IDG

COMPUTERWOCHE

Ausgabe 2018 – 21-22 22. Mai 2018 Nur im Abonnement erhältlich VON IDG

Deutschland entdeckt KI

Vor allem Fachbereiche wollen vorankommen.

Seite 8

ServiceNow hegt große Pläne

ITSM-Spezialist will seine IT-Nische verlassen.

Seite 22

Wo die Youngster arbeiten wollen

Wunscharbeitgeber Google hat Konkurrenz.

Seite 38

Mit Low Code zu mehr Speed

Geschwindigkeit in der Softwareentwicklung ist Trumpf. Low-Code-Plattformen können helfen.

Seite 14

IDG Business Media GmbH, DataM-Services GmbH, Postfach 9161, 97091 Würzburg

Warum Low Code mehr als ein Hype ist

Entwickler werden händeringend gesucht – zumindest heute. Ob das morgen auch noch so ist, scheint ungewiss. Low-Code-Plattformen und KI werden die IT-Landschaft verändern.

Heinrich Vaske, Editorial Director

Alles schon mal da gewesen, werden Skeptiker sagen, wenn es um Low-Code- und No-Code-Programmierung geht. Sie werden auf die Programmiersprachen der vierten Generation (4GL) oder auf Rapid Application Development (RAD) verweisen – Trends der 80er und 90er Jahre. Hinter der Low-Code-Bewegung steckt aber mehr: Es geht um die Demokratisierung der Softwareentwicklung.

In den Digital Labs und zunehmend auch den Fachbereichen der Unternehmen möchten die Mitarbeiter Innovationen schnell in Prototypen und einfache Produkte umsetzen, agil auf Kundenverhalten reagieren und den Wissensschatz, der in den anschwellenden Datenseen lagert, zeitnah heben können. Hier werden die Fachspezialisten nicht auf ihre IT-Teams warten, wenn sie nicht müssen. Eher werden sie ihre technisch versierten „Citizen Developers“ beauftragen, zumindest die einfacheren Aufgaben selbst zu lösen.

Mit der künstlichen Intelligenz dürfte sich dieser Trend massiv verstärken und beschleunigen. Die Low-Code-Plattformen werden intelligenter, sie gehen den semiprofessionellen Entwicklern mit Simulationen, Handlungsempfehlungen und Vorschlägen für die automatisierte Vervollständigung von Anwendungen zur Hand.

Der Trend ist zweifellos disruptiv – insbesondere für die zahllosen Programmierer und Consultants, die heute von sechsmonatigen Softwareentstehungszyklen leben und eigentlich kein großes Interesse haben, daran Grundsätzliches zu ändern. Aber sie werden sich dem Druck, der aus den Digitallabors und den innovativen Fachbereichen kommt, auf Dauer nicht widersetzen können.

Herzlich,
Ihr

Heinrich Vaske, Editorial Director

Low-Code-Plattformen

Vier Erfolgsbeispiele, wie Anwenderunternehmen Low-Code-Frameworks für die Programmierung nutzen:
computerwoche.de/a/3544689

Titelfoto: Vector Tradition/Shutterstock

Ohne Low-Code-Programmierung sind Speed und Agilität unrealistisch

Low-Code-Plattformen ermöglichen die interaktive Entwicklung datenbankbasierter Fachanwendungen nahezu ohne Programmierung. Ihre Eignung für Kleinanwendungen, aber auch für große, kritische IT-Projekte ist nachgewiesen. Praktiker wissen auch, dass die Versprechen hinsichtlich Schnelligkeit und Flexibilität, Anwenderzufriedenheit und Projektsicherheit in der Regel erfüllt werden. Dennoch gibt es in vielen Unternehmen Vorbehalte.

Von Karsten Noack, Gründer und CEO der Scopeland Technology GmbH

Wie nimmt die digitale Transformation Geschwindigkeit auf, ohne dass langwierige Programmieraufgaben eine bremsende Wirkung entfalten? Durch interaktives Zusammenklicken vorgefertigter Softwarebausteine. Dadurch können Anwender und Power-User der Fachabteilungen besser in die Entwicklung ihrer Softwarelösungen einbezogen werden.

So werden eine wirklich agile Softwareentwicklung und ein gemeinsames Erarbeiten optimaler Lösungen direkt am „lebenden Objekt" möglich: durch Konzeption und Umsetzung in einem Zuge, mit modernen Methoden des Design Thinking über alle Projektphasen hinweg.

Hinter Low-Code-Programmierung steckt nicht nur die Idee, Kosten zu sparen und schneller zu Ergebnissen zu kommen. Es ist ein völlig anderer Ansatz, Softwarelösungen konzeptionell zu erarbeiten. Es geht auch darum, nicht am realen Bedarf vorbei zu entwickeln und den gesamten Prozess der Konzeption, Entwicklung und Pflege in einem Maße zu flexibilisieren, wie man es sich bislang kaum vorstellen konnte. Das ist der Grund, warum Forrester Research den Low-Code-Plattformen eine Schlüsselrolle bei der Digitalisierung aller Geschäftsprozesse zuschreibt und warum künftig kein Unternehmen mehr umhinkönnen wird, für Low-Code-Plattformen eine Rolle in seiner IT-Strategie vorzusehen.

Software entwickeln, ohne zu programmieren

Schon seit vielen Jahren arbeiten Unternehmer an immer neuen Ideen, um den langwierigen Prozess der Softwareentwicklung deutlich zu verkürzen. Dabei kommen unterschiedliche Lösungsansätze zum Tragen. Sie reichen von den 4GL-Sprachen der 90er Jahre über Computer-aided Software Engineering (CASE) bis hin zur sogenannten modellgetriebenen Softwareentwicklung. So ziemlich alles wurde versucht, um irgendwie Programmcode automatisch generieren zu können.

Der Durchbruch aber kommt aus einer völlig anderen Ecke: von Plattformen, die die gesamte theoretisch benötigte Funktionalität bereits vorgefertigt parat halten und die in sich intelligent genug aufgebaut sind, damit Objekte und Features weitgehend wie Legosteine zusammengestöpselt werden können, ohne dass sich die Verantwortlichen im Detail um technische Belange kümmern müssen.

Komplette Neuentwicklungen sind so nach wenigen Wochen statt nach vielen Monaten oder Jahren produktiv verfügbar, Programmerweiterungen in Tagen statt in Monaten. Das ist ein Traum für die Verantwortlichen der Fachabteilungen und eine große Herausforderung für die ITler, die für Sicherheit und Stabilität der gesamten Infrastruktur sorgen müssen.

Bedenken sind verständlich, aber die Vorteile von Low-Code-Ansätzen im Vergleich zur klassischen Softwareentwicklung sind so gravierend, dass man nicht umhinkommt, sich damit zu beschäftigen. Die etablierten Low-Code-Plattformen genügen heute bereits den höchsten Sicherheitsanforderungen, vielleicht sind die Ergebnisse sogar besser als manche handgeschriebene Software. Der Standardsoftware-Ansatz ermöglicht Datensicherheit, Barrierefreiheit und vieles mehr out of the Box.

Was bedeutet das „Low" in Low Code?

Mit dem von Forrester Research 2014 aufgebrachten Begriff Low Code ist gemeint, dass bereits IT-Basiswissen ausreicht, um sich einigermaßen sicher darin zu bewegen. Auch wenn gelernte Informatiker mehr herausholen können, werden auch Nichtinformatiker den Code lesen, verstehen und wahrscheinlich auch verändern können. Die Objektstrukturen wurden bereits automatisch von der Plattform erzeugt, Programmschleifen sind über alle selektierten Datensätze von Haus aus vorhanden, und alle gängigen Rechen- und Konvertierfunktionen liegen direkt zur Verwendung bereit. Damit kann sich das bisschen Programmcode, das man hin und wieder ergänzen muss, auf vergleichsweise einfache Funktions- und Variablendefinitionen mit simplen Aktionsfolgen und Verzweigungen beschränken.

Die meisten Low-Code-Produkte zielen auf die Schnittstelle zwischen den zentralen IT-Abteilungen der Unternehmen und den Anwendern ab. Manche Tools sind eher für Endanwender, andere für IT-Profis optimiert. Die wichtigste Zielgruppe der Low-Code-Plattformen ist aber wohl die, die genau dazwischenliegt: also IT-kundige Mitarbeiter der Fachbereiche sowie IT-Spezialisten, die sich über viele Jahre hinweg tief in die Themen der Fachabteilungen eingearbeitet haben. Für diese Gruppe von fach- und zugleich IT-kundigen Mitarbeitern steht der neu erfundene Begriff „Citizen Programmer". Das ist jemand, der in der Fachabteilung zu Hause ist, zugleich aber auch im Bedarfsfall etwas programmieren kann.

Damit kommt die Low-Code-Programmierung dem Trend entgegen, dass sich die Grenzen zwischen IT und Fachabteilung ohnehin mehr und mehr aufheben. In nahezu allen größeren Unternehmen hat sich längst eine Schatten-IT entwickelt, in der sich die Fachabteilungen um ihre eigenen Anwendungen selbst kümmern. Dabei werden sie mehr oder weniger intensiv von der Zentral-IT gestützt. In Wahrheit werden sie von Power-Usern, fachlich orientierten Informatikern sowie „Beinahe-Informatikern" getrieben, eben den Citizen Programmern.

Mit oder gegen die Schatten-IT?

In manchen Unternehmen bemüht man sich, die Schatten-IT zurückzudrängen, aber das ist wohl ein aussichtloses Vorhaben. Oft ist die Eigen-IT der Fachbereiche heute wichtiger denn je, kommen doch die Bedarfe und Ideen zur Digitalisierung meist aus den Fachbereichen selbst, nicht aus der Zentral-IT. Zentralistische Ansätze, etwa nach dem Prinzip „Wir machen alles mit SAP", sind für zentrale, klar geregelte und gut standardisierbare Aufgaben zweifelsfrei richtig. Dies gilt aber nicht für die vielen kleinen Besonderheiten der Fachbereiche, über die sich die Unternehmen ja auch vom Wettbewerb abgrenzen wollen. Zudem sind zentrale IT-Aufgaben durch die ausgereiften ERP-Systeme bereits so weit durchdigitalisiert, dass sich die Frage stellt, was man außer den letzten verbliebenen Lücken – der elektronischen Rechnung etwa – noch weiter digitalisieren kann. Anders sieht es in den Fachbereichen aus. In den Verästelungen des täglichen Geschäfts steht die Digitalisierung noch ganz am Anfang. Die Mitarbeiter dort wissen auch oft, wie sie ihre Prozesse weiter optimieren und automatisieren können.

Die Zentral-IT ist meistens damit überfordert, die unzähligen Ideen aus den Fachbereichen umzusetzen. Zudem verzweifeln die Fachverantwortlichen an der Langwierigkeit der klassischen, ERP-nahen Softwareentwicklung und den damit verbundenen Entscheidungsprozessen. So kommt es, dass viele Potenziale der Digitalisierung nicht genutzt werden. Kein Wunder also, wenn sich die Fachbereiche großer Unternehmen lieber selbst helfen. Die Zentral-IT sollte diese Entwicklung als Chance sehen und sie selbst steuern und vorantreiben. Low-Code-Plattformen, die von der Zentral-IT betrieben werden, können helfen, Licht in die Schatten-IT zu bringen.

Wie funktionieren Low-Code-Plattformen?

Im Detail folgen die einzelnen Produkte unterschiedlichen Konzepten. Allen gemeinsam ist aber, dass man eine Cockpit-ähnliche Entwicklungsplattform hat, mit der Browser-Anwendungen oder Apps interaktiv zusammenge-

Mehr über Low-Code-Programmierung und Entwicklungstrends:
Was Low-Code-Programmierung bringt
www.cowo.de/a/3544834
Softwareentwicklung meets Lego
www.cowo.de/a/3544689
Was steckt hinter den Buzzwords Low Code und Rapid Application Development
www.cowo.de/a/3332221

Foto: Vector Tradition/Shutterstock

Wie reif sind Low-Code-Plattformen?

Sind Low-Code-Plattformen schon erwachsen oder stecken sie noch in den Kinderschuhen? Die Antwort lautet: sowohl als auch. Die Vorreiter der Low-Code-Idee sind schon viel länger auf dem Markt, als der Begriff existiert. Einige sind schon über zehn Jahre erfolgreich im Echteinsatz und gehen auf zum Teil noch viel länger zurückliegende Entwicklungs- und Pilotierungsphasen zurück.

In den USA gibt es bereits seit geraumer Zeit sehr viel Erfahrung mit von Endanwendern selbst zusammengeklickten Kleinstanwendungen, die dann in der Regel auch nur in der Cloud, also völlig losgelöst von der Zentral-IT, betrieben werden oder die sich um Cloud-basierte CRM-Lösungen herumranken.

In Deutschland setzen die frühen Anbieter eher auf die Kooperation zwischen zentraler und dezentraler IT und können auf zum Teil beeindruckende, komplett mit Low-Code-Technologien umgesetzte Großprojekte als Referenz verweisen. Solche Lösungen werden dann zumeist on Premise in den Rechenzentren der Kunden installiert und dafür auch als generierter Programmcode frei von Runtime-Komponenten ausgeliefert.

Die Eignung der Low-Code-Technik, sowohl für Klein- und Kleinstanwendungen als auch für große und kritische IT-Projekte, ist längst nachgewiesen. Und auch der Nachweis, dass die großen Versprechen hinsichtlich Schnelligkeit und Flexibilität, Anwenderzufriedenheit und Projektsicherheit tatsächlich erfüllt werden, ist erbracht.

Die wirtschaftliche Vernunft spricht derart massiv für den flächendeckenden Einsatz von Low-Code-Plattformen für typische dezentral entwickelte Fachanwendungen, dass es verwundert, wie viel Vorbehalte dem immer noch im Wege stehen.

Auf der anderen Seite aber schießen seit der Namensgebung für die Technologie durch Forrester Research und seit den damit verbundenen atemberaubenden Umsatzprognosen Startups mit immer neuen Low-Code-Produkten wie Pilze aus dem Boden. Dutzende neue Anbieter, zumeist mit viel Venture Capital ausgestattet, beleben die Szene auf beeindruckende Weise. Welche davon sich am Markt behaupten und welche neu aufgebrachten Ideen die Technologie insgesamt weiter vorantreiben werden, bleibt abzuwarten. Die Technologie als solche ist ausgereift, aber durchaus noch offen für weitere Innovationen.

Ein besonderer Vorteil ergibt sich zudem durch die vielen Features, die mit den Plattformen einfach mitgeliefert werden. Ein Beispiel dafür ist die Ausweitung des Low-Code-Ansatzes auf den Umgang mit Geodaten.

Das ist sehr sinnvoll, weil heutzutage nahezu alles einen Geodatenbezug hat. Dergleichen mit händischer Programmierung im Einzelprojekt umzusetzen, erscheint aus Kostengründen fast unmöglich. Deshalb sind Low-Code-Produkte ein besonders aussichtsreicher Weg zu bezahlbaren, kombinierten Sach- und Geodaten-Anwendungen, ohne hochspezialisierte Experten, die in beiden Welten zu Hause sind.

Man kann sagen, dass die vielen Vorteile, die sich zusätzlich noch aus dem Standardsoftwareansatz ableiten lassen, die wenigen verbliebenen funktionalen Einschränkungen mehr als wettmachen.

klickt werden. Das Ergebnis liegt dann in einer strukturierten und deklarativen Form vor, zum Beispiel als Metadaten in einer Metadatenbank, so dass die entsprechenden Programmteile auch weiterentwickelt werden können.

Einige Low-Code-Plattformen sind streng an den Einsatz und die Entwicklung in der Cloud gebunden, während andere Tools beziehungsweise ihre Runtime-Systeme auch on Premise installiert werden können. Dementsprechend arbeiten einige interpretativ, benötigen also zur Laufzeit ein auf dem Server oder in der Cloud installiertes Laufzeitsystem. Andere Tools sind eher eine Weiterentwicklung der modellgetriebenen Softwareentwicklung und generieren ausführbaren Code, der dann im günstigsten Fall final ist und – um die Flexibilität nicht zu verlieren – nicht manuell weiter ausprogrammiert werden sollte. Teilweise wird auch auf eine Kombination beider Methoden gesetzt, um hochinteraktiv am echten „lebenden Objekt" entwickeln und dennoch abschließend reinen Programmcode ausliefern zu können. Vermutlich ist das, zumindest für den Einsatz der fertigen Anwendungen im eigenen Rechenzentrum, der erfolgversprechendste Weg.

Nahezu allen Methoden gemeinsam ist, dass im interaktiven Entwicklermodus nicht nur die Programme selbst, sondern auch das zugrunde liegende Datenmodell quasi in einem Zuge mitentwickelt werden. Ebenso interaktiv werden Datenbankabfragen und -sichten einfach zusammengeklickt und dann jeweils geeignete Visualisierungen ausgewählt, zum Beispiel Formular- oder Tabellenansicht beziehungsweise Map-, Tree- oder Chart-Controls. Die Anbindung der Controls an die Datensichten erfolgt dabei weitgehend eigenintelligent.

Programmoberflächen werden aufgebaut, indem man sich die jeweiligen Datensichten in der gewählten Visualisierungsform einfach auf den Bildschirm zieht und miteinander verlinkt. Zum Feintuning des Layouts, und auch für solche Herausforderungen wie das Responsive Design, sowie für das Einbringen geeigneter zusätzlicher Bedienelemente stehen dann Form Designer und andere herstellerspezifische Designwerkzeuge bereit. Im Zuge dieser Konfigurationen entsteht auch der Grundstock der benötigten Anwendungslogik automatisch, zumindest soweit sie sich bereits aus dem evolutionär entwickelten Datenmodell und den diversen Eigenschaftsklassifikationen, die sich im Konfigurationsprozess ansammeln, ableiten lässt.

Grundmethodik hilft schon weiter

Alle Anbieter verfolgen damit eine relativ einheitliche Grundmethodik, und man kommt damit erstaunlich weit. Die meisten Anwendungen sind tatsächlich fast fertig, ehe man das erste Mal an die Stelle gelangt, wo ein Gap zwischen der automatisch mitgenerierten Integritäts- und Anwendungslogik und den realen Anforderungen an die Geschäftslogik aufkommt.

Diese Lücke versuchen die Hersteller mit unterschiedlichen Ansätzen zu füllen. Einige Produkte setzen massiv auf das Prinzip regelbasierter Systeme, indem alle weiterführende Programmlogik in Form von bedingten Regeln definiert wird, beispielsweise in der Art von „Wenn Feld A größer als null und Feld B kleiner als 100, dann muss Feld C größer sein als das Doppelte von Feld D", allerdings in herstellerspezifischen kompakten Notationen, die teilweise an Excel-Formeln erinnern.

Mit einem Mix aus unterschiedlichen Regeltypen wie zum Beispiel Plausibilitäts-, Berechnungs-, Sichtbarkeits- und Zugriffsregeln lässt sich auf rein deklarative Weise komplexe Anwendungslogik komplett beschreiben. Was zur Laufzeit mit den Regeln jeweils zu tun ist, das weiß die Low-Code-Plattform dann von selbst, so dass man sich auch hierbei auf die Inhalte konzentrieren kann und um die technische Umsetzung nicht kümmern muss. Der Charme solcher regelbasierter Systeme ist die Einfachheit, mit der sich das Verhalten und die Eigenschaften von Daten beschreiben lassen.

Andere Hersteller setzen verstärkt auf grafische Werkzeuge, um Programmabläufe bildhaft darzustellen, auf eingebettete Workflow-Engines oder andere Mechanismen. Vermutlich werden hier in der nächsten Zeit noch etliche spannende Konzepte aufkommen, um auch den letzten noch verbliebenen Herausforderungen so gut wie möglich zu begegnen.

Und schließlich gibt es noch einen Ansatz, der nahezu immer funktioniert: Er besteht darin, für besondere Anforderungen kleine handgeschriebene Programmbausteine ergänzen zu können, und zwar als niedrigschwelligen Code. Dieser greift direkt auf die automatisch generierten Objekte zu und kann mit diesen all das tun, was das interaktive Klickersystem nicht von Haus aus abdeckt. Einige wenige Produkte verzichten auf diesen Weg, und nennen sich deshalb No-Code-Plattformen, allerdings bleibt zu bezweifeln, ob sich dieser hehre Anspruch auf Dauer aufrechterhalten lässt. Die Praxis zeigt, dass man doch immer mal wieder an den Punkt kommt, wo für bestimmte Spezialalgorithmen kleine eingebettete Low-Code-Scripts erforderlich sind.

Möglichkeiten sind endlich

So oder so, letztlich kommt man nicht umhin, zuzugeben, dass der Low-Code-Ansatz in seinen Möglichkeiten endlich ist, gewissermaßen ein 98-Prozent-Ansatz. Während man theoretisch alles nur Denkbare programmieren kann, wird man in der Praxis an Grenzen stoßen. Die Erfahrung zeigt aber auch, dass man bei der Entwicklung kundenspezifischer Softwarelösungen sowieso immer an Grenzen stößt. Und oft ist es weit weniger schmerzlich, auf das eine oder andere Luxus-Feature technologiebedingt zu verzichten, als wenn ein theoretisch perfektes Programm niemals fertig wird und Termine und Budgetgrenzen weitaus härtere Restriktionen mit sich bringen.

Und ganz viel zusätzlich out of the Box

Low-Code-Plattformen haben noch einen weiteren großen Vorteil. Da sie auf einem Fertigsoftware-Ansatz beruhen, können sie eine große Menge an Funktionalität standardmäßig mitbringen, deren Entwicklung im Einzelfall zu teuer wäre: Features, die Sie nur gelegentlich benötigen, dann aber sehr zu schätzen wissen werden. Beispielsweise bringen viele Produkte gleich ihre eigene Test-, Freigabe- und Deployment-Umgebung mit. Andere wiederum punkten mit integrierten Reengineering-Werkzeugen oder mit einer mit einem Mausklick zuschaltbaren Vollhistorisierung aller Änderungen in der Datenbank, mit eingebauten intelligenten Ad-hoc-Reporting-Werkzeugen, eingebetteten Analysefunktionen oder vorimplementierten Web-Service-Schnittstellen zu gängigen ERP-Produkten. Gängige Erweiterungen einiger Low-Code-Plattformen sind auch einfach bedienbare Schnittstellengeneratoren, die direkt auf die Metabeschreibungen der Plattform abgestimmt sind, sowie leistungsfähige Systeme zur Generierung etwa von Word-, Excel- und Powerpoint-Dokumenten, HTML-Dateien oder E-Mails. *(hv)*