigbausteine und Drag-&-Drop-Systeme ersetzt wird, lassen sich mit LCD-Plattformen auch komplexe – gar geschäftskritische – Unternehmensanwendungen umsetzen. Zwar ist auch bei Low-Code das Grundprinzip, dass in einer grafischen Entwicklungsumgebung Applikationen aus vorgefertigten Bausteinen zusammengesetzt werden. Doch können Entwickler diese Bausteine mittels klassischen Codes ergänzen und individuell anpassen.

Doch warum sollte es für Unternehmen interessant sein, manuelles Programmieren zu reduzieren? Wer sich heute in der IT-Wirklichkeit umschaut, der erkennt schnell: Die Backlogs in den Unternehmen wachsen und wachsen. Ob es darum geht, ein (veraltetes) Legacy-System zu migrieren oder eine Digitalisierungsstrategie oder Prozessoptimierung umzusetzen. Ob es heißt, Lücken durch eine neue oder bessere Integration zu schließen oder digitale Mehrwerte mithilfe von Apps mit besserer Nutzererfahrung zu erstellen. Deutschlands Unternehmen verwalten meist den Mangel, anstatt neue Vorhaben mit Elan zu beginnen. Warum? Weil die klassischen Methoden der Softwareentwicklung schlicht zu aufwendig sind. Und weil eine Ressource dabei eigentlich immer zu knapp ist: der Entwickler.

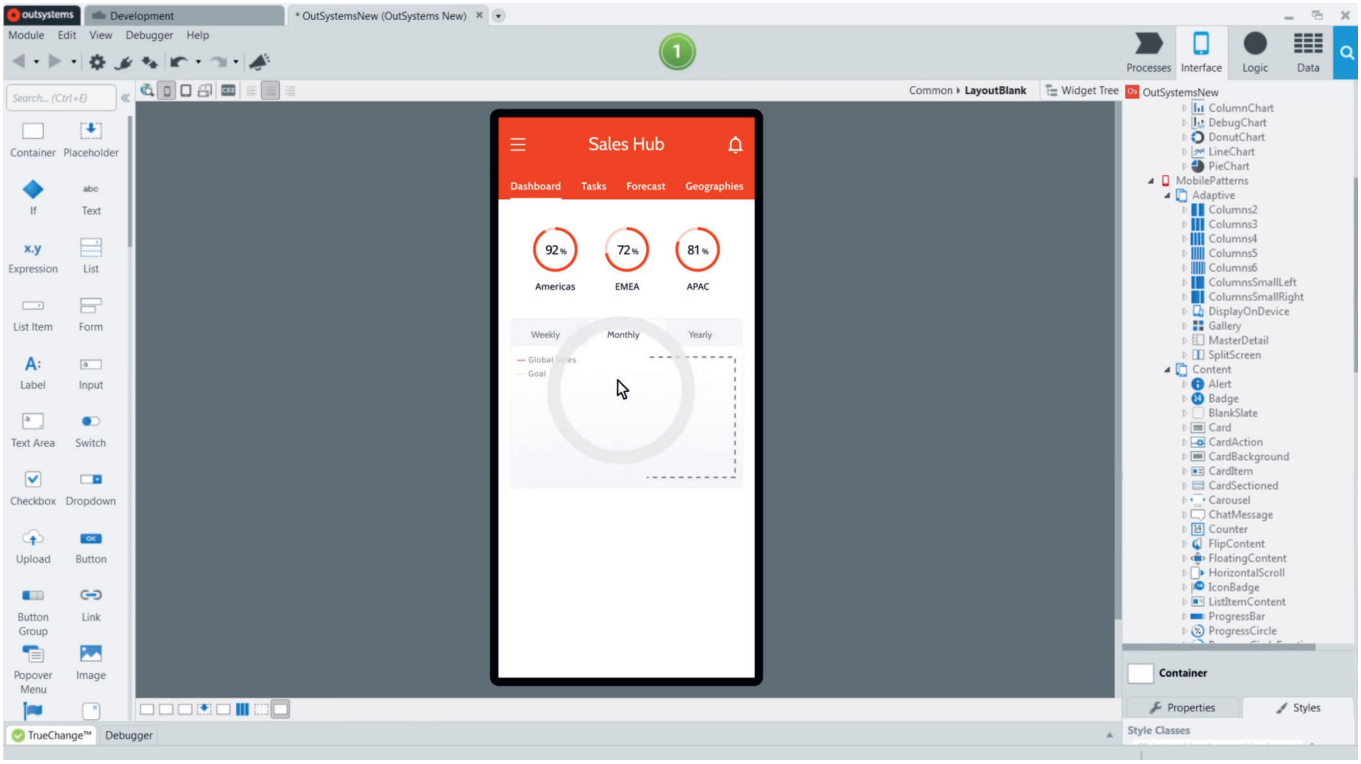
Low-Code-Development (LCD) soll dieses Problem lösen. Denn mittels Low-Code-Plattformen können Teams agil und schnell erste, funktionstüchtige Applikationen erstellen – und zwar über eine visuelle Modellierung. Damit braucht das LCD viel weniger Code als die klassische textbasierte Programmierung. So können auch technisch unerfahrene Personen selbst Prototypen erstellen oder zumindest erste visuelle Entwürfe kommentieren. Gleichzeitig reduziert das LCD den Aufwand für das Setup von Entwicklungsteams und erleichtert die iterative Entwicklung anhand des Prototyps. Handgeschriebener Code ist dabei zwar möglich, aber nicht unbedingt notwendig. Außerdem können Entwicklungsteams so Module einfach wiederverwenden und die Software über ihren gesamten Lebenszyklus hinweg ressourcenschonender verwalten.

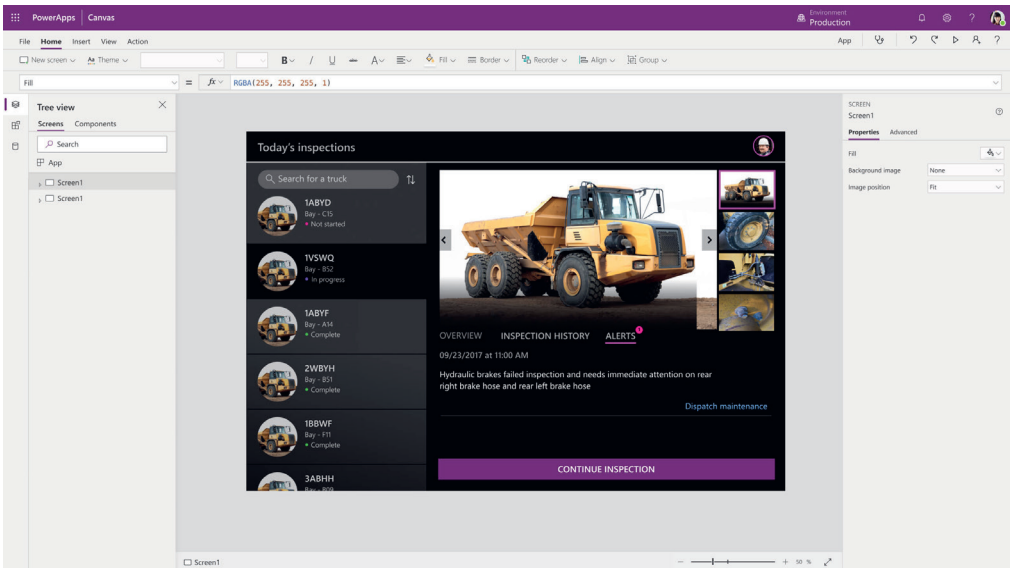
CODEN OHNE TEMPOLIMIT

Um die Anforderungen an die Anwendung vonseiten der Kunden, Nutzer und Stakeholder aufzunehmen, braucht es natürlich nach wie vor Zeit. Doch wenn diese erst mal stehen, und die (agile) Umsetzung beginnt, kann eine LCD-Plattform ihre Stärken ausspielen. In kurzer Zeit können die Teams erste funktionsfähige Prototypen erstellen. Zum Beispiel, um die Ergebnisse eines Workshops gleich umzusetzen und zu beurteilen. Dabei nutzt LCD einen visuellen Ansatz, um Algorithmen zu programmieren und Nutzeroberflächen zu gestalten – frei nach dem Motto: „Ein grafischer Low-Code sagt mehr als 1.000 Zeilen Code-Text." Das macht vieles leichter und damit schneller verständlich.

Dazu kommt, dass die Anbieter von LCD-Plattformen üblicherweise Code-Schnipsel, UI-Templates und ganze Funktionsmodule zur Verfügung stellen. Um nur ein Beispiel aus der Praxis zu nennen: Eine GPS-Funktion lässt sich so innerhalb weniger Minuten in eine mobile App integrieren. Auf diese Weise vereinfacht und beschleunigt LCD die Abstimmung und Entscheidungsfindung in agilen Teams und bei deren Kunden.

LCD ist aber nicht nur schneller. Es liefert nicht selten auch bessere Anwendungen. Das betrifft zunächst einmal die Nutzeroberfläche. Nicht in allen Teams sind professionelle UI-Designer am Werk. Viele kleinere Teams können oder wollen sich eine pixelgenaue Umsetzung der Corporate-Identity nicht leisten – manchmal ist noch nicht mal eine ergonomische Nutzeroberfläche möglich. Mit LCD kann das anders aussehen, da die Entwicklung hier im Regelfall auf Templates basiert. Das bedeutet, dass alle Investitionen in ein gutes Interface allen Apps zugutekommen. Und dass alle Apps ein einheitliches UI bekommen, was wiederum die Akzeptanz durch die Nutzer bei der Einführung einer App fördert.





Oben: Outsystems ist einer der Anbieter von Low-Code-Plattformen. Mobile Apps lassen sich beispielsweise komplett über eine visuelle Oberfläche erstellen.

Links: Auch Branchengrößen wie Microsoft oder Salesforce haben den Trend erkannt und Low-Code-Systeme ins Portfolio aufgenommen. Bei Microsoft nennt sich die Software „Powerapps".

Anders sieht es beim Backend, dem Unterbau, aus. Hier unterscheiden sich die LCD-Systeme. Entscheidend ist, wie durchdacht die zugrunde liegende Softwarearchitektur ist, ob sie den ausführbaren Code erzeugt oder nur interpretiert und wie einfach sie sich integrieren lässt (etwa in Soap, Rest oder SAP). Diese Aspekte wirken sich direkt auf die Performance und Stabilität der per LCD erzeugten Software aus. Eine für alle Hersteller gültige Aussage ist jedoch leider nicht so einfach möglich.

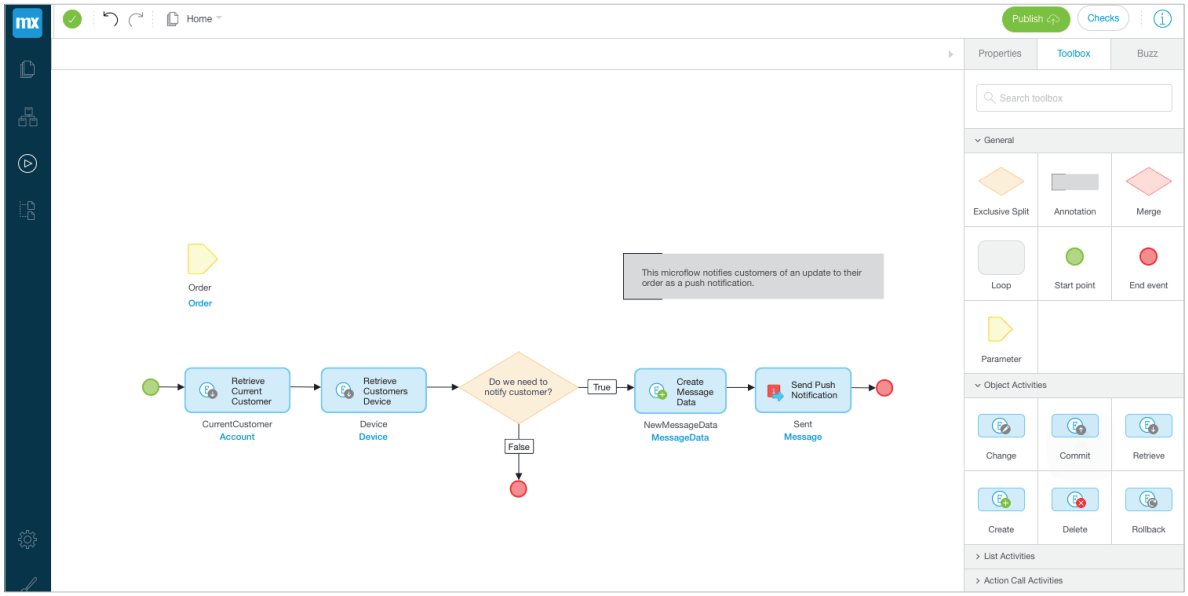
KEINE ANGST VOR EINER SCHATTEN-IT

So verlockend die LCD klingt – es gibt auch einige Kritikpunkte. Um es gleich vorwegzunehmen: Die Befürchtung mancher IT-Manager, dass sich in Verbindung mit LCD mehr Schatten-IT entwickeln könnte, hat sich in der Praxis nicht bestätigt. Schatten-IT meint informationstechnische Prozesse und Systeme, die sich in Organisationen neben der offiziellen Infrastruktur ausgebildet haben und meist einer begrenzten Personengruppe zugänglich sind.

Im Gegenteil: LCD ist eher ein Hebel, um die Schatten-IT in Unternehmen zu vermeiden. Gleichwohl sollten Unternehmen überlegen, inwieweit sich LCD bei ihnen lohnt. Denn aufgrund ihres Komplexitätsgrades lassen sich manche Low-Code-Plattformen nur von Entwicklern oder nach erheblichem Schulungsaufwand nutzen.

Außerdem sollten Unternehmen bei der Auswahl der passenden Low-Code-Plattform darauf achten, dass sie keinen

Screenshots: OutSystems, PowerApps



Mendix Assist unterstützt Anwender bei der Konfiguration von sogenannten „Microflows“. Ein Microflow kann beispielsweise Objekte erstellen, einzelne Seiten anzeigen und sogar Entscheidungen treffen.

Vendor-Lock-in erleben. Manche Plattformen bieten jederzeit Zugriff auf das gesamte Projekt, inklusive der Open-Source-Bibliotheken und des Codes. Bei anderen Anbietern ist das mit zusätzlichen Kosten verbunden. Und wieder andere ermöglichen das gar nicht. Auch beim Lizenzmodell sollten Unternehmen eine klare Kalkulation aufstellen: Welche Kosten fallen für den laufenden Betrieb an? Wie viele Benutzer oder Applikationen braucht es? Gibt es sonstige Runtime- oder Entwicklerlizenzierungen? Und wie kann es die Skalierung der eigenen Apps sicherstellen?

EINE FRAGE DER KULTUR

Neben vielen kleineren Anbietern tummeln sich auch die gängigen großen Anbieter im Markt für Low-Code-Plattformen. Dazu gehören zum Beispiel Outsystems, Mendix, Microsoft Power Apps, Salesforce oder Appian. Jedes System hat seine Stärken und Schwächen. Diese konkret auszuwerten, ist erst nach einer genauen Anforderungsanalyse eines Unternehmens möglich. Faktoren, die unter anderem berücksichtigt werden sollten, sind die Anzahl der zu entwickelnden Apps, der Nutzer und der Entwickler sowie die Anforderungen an die Skalierbarkeit und Sicherheit sowie die Offline-Fähigkeit bei mobilen Apps. Und natürlich sollten IT-Verantwortliche auch die bereits vorhandene IT-Landschaft berücksichtigen.

Bei der Wahl einer LCD-Plattform spielen viele technische Aspekte eine Rolle. Und doch kann die Antwort auf die Frage, welche Plattform die richtige ist, auch ganz einfach sein: Es ist diejenige, die die Mitarbeiter gerne nutzen und die ihre Probleme (Pain-Points) löst. Denn generell gilt, dass Unternehmen immer dann den größten Nutzen (ROI) aus der Einführung einer LCD-Plattform ziehen, wenn möglichst alle Teams möglichst viel – am besten alles – damit entwickeln. Eine Plattform auszuwählen, zu kaufen und zu installieren, reicht dafür aber nicht.

Die Einführung eines solchen Systems erfordert immer auch einen Kulturwandel und einen Veränderungsprozess. Diesen müssen die Führungskräfte gezielt begleiten, wollen sie erfolgreich sein. So sind einige Softwareentwickler vom LCD-Paradigma spontan begeistert und machen sofort mit. Andere haben durchaus Bedenken und den Wunsch, an traditionellen Methoden festzuhalten. Unternehmen sollten sich auf beides gefasst machen und damit umgehen können. Erfahrungsgemäß erkennen aber auch skeptische Entwickler nach einer gewissen Eingewöhnungsphase die Vorteile des LCD.

Eine gute Vorgehensweise ist daher in vielen Fällen: Die Entscheider wählen aus den Topsystemen drei aus. Mit ihnen führen sie einen Proof-of-Concept (PoC) durch. Dabei sollen Mitarbeiter die schwierigste (Teil-)Anforderung aus einem kürzlich beendeten Softwareprojekt innerhalb von drei bis fünf Tagen prototypisch umsetzen. Wer danach die Umsetzungszeit und die Ergebnisse miteinander vergleicht, wird schnell zu einer Entscheidung finden. Wichtig dabei: Devops-Team sollten unbedingt am Auswahlprozess teilnehmen.

FAZIT

Low-Code-Development und Low-Code-Plattformen haben viele Vorteile. Das richtige System auszuwählen und einzuführen, ist sicherlich keine triviale Angelegenheit. Dennoch lohnt es sich, jetzt darüber nachzudenken. Denn LCD beschleunigt und vereinfacht nicht nur komplexe Softwareentwicklungen. Auch sogenannte „Citizen-Developer“, also Anwender oder Mitarbeiter einer Fachabteilung, die keine Softwareentwickler sind, profitieren von der schnelleren Umsetzung und dem geringeren Aufwand. ⊗

PETER SCHRÖDER beschäftigt sich seit 1995 mit Rapid-Application-Development (RAD). In den letzten drei Jahren hat er sich auch mit BPM-, RPA- und Low-Code-Entwicklung auseinandergesetzt – und zwar als Freelancer, Mitarbeiter einer IT-Abteilung sowie eines Softwareberatungshauses.

Screenshot: Mendix