

MAGAZIN FÜR DIE ENTERPRISE IT

LOW-CODE PLATTFORMEN

Phasenagiles Vorgehen
in IT-Projekten

CONTAINER AN DIE MACHT

Docker und Kubernetes
als Tandem der Zukunft

CUSTOMER EXPERIENCE

Warum Künstliche
Intelligenz so wichtig ist

ENTERPRISE-RESOURCE-PLANNING

FORTSCHRITT DURCH INNOVATION

Martin Hinrichs, ams.Solution AG



Megatrend Konnektivität
ab Seite 14



Edge Computing
in der Praxis
ab Seite 50

www.it-daily.net

30



54



INHALT

IT MANAGEMENT



10 Coverstory – ERP: Fortschritt durch Innovation
Digitalisierung entlang der Wertschöpfungskette.

12 Digitalisierung mit System
Anspruch und Wirklichkeit.



14 Megatrend Konnektivität
Warum der Erfolg bei IIoT
von der Datenintegration abhängt.

16 Die Autonomie der Systeme
Integrated Industry – Industrial Intelligence.

18 Die Migrationsfrage
SAP-Lizenzmanagement.

20 Hallo Wach!
Wie gelingt die Digitale Transformation?

24 Zeit für die wirklich wichtigen Dinge
Gut dokumentiert dank Digitalisierung.

26 Künstliche Intelligenz trifft ECM
Chancen und Grenzen von KI
im Informationsmanagement.

30 Multifunktionssysteme
Effizient, sicher und schonend drucken.

34 Der IT-Markt im Wandel
Konsolidierung und Digitalisierung
treiben M&A im IT-Sektor.



36 Low-Code-Plattformen
Wie man Softwareprojekte „in time“ realisieren kann.

38 MITO-Konfigurationsmanagement
Der Weg zur ganzheitlichen Businessmodell-Entwicklung.

42 Modellbasiertes Prozessmanagement
Neue Produkte zu entwickeln war noch nie einfach.

46 AIOps im Einsatz
Warum wir künstliche Intelligenz im IT-Betrieb brauchen.

PHASENAGILES VORGEHENS LOW-CODE-PLATTFORMEN

Low-Code-Plattformen ermöglichen hochgradige Agilität. Eine direkte Zusammenarbeit von Softwareentwicklern und Anwendern stellt sicher, dass zukünftig nicht mehr am Bedarf vorbei entwickelt wird. Neue Erkenntnisse können jeder Zeit direkt in den Entwicklungsprozess mit einfließen. Was bedeutet das für das Vorgehen in IT-Projekten? Darf oder muss man Vorgehensmodelle neu denken? Dazu sollte man grundlegende Prinzipien beachten.

Design-Thinking-Workshops

Da es mit modernen Low-Code-Plattformen heutzutage möglich ist, direkt am Bildschirm Änderungen an der Anwendung vorzunehmen oder diese sogar direkt zu erstellen, ist Scrum als Vorgehensmodell zu starr und zu langsam. Sinnvoller hingegen sind regelmäßige Design-Thinking-Workshops über den gesamten Projektverlauf hinweg. Design Thinking steht in diesem Zusammenhang für interdisziplinäre Teams, bildliches Arbeiten mit Skizzen und Screenshots, Offenheit für neue Ideen und die Erlaubnis, Fehler zu machen – je früher desto besser. Korrekturen sind umso kostengünstiger, je früher sie vorgenommen werden. Regelmäßig stattfindende Workshops erlauben deren Früherkennung.

An einem Tisch sitzen Vertreter der Auftraggeber- und Auftragnehmer-Seite: Das bewusst gemischte Team diskutiert den aktuellen Entwicklungsstand, konkretisiert, verändert und verfeinert die mit der Low-Code-Software erstellte Anwendung, das Datenmodell oder die Konzepte. Dies erfolgt jeweils iterativ, entsprechend der sich in der jeweiligen Projektphase ergebenden Arbeiten. Der Projektleiter übernimmt dabei die Rolle des Koordinators und verantwortet die Zielgerichtetheit der Arbeitstreffen.

Strukturierung eines Projekts

Die entscheidende Frage ist, wie hoch-

gradige Agilität auf der einen Seite und die Einhaltung des vereinbarten Festpreises auf der anderen Seite zusammenpassen? Ist das Korsett eng, muss man sehr schnell und effizient die Bedürfnisse des anderen verstehen und gemeinsam und intensiv zielgerichtet arbeiten. Eine sinnvolle Aufteilung des Projekts in Phasen erweist sich hier als praktikabel.

Das Softwareprojekt wird nicht inhaltlich in einzelne Phasen unterteilt, sondern ganzheitlich betrachtet. Der Inhalt der jeweiligen Arbeit bestimmt die Phase. Jede Phase ist in sich hochgradig agil, wobei wichtig ist, dass man zum Abschluss einer Phase kommt und auf der so geschaffenen Grundlage weiterarbeiten kann. Begonnen wird mit dem „Sich-Hinendenken“ – dabei stehen konzeptionelle Arbeiten im Vordergrund, gefolgt von der Datenbasis als Grundlage jeder Datenbankanwendung. In diesen frühen Phasen wird der Grundstein gelegt. Anschließend wird der Programmrahmen definiert, und die Fachmodule kümmern sich danach um den eigentlichen Inhalt der Software. Die Finalisierungsphase stellt den Abschluss des Projekts dar. Eine Qualitätssicherung begleitet alle Phasen, denn Konzepte sind genauso überprüfbar wie ein angereichertes Datenmodell und eine funktionierende Anwendung selbst.

Die Projektphasen

1. Phase: Erst denken, dann handeln

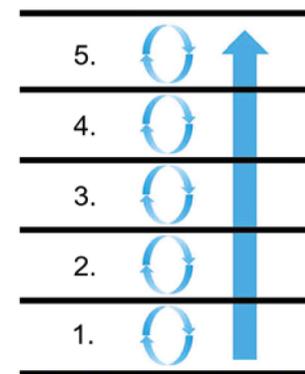
Um die konzeptionelle Phase effizient zu halten, muss man kein Pflichtenheft schreiben, aber Themen spezifizieren und konzeptionieren, die je nach Plattform wirklich zu klären sind und Einfluss auf den Projektverlauf haben. Man braucht auch nicht alles bis ins Kleinste ausfeilen, aber kritische Funktionalitäten sind zu spezifizieren. Weiterhin werden grundlegende Entscheidungen, die beispielsweise den Umgang mit

unterschiedlichen Nutzergruppen betreffen oder infrastrukturabhängig sind, getroffen und länger laufende Tasks, wie besondere Programmierarbeiten, die eventuell nicht durch die Low-Code-Plattform abgedeckt werden, identifiziert.

Am Ende steht außerdem ein Projektplan, der, wenn man sich im Auftraggeber-Auftragnehmer-Verhältnis befindet, von beiden Seiten abgenommen ist. Auch Umfeld-, Stakeholder- und Risikoanalyse werden durchgeführt.

2. Phase: Die Datenbasis

Für Datenbankanwendungen ist es außerdem sehr sinnvoll, das Datenmodell in seinen Grundstrukturen frühzeitig zu entwickeln und verbindlich festzulegen, um Kostenexplosionen in späteren Projektphasen zu vermeiden. Auch dies erfolgt in gemischten Teams. Das Beschreiben von Business Rules gehört dabei genauso dazu wie die Anreicherung



Phasenagiles Vorgehensmodell.

MODELL UND

WIE MAN SOFTWAREPROJEKTE
„IN TIME“ REALISIEREN KANN.

des Datenmodells um genaue Erläuterungen und Funktionalitäten der einzelnen Tabellen und Felder. Gearbeitet wird mit den realitätsnahen Daten der Kunden oder anonymisiert, um so wirklichkeitsnäher und unter Last zu entwickeln.

3. Phase: Der Programmrahmen

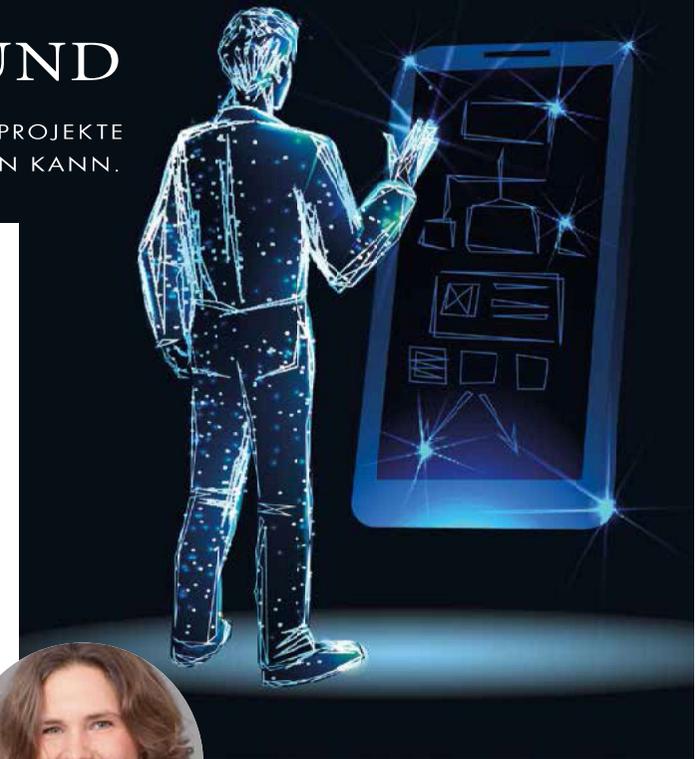
Der Programmrahmen bildet das Gerüst der Anwendung, den technischen Rahmen sowie Grund- und Basisfunktionen ab. Dabei werden das Rechte- und Rollenkonzept inklusive UX-Richtlinien umgesetzt, und die Menüstruktur im Groben entsteht genauso wie standardisierte Schnittstellen und zentrale Funktionalitäten.

4. Phase: Die Fachmodule

Mit einem gut entwickelten Datenmodell inklusive Business Rules und einem Programmrahmen kann man nun die konkreten fachlichen Anforderungen umsetzen. Diese Phase im Projekt ist die intensivste, möglicherweise wird man die regelmäßig stattfindenden Workshops parallelisieren. Der Projektleiter spielt eine moderierende Rolle und achtet darauf, dass die Teamzusammensetzung stimmig ist und dass nicht von den zuvor definierten Themen abgewichen wird. Anwender bringen ihre speziellen Anforderungen mit ein, es wird an der Tafel entworfen, an den ersten Projektständen diskutiert, variiert, verfeinert und iteriert. Die Akzeptanz für die spätere Softwareanwendung ist umso höher, je mehr das Projektteam auf die speziellen Bedürfnisse der Anwender einght.

5. Phase: Die Finalisierung

Die letzte Projektphase zeichnet sich durch den letzten Feinschliff, ausführliche Tests und kleinste Anpassungen aus. Eine letzte Altdatenübernahme



„
DIE ENTSCHEIDENDE
FRAGE IST, WIE HOCH-
GRADIGE AGILITÄT AUF
DER EINEN SEITE UND
DIE EINHALTUNG DES
VEREINBARTEN FEST-
PREISES AUF DER
ANDEREN SEITE
ZUSAMMENPASSEN?“

Claudia Peißert, Leiterin
Projektmanagement / Production Services,
Scopeland Technology GmbH
www.scopeland.de

und Unterstützung bei der Inbetriebnahme gehören ebenso, falls gefordert, dazu, sodass einer Abnahme nichts mehr im Wege steht. Auf Schulungen kann meist verzichtet werden, da die Anwender die Software aktiv mitgestaltet haben. Dadurch hat man es von Anfang an mit einer erhöhten Akzeptanz bei den Personen zu tun, die später mit der Anwendung arbeiten sollen.

Fazit

Das phasenagile Vorgehen in IT-Projekten mit Low-Code-Plattformen wie SCOPELAND ermöglicht eine hochgradige Agilität innerhalb der Phasen und stellt so auf der einen Seite sicher, dass der Bedarf getroffen wird, und auf der anderen Seite verhindert die Zerteilung eines agilen Projekts in wohl definierte Phasen eine Kostenexplosion und zeitliche Verzögerungen.

Claudia Peißert