

Das Ende des **Software-Release-Zyklus?**

Dieser Artikel beschreibt, wie man die Dynamik der Marktanforderungen und die Strategie des Unternehmens ohne lange Software-Release-Zyklen optimal aufeinander abstimmt.

von Wolfgang Martin, Thomas Cotic und Lars Wunderlich

Serviceorientierte Softwareentwicklung, Agilität, Industrialisierung, Softwarefabriken und Geschäftsregel-Modellierung sind Schlagwörter, zu denen man gerne greift, um seine moderne, technische IT-Landschaft zu präsentieren. Wie lässt sich dies nun mit den mehr fachlich orientierten Zielsetzungen schneller Anpassung an Marktbedürfnisse, Restrukturierung der Geschäftsprozesse und kurzen Entwicklungszeiten mit hohem Return-on-Investment in Einklang bringen?

Agilität und Industrialisierung

Immer mehr Unternehmen bezeichnen sich heute als prozessorientiert. Prozessorientierung bedeutet ein konsequentes Denken in Geschäftsprozessen und Geschäftsregeln und damit eine optimale Adressierung der Kundenwünsche. Prozesse basieren auf einer gemeinsamen Steuerung, die die dem Geschäftsprozess-spezifische Logik, wie die Kontrolle des Ablaufs der Prozessaktivitäten, Einhalten von Deadlines und Ausnahmebehandlung, beschreibt. Sie werden softwareseitig, z.B. mittels eines entsprechenden Business Process Management (BPM), implementiert. Über Regeln beschreibt man die Entscheidungslogik, die die prozessübergreifenden Managementpolitiken und Prinzipien darstellt. Sie wird z.B. mit einer Regelmaschine im Rahmen von Business Rules Management (BRM) implementiert. Die Prozess- und die Regelmaschine dienen beide der Modellie-

rung, Ausführung und Governance der entsprechenden Logiken.

Prozesse und Regeln sind zueinander komplementär, denn ein Prozess nutzt typischerweise mehrere Regeln, während eine Regel in verschiedenen Prozessen vorkommen kann. Man sagt, Prozesse stünden zu Regeln in einem Verhältnis von m:n. Daher muss man Prozesslogik und Entscheidungslogik strikt voneinander trennen. So erhält das Business eine transparente Modellierung und Darstellung der Abläufe, Politiken und Prinzipien im Unternehmen. Dies ist die Voraussetzung für Agilität und Industrialisierung.

Agilität bedeutet die Fähigkeit zur Innovation und die Anpassungsfähigkeit der eigenen Geschäftsmodelle. Schnelle Reaktionen sowie pro-aktive Initiativen begeistern und binden Kunden und verblüffen und schlagen Wettbewerber aus dem Feld. Eine immer höhere Marktdynamik und die Einhaltung von Vorschriften sind die treibenden Kräfte für Agilität. Dynamik bedeutet, dass der Lebenszyklus von Prozessen und Regeln immer kürzer wird, also Änderungen in immer schnelleren Zyklen notwendig sind.

Antriebskräfte der heutigen Softwarebranche sind die Forderung zu kontinuierlicher Optimierung und weiterer Kostenreduktion: Die Industrialisierung hilft dabei mittels Automatisierung und Standardisierung. So werden Durchlaufzeiten, Durchlaufvolumen und die Qualität u.a. durch Wiederverwendung gesteigert.

Agilität und Industrialisierung sind aus der Sicht der Implementierung von Prozessen und Regeln mittels Informationstechnologie (IT) eigentlich zwei widersprechende Anforderungen, jedoch wenn Prozesse und Regeln im Kontext einer serviceorientierten Architektur (SOA) gemanagt werden, dann bringt man beide Zielsetzungen mittels eines SOA-basierenden BPM und BRM zusammen. Denn eine SOA ist eine spezielle Architektur, die darauf abzielt, „Software for Change“ zu ermöglichen.

Prozesse und Regeln

Der Begriff der serviceorientierten Architektur geht in viele Richtungen und hat viele Propheten. Oft allerdings liegt einer Palette von omnipotenten Softwarelösungen wie Enterprise Service Bus, Java Business Interfaces, Web Services und Co. letztendlich aber ein einfacher und verständlicher Wunsch zu Grunde: Kunden wollen Wiederverwendung bestehender Softwarefunktionen, sie wünschen sich mehr Kontrolle und gleichzeitig mehr Flexibilität. Die bestehende Software ebenso leicht wie Unternehmensbereiche, Mitarbeiteraufgaben und Geschäftsprozesse umstrukturieren können, um sich besser am Markt auszurichten, ist aber mit statischen Codestrukturen schwer.

Der Kitt schwenkt zu Services zwischen den Anwendungen, gespickt mit technischen Kommunikationsprotokollen wie XML und die Middleware-Technik

nologien, die uns Ausfallsicherheit, Clustersupport und Verschlüsselung bieten. Sind jedoch ohne „Gehirn“ faktisch nutzlos. Anstatt wie früher alle diese Systeme unter eine gemeinsame Kontrolle zu stellen, die allen vorangeschaltet die Überwachung übernimmt, begreifen wir heute die Regeln selbst – und ihre Anwendung als einen Service wie jeden anderen auch. Die Prozessmaschine orchestriert sie, führt also die Regeln mit anderen Services zu einem neuen Gesamtbild zusammen.

Rule-Services werden somit zu einer Art eigener Kategorie von Services. Ein Rule-Service kann als Kapselung komplexer Regeln verstanden werden. Mit anderen Worten, ein Rule-Service kann auch einen anderen Rule-Service in seiner Implementierung unabhängig von einer steuernden SOA aufrufen. Das ist analog dem Unterprozessprinzip im BPM. Damit wird eine Regelmaschine zum Bestandteil einer SOA-Infrastruktur und die Entscheidungslogik wird Teilmenge der Geschäftslogik. So haben die Regeln im Rahmen einer SOA ihre eigene Administration im Repository, in der Registrierung und in der Governance-Infrastruktur und basieren auf dem gleichen Businessvokabular wie die Prozesse (Abb. 1).

Die Verwendung von Regeln als Services zeigt die Eleganz des SOA-Ansatzes. Regeln werden eben einmal modelliert und implementiert und stehen dann zur Verwendung in verschiedenen Prozessen zur Verfügung.

Business Rules Management – Agilität schaffen und managen

Der Gewinn der Anwendung eines Business-Rules-Management-Systems im Verhältnis zur traditionellen If-Then-Else-Programmierung ist zweifelsohne die Fähigkeit mit Kunden auf einer gemeinsamen Wellenlänge zu arbeiten und sich bei der Beschreibung von Geschäftslogiken weitgehend unabhängig von den Problemen wie Programmierschlüsselwörtern, Framework-Produkten, Compiles und Unit-Tests zu bewegen. Wir reden in dieser idealisierten Welt nicht mehr über Programmiersprachen und endlose Word-Dokumente mit Anforderungspapier, die zu reviewen und dann aufwändig in Source zu transformieren sind. Analyse, Prototyping, Modellierung, Ausführung

und Test gehen fast nahtlos ineinander über. Mit einfach zu erstellenden Regelbäumen unter einer ansprechenden Oberfläche bildet beispielsweise Visual Rules (www.visualrules.de) von Innovations für den Endkunden leicht nachvollziehbare Entscheidungsgraphen ab, die in Testläufen dem Kunden bereits direkt nach der gemeinsamen Modellierung plastisch die Konsequenzen seiner Entscheidungen aufzeigen können. Andere Produkte bevorzugen textuelle „Wenn-Dann“-Sätze oder sogar noch niedere Formulierungen, die sehr programmiersprachennah sind.

Allen gemeinsam ist, dass der Kunde in die Lage versetzt werden soll, unmittelbar in der eigenen Wissensdomäne seine Kreativität und sein Know-how einzubringen und auszuleben, ggf. nach kurzer Einarbeitung ohne aktive Unterstützung eines Entwicklers. Der IT-Berater oder -Designer konzentriert seine Arbeit auf die Zurverfügungstellung einer architektonischen Basis und die Öffnung von Schnittstellen zu normalerweise arbeits-, kosten- und änderungsintensiven Bereichen der Applikation bzw. des Geschäftsprozesses. Der Business-Domain-Experte agiert ohne langen Entwicklungsprozess direkt in und an der Entscheidungslogik, die sein Geschäft bewegt. Auf psychologischer Ebene geben wir dem Kunden das Gefühl zurück, nahezu IT-unabhängig direkt auf Marktbedürfnisse reagieren und damit als Auftraggeber der Software auch die „Herrschaft“ über selbige zurückerlangen zu können, die wir ihm mit langen Release-Zyklen und veralteter Software oftmals nehmen.

Rein technisch gesehen, kommt der Rule Service auf Grund seiner Einbettung in die SOA oft als Web-Service-Angebot daher. Die Regeln sind z.B. als JAR verpackt und werden in einem http-Server aufrufbar gemacht.

BRM-Services lohnen sich tatsächlich dort, wo eine große Menge an Regeln technisch unabhängig von umgebender IT-Software ins Geschehen eintritt. Rules sind also keine klassische Lösung für die Konfiguration von IP-Adressen und Ports oder das Parsen von verschlüsselten Daten oder Logiken zur Erzeugung von Check-Summen. Business Rules sind Geschäftsregeln, weil es genau die Regeln sind, die sich gerne in Prosatexten

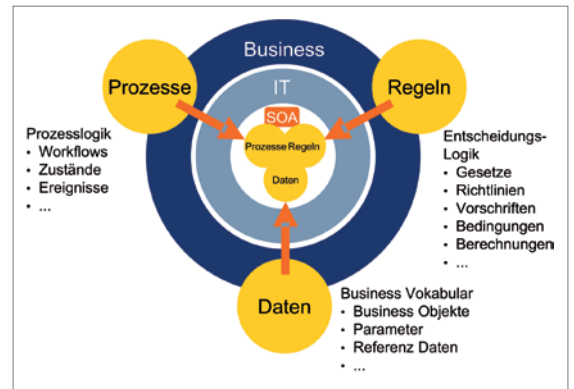


Abb. 1: SOA-basierte Regeln

Unternehmen	Szenario	Einsparung
PostFinance, Schweiz	Transaktionsanalyse, Geldwäschereierkennung	25%
Sungard Futures Systems, USA	Handelssystem für börsennotierte Derivate	50%
MobileSelect, Australien	Web-basiertes Beratungsportal für Geräte und Tarife	66%
Sungard Forbatec	Steuerung eines Fond Management Systems	70%
		Einsparung
Initialeinsparung		25% - 50%
Änderungseinsparung		50% - 70%

Abb. 2: Gehobene Einsparungspotenziale

mit Entscheidungstabellen wiederfinden oder als „undokumentierbare“ oder sogar politisch oftmals „geheime“ Logiken beschrieben werden, die man dem artfremden IT-Dienstleister eigentlich nur ungern offenbaren will – non-disclosure agreement hin oder her. Somit finden wir hier oftmals Marketinginstrumente wie Filterung oder Sortierungen von Angebotslisten zur Unterstützung von Top-sellern und Superpreisangeboten. Wir reden über Preiskalkulationsmodelle zur Optimierung von Angebot und Nachfrage, wir sprechen über Risiko- und Tarifiermittlungsmodelle für den Verkauf von Dienstleistungen. Wir finden Marktanalysen und Produktoptimierungen beispielsweise in allen Bereichen des Dienstleistungssektors von Großbanken und Versicherungen bis hin zu kleineren Webportalen und Internetanbietern. In diesen Fällen identifizieren Kunden mit dem IT-Berater ihre „Hotspots“ hoher Dynamik und lassen sich vom IT-Dienstleister dedizierte Services innerhalb ihrer SOA schaffen, die ihnen unmittelbaren Zugang zu diesen Logiken innerhalb des Geschäftsprozesses verschaffen.

Im Fall einer Software wie Visual Rules kann es bedeuten, sich nach kurzer Analyse und einem Vorgespräch, gemeinsam an die Modellierung von Regeln in einer clientseitigen Eclipse-Umgebung zu machen und dort in Regelbäumen die eigenen Entscheidungsschritte niederzulegen. Testskripte und Dokumentationsmöglichkeiten heben den daraus generierten Java-Code auf eine Stufe der Testsicherung und Dokumentationsqualität, wie sie innerhalb der reinen Programmierung selten mit Javadoc und JUnit-Tests erreichbar sind bzw. praktiziert werden. Mit wenigen Klicks ist die neue Implementierung hinter der Web-Service-Schnittstelle auf einen Server deployt und einsatzbereit.

Das Ende des Software-Release-Zyklus?

Der klassische Software-Release-Zyklus von Konzeption, Realisierung, Test mit begleitender Qualitätssicherung und Dokumentation ändert sich wie wir gesehen haben beim Einsatz eines BRM Systems. Die Konzeptionsphase wird deutlich verkürzt, da direkt mit der Tool-gestützten Modellierung der Geschäftsregeln begonnen wird. Bereits bei der Modellierung werden die Testdaten und erwarteten Referenzergebnisse definiert. Dieser Ansatz ermöglicht nicht nur ein iteratives Vorgehen bei der Definition der Geschäftsregeln – was bei komplexen Regeln unverzichtbar ist –, sondern bildet auch die Basis für die automatisierten Testverfahren im TQM. Die Kodierung der Geschäftsregeln entfällt, und damit auch die Abstimmungsschleifen zwischen Fachbereich und IT, die benötigt werden, bis die Geschäftsregeln fehlerfrei implementiert sind. Statt eines traditionellen Release-Verfahrens kommen wir so zu einem dynamischen, kontinuierlichen Lebenszyklusmanagement der Geschäftsregeln.

Natürlich gehört es sich auch bei BRMS wie bei jeder Produktlinie und Geschäftsentscheidung, Kosten und Nutzen gegenüber zu stellen. Haben wir ein tatsächliches Problem, das mit einem solchen Tool gelöst werden kann oder meinen wir, ein solches bei uns entdecken zu können, weil uns eine Pro-

duktbroschüre Probleme beschreibt, die wir assoziativ auf unseren Kontext übertragen?

Einer IDC-Studie zufolge werden immerhin 9 % der Geschäftsregeln täglich, immerhin weitere 17 % wöchentlich und nochmals 17 % monatlich geändert. Somit haben wir schon gut 45 % der Regeln, die sich innerhalb eines angenommenen Release-Zyklus von mindestens einem Monat ändern würden. Klar ist auch, dass die Dokumente und die vom Kunden beschriebenen Logiken umso überholter sind, je mehr Zeit zwischen Analyse und In-Produktionsnahme vergeht. Release-Zykluslänge und Änderungsmenge sind also essenzielle Gegenspieler von Flexibilität im Geschäft.

Wichtig ist hier der Einsatz von Business Rules, SOA und Regeln dort ansetzen zu lassen, wo auch der häufigste Änderungsaufwand entsteht bzw. der Return-on-Investment aufgrund der kürzeren Release-Zyklen und des schnelleren Time-to-Market der eigenen Firmenphilosophie und –strategie wirklich sichtbar wird. Wir sprechen hier auch von Continuous Integration als Vorgehensmodell: Regeln lassen sich kurzfristig ergänzen und modifizieren; tägliche Releases bieten kontinuierliche Weiterentwicklung; Codegenerierung, Dokumentation und Deployment werden automatisiert. Ein falsch oder nicht angewandtes Tool bringt natürlich auch keinen strategischen Vorteil.

Ein hoher Grad an Automatisierung ist aber nur dann möglich, wenn der Anwender die Regelmaschine optimal zum Einsatz bringen kann. Bei

komplexen Szenarien und neuen Produktteilen bedeutet das ggf. Consulting-Support und eine klare Etablierung von Geschäftsprozessen, Aufgabenträgern und Release-Zyklus-Entscheidungen. Denn schnell ist zu Beginn auch mal unbedachte Logik in Produktion gebracht, die der althergebrachte Prozess durch die Menge der reviewenden Personen manchmal verhinderte. Ist die erste Hürde und Eingewöhnung aber genommen, werden ggf. schnell Einsparungspotenziale durch die engere Zusammenarbeit aufgedeckt.

Innovations veröffentlichte 2007 Informationen von vier ihrer international operierenden Kunden, die in verschiedenen Szenarien zwischen 25 % und 70 % Einsparung diagnostizierten. Wenig überraschend findet sich das größte Potenzial bei Änderungsaktivitäten und deren Implementierung, die mit den klassischen Datenbankschaltern und Property-Files zur Konfiguration der Software nur noch wenig gemeinsam haben.

Wer der Überzeugung ist, auch solche Stellen in seiner Software identifiziert zu haben, die es wert sind, flexibler gestaltet zu werden, hat mit einem Business-Rules-Management-System nicht nur im Kontext einer SOA eine Chance, wirtschaftliche Potenziale zu heben.

BRMS rationalisiert zudem den klassischen Entwickler nicht etwa weg, sondern gibt dem IT-Dienstleister die Möglichkeit, seinem Fachbereich agilere Entwicklungsprozesse anzubieten, um Wettbewerbsvorteile zu erringen und bestehende Software und Modifikationen schneller zu harmonisieren. ■



Dr. Wolfgang Martin ist unabhängiger Analyst und europäischer Experte in den Bereichen BI/CPM, Business Integration, SOA und CRM. Er ist Koautor und Herausgeber von Büchern und Fachartikeln zu den genannten Themen.



Thomas Cotic ist Gründungsmitglied und Geschäftsführer der Innovations Softwaretechnologie GmbH, die u.a. Beratung zum Thema BRM und den Einsatz ihres Produktes Visual Rules anbietet.



Lars Wunderlich ist Systemarchitekt im B2B-Bereich der TUI InfoTec GmbH und beschäftigt sich mit der Umsetzung von Architekturen im Java-Enterprise-Umfeld. Er ist Autor zahlreicher Publikationen im Bereich Java.