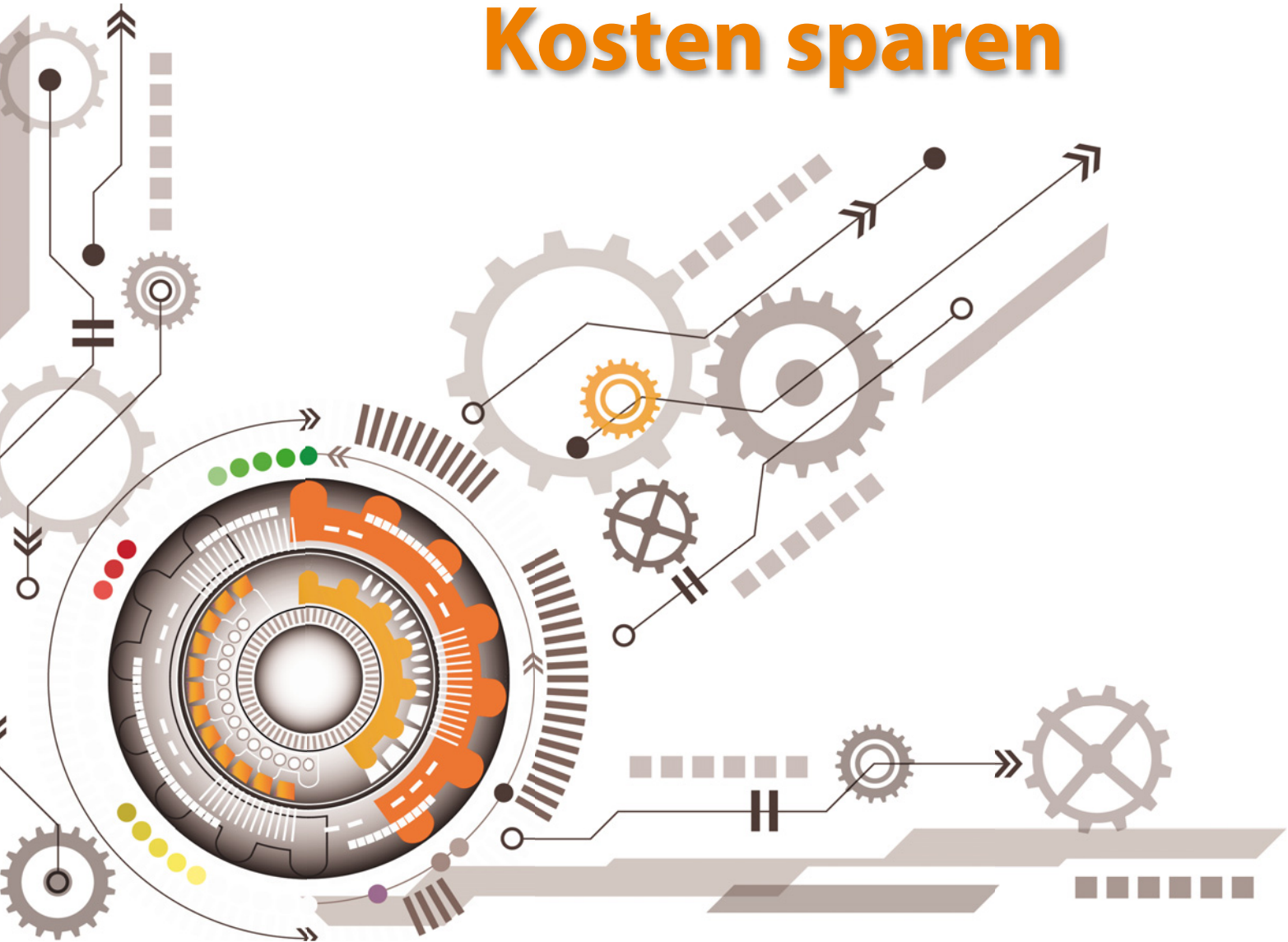


Total Cost of User Ownership

Investition tätigen – Kosten sparen



Eine User Owned Application (UOA) gibt den Fachanwendern die nötige Flexibilität, um Änderungen schnell umzusetzen. Ihre Konzeption und Entwicklung ist aufwändiger als die klassischer Unternehmensanwendungen und erfordert einen anderen Analysefokus.

Die Wettbewerbsfähigkeit hängt heute mehr denn je davon ab, wie schnell ein Unternehmen sich an veränderte Marktbedingungen anpassen kann. Zusätzlich zu den bekannten Wettbewerbern drängen völlig neue Player aus anderen Branchen in den

Markt und zwingen die etablierten Unternehmen dazu, ihre bestehenden Produkte und Prozesse laufend zu verfeinern und neue zu entwickeln. Gleichzeitig nimmt in Branchen wie der Finanzdienstleistung der Umfang der regulatorischen Anforderungen stetig zu. Die damit verbundenen Umset-

zungszeiträume werden im Gegenzug immer kürzer. Für die Zukunft der Unternehmen wird die Strecke zwischen dem Identifizieren eines Änderungsbedarfs und der Sichtbarkeit der Änderung am Markt zum Schlüssel: Wie lange benötigt die Organisation für die Änderung (Änderungsgeschwindigkeit).

keit), und wie viel hohe Kosten (Änderungskosten) entstehen dabei?

User Owned Applications

Die klassische Art, IT-gestützte Produkte und Prozesse in einem Unternehmen zu verändern, läuft in der Regel wie folgt ab: Eine Fachabteilung identifiziert und beschreibt den Bedarf einer Änderung. Die IT-Abteilung leitet aus der Beschreibung ein technisches Konzept ab, implementiert die Änderung und nimmt sie in Betrieb. Die fachliche Änderung wird von der Identifikation bis zur Sichtbarkeit am Markt mehrfach „übersetzt“. Diese Übersetzungen verringern entscheidend die Änderungsgeschwindigkeit. Umgekehrt lässt sich die Geschwindigkeit erhöhen, indem die Übersetzungen auf ein Minimum reduziert werden. Dazu müssen die Änderung durch diejenigen durchgeführt werden können, die den Bedarf identifiziert haben – die Fachanwender. Damit das möglich wird, gibt es die folgenden Rahmenbedingungen zu beachten:

- Da die wenigsten Fachanwender über entsprechende Programmierkenntnisse verfügen, darf für die Umsetzung der Änderung keine Anpassung des Anwendungs-Sourcecodes notwendig sein.
- Die Anwendung muss möglichst robust gegen Fehler sein, die der Anwender bei der Anpassung potenziell begeht. Diese Robustheit besteht aus zwei Aspekten:
 - Fehler vermeiden: Die Anwendung enthält Vorkehrungen, damit vorhersehbare Fehler nicht entstehen können (Beispiel: USB-Stecker lässt sich nur richtig herum einstecken).
 - Fehler aufdecken: Die Anwendung erlaubt dem Benutzer, nicht vorhersehbare Fehler (meist inhaltlicher Natur) möglichst schnell zu entdecken (bspw. durch Voransichten).
- Änderungen dieser Art müssen ebenso wie Änderungen am Sourcecode nachvollziehbar („Wer hat was, wann geändert?“) und reversibel sein.

Diese Anwendung wird nicht mehr von der Unternehmens-IT, sondern von den Fachanwendern selbst angepasst. Wir wollen im Folgenden von User Owned

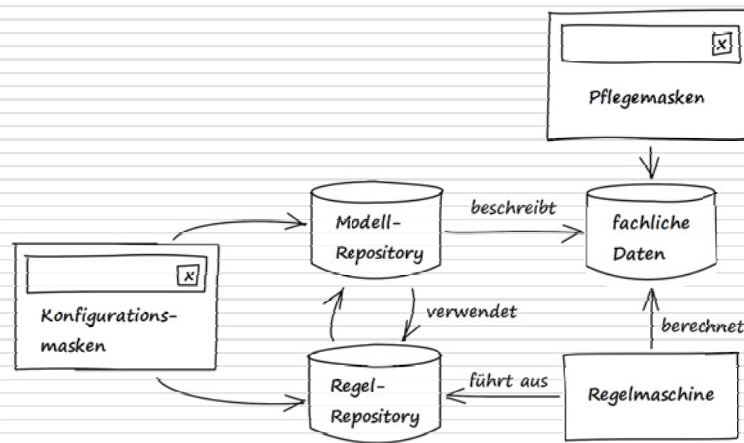


Bild 1: Aufbau einer User Owned Application.

Applications sprechen. Der Begriff ist umso treffender, wenn man bedenkt, dass nicht nur Spezifikation und Umsetzung von Änderungen, sondern auch Test, Abnahme und Deployment durch die Fachanwender durchgeführt werden. Die IT-Abteilung schafft dazu die passenden Rahmenbedingungen.

Aufbau einer User Owned Application

Beobachtet man die Änderungen an einer klassischen Unternehmensanwendung über einen längeren Zeitraum,



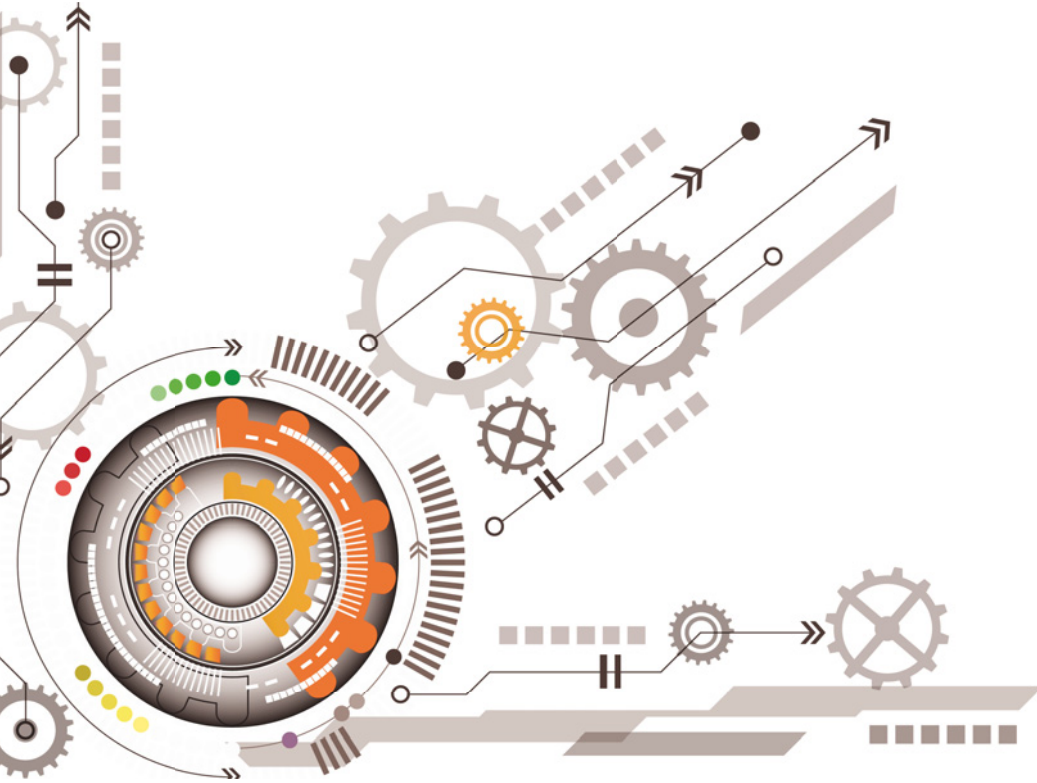
Moderne Unternehmen müssen für eine hohe Änderungsgeschwindigkeit sorgen. Genau wie zu wenig Flexibilität führt auch zu wenig Fokussierung zu einer niedrigen Änderungsgeschwindigkeit.

Konstantin Diener,
Leading Consultant, Cofinpro AG

bilden sich verschiedene Bereiche heraus. Das Datenmodell und die Geschäftslogik ändern sich regelmäßig, Infrastrukturcode allerdings nach einer anfänglichen Stabilisierungsphase nur noch sehr selten. In einer UOA sind diese beiden Bereiche technisch voneinander getrennt und werden durch unterschiedliche Zielgruppen betreut:

- Datenmodell und Geschäftslogik liegen in der Verantwortung der Fachanwender. Die Daten werden in Form eines konfigurierbaren Domänenmodells, die Geschäftslogik in Form von Geschäftsregeln beschrieben. Für die Formulierung der Regeln bietet die UOA eine domänenspezifische Sprache (DSL), eine Sprache deren Vokabular direkt aus dem Vokabular der Fachanwender abgeleitet ist (beispielsweise Finanz- oder Versicherungsmathematik).
- Die IT-Abteilung betreut die Plattform und den dazugehörigen Infrastrukturcode für die effiziente Speicherung des Domänenmodells und der zugehörigen Daten sowie die performante Ausführung der Geschäftsregeln. Sie sorgt auch dafür, dass Änderungen nachvollziehbar und reversibel sind, und schließt andere Systeme an den „fachlichen Datentopf“ an.

Der Aufbau einer User Owned Application ist in Bild 1 dargestellt. Das Domänenmodell ist in einem Repository abgelegt und kann über entsprechende Konfigurationsmasken zur Laufzeit



für die Fokussierung der UOA. Mit der erhöhten Flexibilität einer UOA gegenüber klassischen Unternehmensanwendungen steigt auch das Risiko von Fehlkonfigurationen. Um dieses Risiko zu reduzieren, ist eine intensive Konzeption von Sicherungsmechanismen notwendig.

Die beschriebenen Punkte zeigen, dass die Domänenmodellierung beim Entwurf und der Entwicklung ein besonderes Augenmerk erfordert. Gleiches gilt für den Betrieb einer UOA. Für das Domänenmodell und die Geschäftsregeln benötigen die Fachwender eine konsequente Governance, um einer Erosion vorzubeugen: Sind alle Elemente noch inhaltlich konsistent, und gibt es keine Duplizierung von fachlichen Sachverhalten? Diese Prüfung wird bei klassischen Unternehmensanwendungen im Rahmen der

verändert werden. Die fachlichen Daten der Anwendung werden durch den Benutzer entweder manuell gepflegt oder durch die Regelmaschine anhand der Geschäftsregeln berechnet.

Unterschiede

User Owned Applications unterscheiden sich hinsichtlich der Rollenverteilung und der Architektur von klassischen Unternehmensanwendungen. Welche zentralen Unterschiede gibt es hinsichtlich Entwicklung und Betrieb?

Domänenmodellierung

Eine UOA unterstützt die Änderungsgeschwindigkeit durch die Flexibilisierung des Domänenmodells und der Geschäftsregeln. Diese Flexibilität gibt es allerdings nicht umsonst. In Bild 2 ist zu sehen, dass eine Erhöhung der Flexibilität zu einer Reduktion der Fokussierung führen kann.

Genau wie zu wenig Flexibilität führt auch zu wenig Fokussierung zu einer niedrigen Änderungsgeschwindigkeit. Microsoft Access beispielsweise ist extrem flexibel bei der Definition von Datenmodellen (Tabellen & Beziehungen) und Geschäftslogik (Visual Basic), aufgrund des generischen technischen Vokabulars für die meisten Fachwender allerdings unbenutzbar.

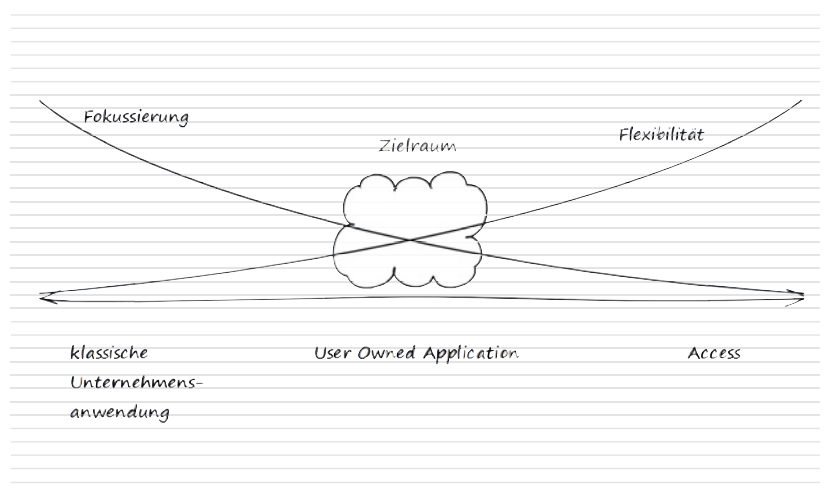


Bild 2: Einflussfaktoren

Eine UOA bietet dem Fachwender eine „dosierte Flexibilität“. Eine besondere Herausforderung beim Entwurf und der Entwicklung einer solchen Anwendung stellt die Annäherung an den im Bild angedeuteten Bereich dar, in dem die Anwendung so flexibel wie nötig und gleichzeitig so fokussiert wie möglich ist. Im Gegensatz zu klassischen Unternehmensanwendungen erfordern deshalb die Ermittlung von Änderungshot-Spots (Stellen, an denen sich die Fachlichkeit oft ändert) und die Definition stabiler Begriffe für die domänenspezifische Sprache besondere Sorgfalt. Die DSL ist das entscheidende Element

Konzeption vorgenommen, für UOAs erfolgt sie kontinuierlich.

Performance

Die Flexibilisierung des Domänenmodells und der Geschäftsregeln stellt auch die Software-Entwickler vor andere Herausforderungen als in einer klassischen Unternehmensanwendung. So muss beispielsweise eine Persistenz implementiert werden, die sowohl das Verändern der Entitäten und Attribute zur Laufzeit als auch eine performante Speicherung einer sehr großen Menge fachlicher Daten ermöglicht. Dazu sind in manchen Anwendungsszenarien

Technologien abseits der in der Welt der klassischen Unternehmensanwendungen allgegenwärtigen relationalen Datenbanken nötig (NoSQL etc.).

Für die Implementierung der domänenspezifischen Sprache gibt es grundsätzlich zwei Möglichkeiten mit verschiedenen Vor- und Nachteilen:

- **interne DSL:** Es wird eine bestehende „Wirtssprache“ (wie Groovy oder Scala) verwendet.
 - Vorteile: kurze Entwicklungszeit, Standardoperationen (Addieren, Subtrahieren etc.) vorhanden
 - Nachteile: zum Teil Abstriche bei der fachlichen Verständlichkeit, Sicherheitsrisiken durch Code Injection, Performance
- **externe DSL:** Es wird eine völlig neue Beschreibungssprache mit Grammatik erstellt.
 - Vorteile: keine Sicherheitsrisiken durch Code Injection, näher an der natürlichen Syntax
 - Nachteile: meist aufwändiger als eine interne DSL, Interpretation muss komplett entwickelt werden

In beiden Fällen stellt die performante Ausführung der Geschäftsregeln die Achillesferse der UOA dar. In klassischen Unternehmensanwendungen handelt es sich dagegen meist um kompilierten Sourcecode, dessen Laufzeitverhalten sich im Vorhinein intensiver untersuchen lässt.



Die Fachanwender konfigurieren selbst das Domänenmodell und die Geschäftsregeln.

So sind Dokumentation und Konfiguration identisch und kompakt in der Fachsprache beschrieben.

Daniel Kaminsky,
Consultant, Cofinpro AG

Fazit

User Owned Applications ermöglichen eine schnelle Anpassung der Fachlichkeit an geänderte Marktbedürfnisse durch die Verlagerung der „fachlichen Anwendungsentwicklung“ von der IT-Abteilung zu den Fachanwendern. Dass Änderungen an der Fachlogik ohne Software-Entwicklung im eigentlichen

Sinne auskommen, hilft den Fachabteilungen, eine kurze Time to Market zu erreichen, und entlastet gleichzeitig die IT-Abteilung.

Darüber hinaus bringt sie Qualitätsvorteile. Die Dokumentation und die Konfiguration sind identisch und kompakt in der Fachsprache beschrieben. Es ist nicht notwendig, eine separate Dokumentation aktuell zu halten. Diese Symbiose von Fachlichkeit und Anwendung führt auch dazu, dass die Fachanwender ein sehr gutes Verständnis „ihrer Anwendung“ haben – und behalten; sie ist in „ihrer Sprache“ beschrieben.

Der Artikel hat gezeigt, dass es eine Erhöhung der Änderungsgeschwindigkeit und die Reduktion der Änderungskosten nicht geschenkt gibt. Der Aufwand für Entwurf und Entwicklung einer UOA ist höher als bei einer klassischen Unternehmensanwendung. Er wird vor allem durch die Aspekte Domänenmodellierung und Performance getrieben.

Verschiedene Untersuchungen zeigen, dass die Kosten für die initiale Entwicklung einer Anwendung maximal ein Viertel von deren Gesamtkosten ausmachen. Vor diesem Hintergrund ist es durchaus sinnvoll, die erste Investition zu erhöhen, um längerfristig Kosten einzusparen und eine bessere Time to Market zu gewinnen.

KONSTANTIN DIENER
DANIEL KAMINSKY

Weiterführende Informationen:
www.it-daily.net

Downloads



Der Button führt Sie in der ePaper-Version direkt zum Ziel. In der Printversion nutzen Sie bitte den QR Code.



SMART DATA Summit



8. Dezember 2015
Empire Riverside Hotel,
Hamburg

Mit den richtigen Daten mehr Effizienz im Marketing



Information & Anmeldung: smart-data-summit.de