

## BPM und Microservices

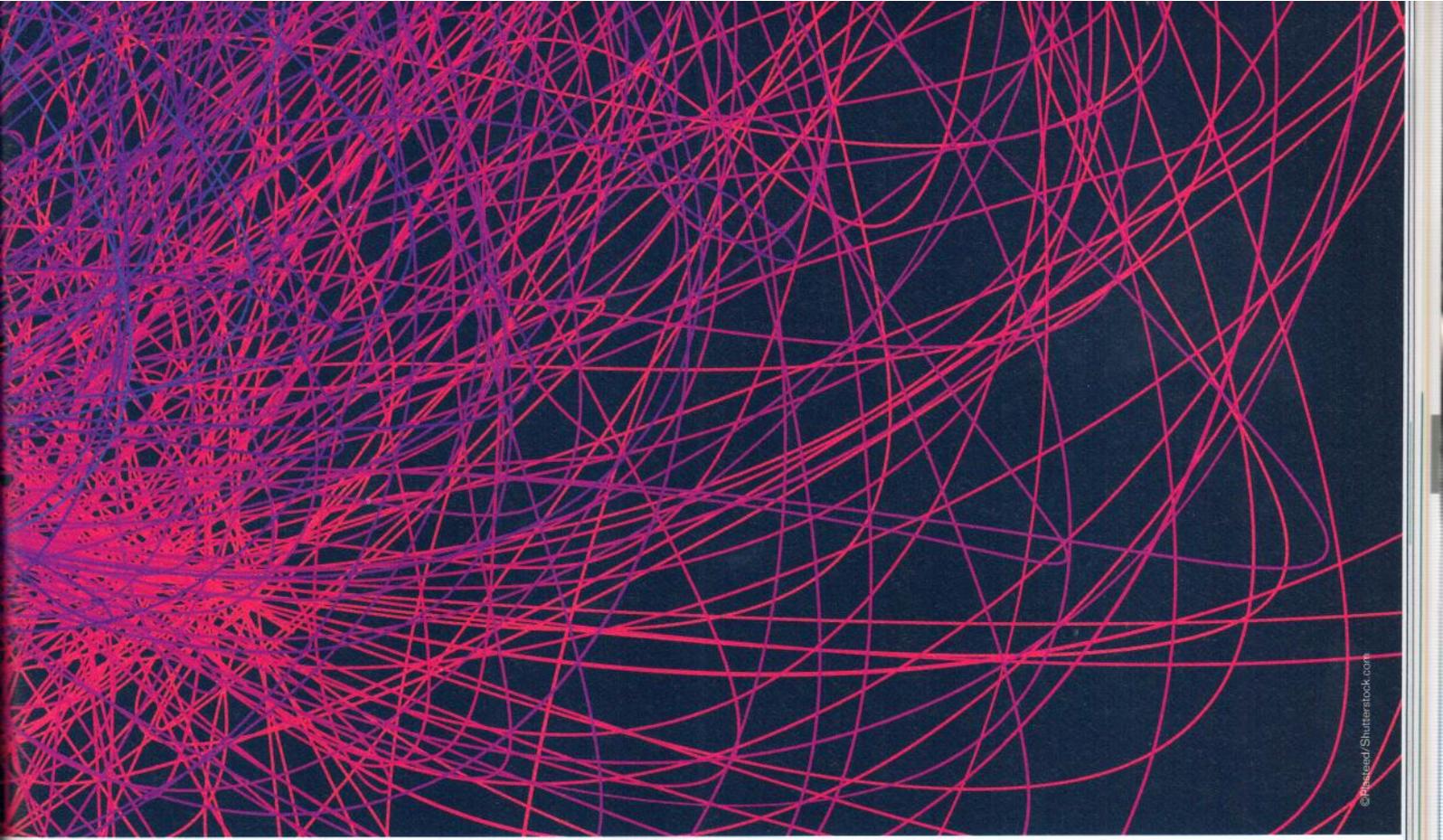
# Die Ordnung über den Haufen werfen?

Steht klassisches Business Process Management (BPM) im Widerspruch zur IT-Architektur nach dem Microservices-Ansatz? Der korrekte Microservices-Ansatz erlaubt immerhin keine zentrale unternehmensweite Prozesssteuerungskomponente. Dieser Artikel schildert, wie eine korrekte Implementierung von BPM-Systemen nach Microservices-Ansatz aussehen müsste, und analysiert die Auswirkungen dieser Fusion der Gegensätze.

AUTOR: AZMIR ABDI

Für was stehen überhaupt BPM, SOA und Microservices? Um ein Verständnis für die elementaren Eigenschaften der verschiedenen Ansätze für die Prozessoptimierung zu schaffen, wird im ersten Teil des Artikels auf die Begriffe BPM, SOA und Microservices,

deren jeweilige Motivation sowie die Unterschiede zwischen SOA und Microservices eingegangen. Im zweiten Teil wird das auf Microservices-Ansatz angewandte, leicht angepasste BPM als rBPM (Reverse BPM) präsentiert.



©shutterstock/Shutterstock.com

## BPM, SOA UND MICROSERVICES

Business Process Management (BPM) ist ein vollumfänglicher Ansatz der Unternehmensorganisation, konzentriert auf die kontinuierliche Optimierung und damit auf die Gewinnmaximierung des Kerngeschäfts. Dabei wird das Kerngeschäft in die End-to-End-Geschäftsprozesse eingegliedert, die die komplette Eigenleistung vom Kundenauftrag bis zu der abschließenden Kundenlieferung umfassen. Die IT-gestützte Lösung von BPM sind die BPM-Systeme (BPMS) bei denen die End-to-End-Geschäftsprozesse durch integrierte zentrale Business Process Engines (BPE) automatisiert gesteuert werden.

Der BPM-Ansatz spielt hervorragend mit der serviceorientierten Architektur (SOA) als Enterprise SOA zusammen, wie im Artikel von Dirk Slama [1] bereits vor über zehn Jahren beschrieben wurde. Mittlerweile wissen wir, dass SOA in der ursprünglichen Form bereits veraltet ist. Der neue, maßgeblich von Martin Fowler beschriebene Microservices-Ansatz ist die nächste Evolutionsstufe der Enterprise-Architektur [2].

### SIND MICROSERVICES KEINE SOA?

SOA ist, genau wie Microservices, eine Antwort auf die schlechte Wartbarkeit der komplexen monolithischen IT. Bei SOA steht die Beherrschung der Komplexität durch Strukturen im Vordergrund. Auch im Fokus steht die Optimierung der IT durch Eliminierung der Verschwendung der mehrfach implementierten Problemlösungen mittels Wiederverwendung. Jedes Problem soll unternehmensweit einmalig gelöst werden. Die Lösung wird intern publiziert und dabei darauf geachtet, dass

sie wiederverwendet wird. Die Wiederverwendung von Services, die ein zentrales Domain Model nutzen, ist bei SOA maßgeblich. Der Enterprise Service Bus (ESB) als zentrale intelligente Integrationskomponente verbindet die Services. Die gesamte Unternehmensarchitektur ist in Schichten organisiert, wie es in **Abbildung 1** beispielhaft dargestellt wird.

BPMS als Zusatz übernimmt teilweise die Serviceorchestrierung, die gewöhnlich über ESB stattfindet. BPM kann damit auf SOA nativ angewandt werden. Die Unterschiede zwischen dem, wie SOA zuletzt verstanden wurde, und dem Microservices-Ansatz sind signifikant! Matt McLarty hat in einem Artikel [3] die Abgrenzung von Microservices zu SOA sehr lesenswert geschildert. Einer der Widersprüche von SOA und BPM zum Microservices-Ansatz liegt genau beim ESB. Der Microservices-Ansatz setzt auf leichtgewichtige Integration. Mit dem „Smart endpoints and dumb pipes“-Prinzip soll statt monolithischer EAI, wie ESB bei SOA angewandt wird, die in SOA angepriesene Abkoppelung der Systeme endlich Wirklichkeit werden. Dafür spricht auch Conways Gesetz, an dem sich der Microservices-Ansatz orientiert. Nach Conway [4] werden IT-Systeme unweigerlich nach den Kommunikationsstrukturen des Unternehmens organisiert. Eine nach SOA strukturierte Organisation weist Einheiten, die innerhalb der horizontalen SOA-Schichten positioniert sind, auf. Dadurch, dass sich bei SOA eine fachliche Domäne vertikal, von der Oberfläche bis in die Persistenz, durch alle Schichten streckt, ist diese damit auch auf mehrere Organisationseinheiten verteilt. Für eine Änderung, die in diese fachliche Domäne im-

plementiert werden muss, müssen sich mehrere Organisationseinheiten abstimmen. Ergebnis davon ist die Notwendigkeit der zentral koordinierten Releases, um diese Änderung herauszubringen.

Mit dem „Organized around business capabilities“-Prinzip wendet Martin Fowler das Conway-Gesetz auf den Microservices-Ansatz an. Die Organisation eines Unternehmens soll anhand der fachlichen Domänen strukturiert sein, die IT ebenso, mit dem Ziel, die Änderung an einer fachlichen Domäne nur innerhalb einer Organisationseinheit umsetzen zu können. Langsame Abstimmungen und komplexe Releases sollen als Verschwendung eliminiert werden. Der Microservices-Ansatz (Abb. 2) ist die Antwort auf die nicht funktionale Anforderung, Änderungen schnell umzusetzen. Zurück zum ESB bei SOA. Wie oft kommt es vor, dass ein Unternehmen eine Organisationseinheit hat, die in der IT vom ESB abgebildet werden kann? Da fallen einem nur

kuriose Beispiele ein: Ein Team mit Aktentransportwägen, ausgestattet mit einem mobilen Arbeitsplatz, wobei die Akten unterwegs für die nächste Abteilung im Zielformat übersetzt werden (Transformation und Mapping).

## SOA HEISST ZENTRALE PLANUNG UND STEUERUNG

Sowohl SOA als auch BPM gehen von einer zentralen Planung und Steuerung aus. Nun, was soll schon daran falsch sein, ein Unternehmen zentral zu planen und zu steuern? Schließlich haben alle Unternehmen genau hierfür das Management, das zentral plant und steuert. Ist es so? Steuern die Manager die Umsetzung direkt, ohne die Zwischenebenen? Oder ist es nicht so, dass bei erfolgreichen Unternehmen das Management die Leitplanken mittels der Vision und der Mission setzt? Daraus folgen Unternehmensziele. Üblicherweise werden die Ziele, und damit auch die Pläne,

auf fachliche Einheiten heruntergebrochen. Die Teams der Fachbereiche erarbeiten daraus für sich selbst detaillierte Pläne und sind selbst für die Umsetzung verantwortlich. Es existiert kein bis zum letzten Detail ausgearbeiteter zentraler Masterplan.

Erfolgreiche Unternehmen leben es vor: Agile Teams versuchen, eigene abgeleitete Ziele mittels eigener Planung und mit integrierter Optimierungsbestrebung (wie Sprint-Retrospektive bei Scrum) optimal zu erreichen. Das Management begnügt sich mit regelmäßigen Rückmeldungen der Ergebnisse aus den Teams heraus (Reporting) und prüft regelmäßig, ob sich das Unternehmen in der Gesamtsicht (Big Picture) auf dem richtigen Kurs befindet. Bei Enterprise SOA wird aber genau mit der zentralen Steuerung der Prozesse durch BPM dieser zentrale Masterplan aufgebaut, gesteuert und kontrolliert. Durch SOA Governance kontrolliert ein Gremium die IT-Änderungen und stellt si-

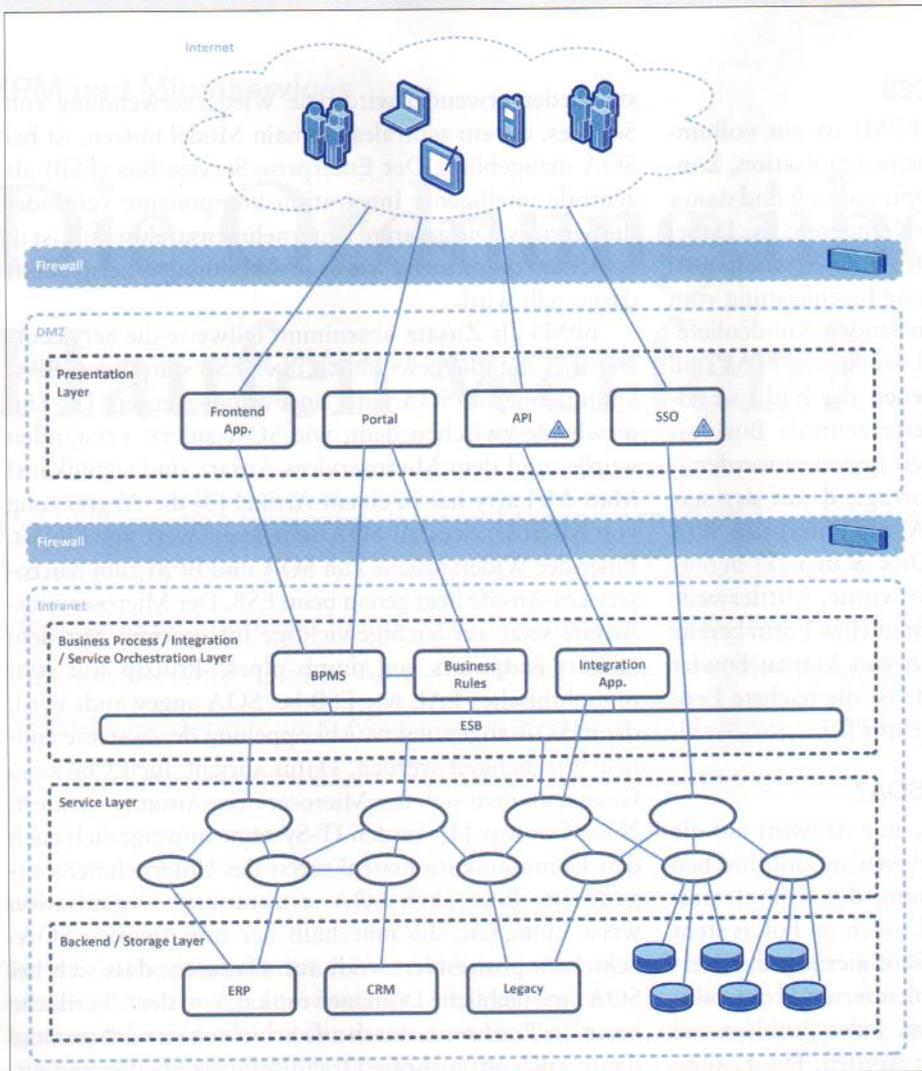


Abb. 1: Serviceorientierte Architektur (SOA)

cher, dass diese zentral abgestimmt sind. Zentrales Domain Model, Systemintegration und Technologieauswahl sind stark reguliert. Dies bestätigt nochmal Conways Gesetz. Die von der IT geschaffenen künstlichen Strukturen müssen mit hohem Aufwand aufrechterhalten werden. Ansonsten würden sie übergangen. Als Beleg suche man bei SOA nach direkten Schnittstellen zwischen Frontend- und Backend-Systemen.

Andersherum betrachtet könnte man sagen, dass SOA und BPM richtig am Platz in Unternehmen sind, bei denen jeder Schritt nach zentraler Planung haargenau umgesetzt werden muss, wo keine ungeprüften Optimierungsaktionen zugelassen sind, zum Beispiel in der stark regulierten Pharmaindustrie. Die meisten anderen Unternehmen wollen agiler werden und von der Erfahrung der eigenen Mitarbeiter profitieren. Für die IT heißt dies, mehr Agilität zulassen, Änderungen schneller zulassen, die IT auf Änderungen ausrichten. Keine zentrale monolithische IT, sondern kleine, nach Fachlichkeit getrennte Applikationen, die zu den Umsetzungsteams genau passen und auch in der heutigen Welt der Agilität imstande sind, im Gleichschritt mitgehen zu können.

Dieser Artikel möchte sich nicht damit befassen, wie diese agile Architektur erreicht werden kann. Dafür gibt es bereits einige Veröffentlichungen, zum Beispiel die Bücher von Eberhard Wolff [5] oder Matt McLarty [6], aber auch die Artikel über Self-contained Systems [7], die eine konkrete Umsetzung des Microservices-Ansatzes beschreiben. In diesem Artikel geht es primär um BPM mit der spannenden Frage „Wo bleibt BPM in dieser neuen agilen Welt der Microservices?“

### BPM SOLL SICH AUCH AN CONWAY ORIENTIEREN

BPM besagt, dass die End-to-End-Prozesse in einem Unternehmen analysiert, erfasst, gesteuert, überwacht und optimiert werden sollten. Die Unternehmen, die dies IT-gesteuert umsetzen, werden nach ISO 9001 oder nach CMMI auf dem höchsten Level zertifiziert. Wie

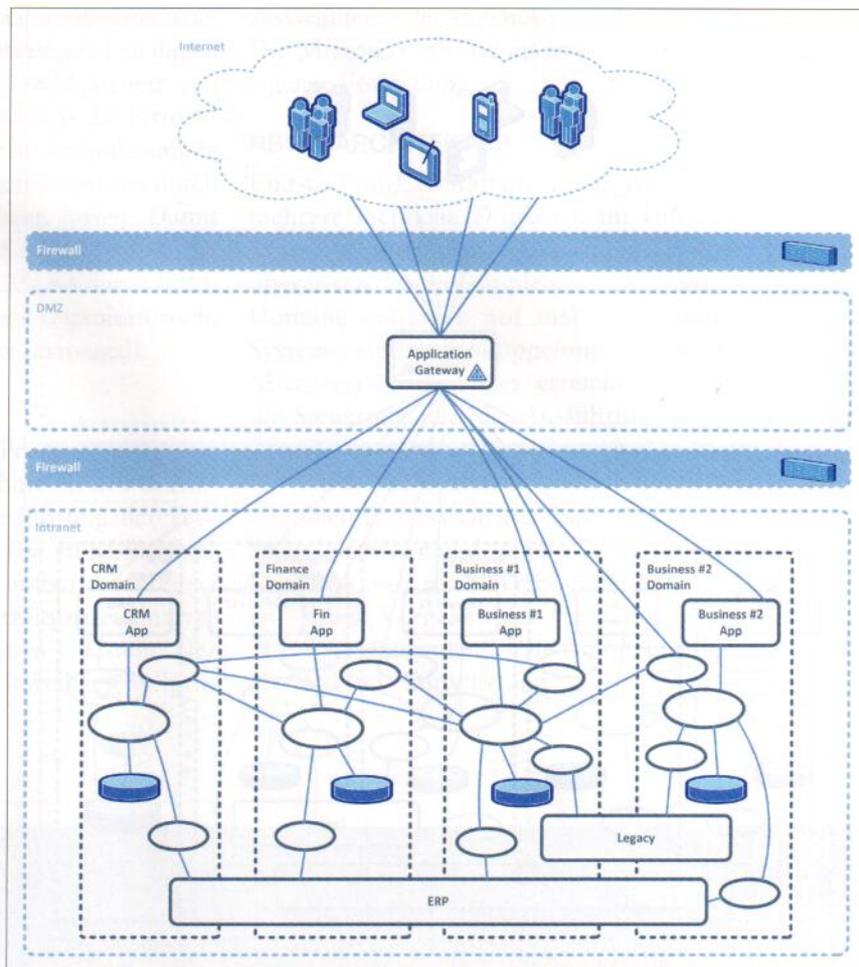


Abb. 2: Enterprise-Architektur nach Microservices-Ansatz

die Umsetzung aussehen soll, steht den Unternehmen frei, dennoch besagt die BPM-Lehre, dass die Prozesssteuerung aus einem BPM-System erfolgen soll [1]. Diese Forderung des BPM nach einer zentralen Steuerung soll der Forderung nach einer zentralen automatisierten Prozessauswertung weichen. Dem Management reicht es, in regelmäßigen Abständen die Ergebnisse zu berichten. BPM soll die Umsetzung der Prozessdigitalisierung nicht vorgeben, genauso wie die zentrale Planung der Planwirtschaft noble Ziele verfolgt, in der Umsetzung aber nicht gegen die freie Marktwirtschaft ankommen kann. So müssen auch die IT-Strukturen in Konzernen freier werden. Für BPM bedeutet das, dass es keine zentrale Steuerungskomponente der End-to-End Prozesse geben sollte, auch keine detaillierte zentrale Planung. BPM kann unterstützen, wenn die Prozesstransparenz aufgrund der Komplexität und der Unternehmensgröße leidet. BPM soll die Prozesse sichtbar machen und die Optimierungspotenziale im Sinne der Unternehmensziele aufdecken.

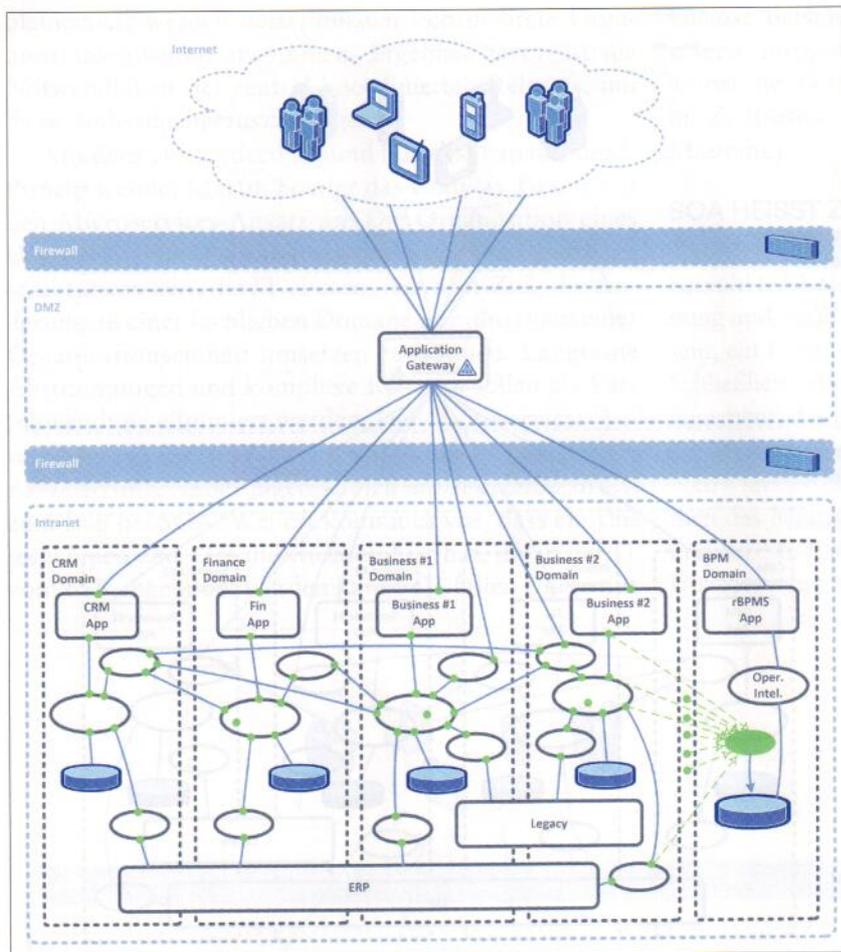


Abb. 3: BPM als Reverse BPM (rBPM) nach Microservices-Ansatz

### REVERSE BPM (RBPM)

Das hier als Reverse BPM (rBPM) betitelte BPM versucht, die beschriebenen Eigenschaften des Microservices-Ansatzes einzuhalten. rBPM verfolgt weiterhin die gleichen Ziele wie BPM, indem die Unternehmensprozesse analysiert, erfasst, gesteuert, überwacht und optimiert werden. Der Unterschied liegt in der Prozesssteuerung. Wie der Name Reverse BPM besagt, wird die Idee des Reverse-Engineerings angewandt, indem die Prozessabläufe mittels Operational Intelligence [8] rekonstruiert werden.

### PROZESSANALYSE, -ERFASSUNG UND -ÜBERWACHUNG

Mit Operational Intelligence (OI) wird Prozesstransparenz erreicht, indem die Daten aus der Verarbeitung (Operations) gesammelt und zu Informationen interpretiert (Intelligence) werden. Das Vorgehen ist mit Business Intelligence (BI) vergleichbar, jedoch unterscheidet es sich in der Geschwindigkeit der Bereitstellung der Informationen. Im Gegensatz zu BI, wo Informationen

in Form von Reports frühestens am Folgetag, gewöhnlich aber nach Monats-, Quartals- oder Jahresabschluss berichtet werden, ermöglicht OI die Gewinnung von Informationen über Geschäftsprozesse nahezu in Echtzeit. Das Verfahren ist ebenso unter dem Namen Business Activity Monitoring (BAM) bekannt [9].

Die operativen IT-Systeme sollen fachliche Events an OI melden, und das nicht einmalig, sondern ständig für jede Prozessinstanz. Für welchen Prozess wird gerade was gemacht und für welche Prozessinstanz in welcher Version? Ein Reverse-BPM-System (rBPMS) soll diese explizit gesammelten OI-Prozessinformationen analog zum BAM kontinuierlich auswerten. Das System soll zusätzlich imstande sein, einen Prozessablauf vollständig zu rekonstruieren und ihn auch nach Business Process Model and Notation (BPMN) darstellen können. Unter der Vorgabe, dass alle fachlichen Prozessschritte an OI gemeldet werden, ist damit die Prozessanalyse und -erfassung teilautomatisierbar. Die erfassten Abläufe würden in allen Fällen immer den Istzustand abbilden. Neue erfasste, noch unbekannte

Abläufe könnten dem Process Owner zum Review gemeldet werden.

### PROZESSSTEUERUNG UND -AUSFÜHRUNG

Bei rBPM werden Service- und Human-Tasks nach eigenen Teamvorgaben durch Microservices implementiert. Falls der Microservice selbst einen eigenen internen Status über lang laufende Prozesse verwalten muss, spricht hier nichts dagegen, eine eingebettete Business Process Engine, wie zum Beispiel Camunda BPMN Engine, einzusetzen [10]. Der Process Owner kann die Sollprozessdefinition im rBPM-System für Soll-Ist-Vergleiche aufnehmen. Entwicklung und Test können diese Vergleiche als Akzeptanzkriterien nutzen. Die Angleichung von Ist und Soll soll auf Entwicklungs- und Testumgebung erfolgen, bis eine Übereinstimmung erreicht ist. Auf der Produktivumgebung sollten dann keine größeren Überraschungen erfolgen.

Der Unterschied liegt darin, dass beim rBPM das BPM-System die Prozesse nachgelagert nach dem Ablauf kontinuierlich analysiert und falls möglich mit einem

Soll-/Planmodell vergleicht. BPM ist bei der Prozesssteuerung nicht führend. Dafür sind die jeweiligen beteiligten Fachapplikationen allein zuständig. rBPM steuert den Prozessablauf nur indirekt, indem es durch die Prozesstransparenz dem Process Owner unerwartete/unbekannte Änderungen meldet. Er könnte diese quittieren oder durch einen Change Request (CR) korrigieren lassen. Damit wird sichergestellt, dass auch bei rBPM die Prozesse durch den zugeordneten Process Owner, und damit durch BPM, gesteuert werden. Dies erfolgt zwar rein organisatorisch, dennoch durch ein IT-System verwaltet (managed).

### PROZESSOPTIMIERUNG

Beim klassischen BPM erzeugt der Process Owner ein Sollprozessmodell, das durch Aufnahme im BPMS zum Ist wird. Das BPMS steuert die Ausführung der Prozessinstanzen nach dem Istmodell. Die BPM-Prozessoptimierung beim rBPM geschieht, indem der Process Owner als Teil von BPM auf die Echtzeitprozessanalyse reagiert und bei Änderungsbedarf einen CR stellt. Der Process Owner stellt die CRs für die jeweiligen Sollpro-

zesssegmente der zugehörigen Microservices-Domänen. Die Microservices-Teams implementieren die CRs nach eigener Vorstellung.

### RBPM-ARCHITEKTUR

End-to-End-Geschäftsprozesse durchlaufen operativ mehrere fachliche Domänen im Unternehmen. Nach Martin Fowler sollten diese nicht in einem einzelnen operativen System implementiert werden, sondern je Domäne aufgeteilt auf mehrere Systeme. Damit die Systeme eigene Entkoppelung als das Hauptziel des Microservices-Ansatzes erreichen können, sollen sie die Steuerung und die Ausführung der verantwortlichen Tasks selbst implementieren. Für die IT-Architektur würde das bedeuten, dass die Implementierung der einzelnen Prozessschritte/Tasks in Microservices jedem Product Owner und seinem Team frei überlassen werden kann, just nach der Empfehlung von Martin Fowler – keine Vorgaben der Technologien durch Business Process Management, freie Umsetzung der Optimierungen und freie Entwicklung.

Anzeige



Dein Potenzial  
Dein Training  
Dein Vorsprung

## Enterprise-Architekturen mit Angular

mit Manfred Steyer

In diesem Tutorial erklärt Manfred Steyer, Google Developer Expert, wie sich mit Angular große und wartbare Geschäftsanwendungen entwickeln lassen.



**entwickler-tutorials.de**

/tutorials/enterprise-architekturen-mit-angular

Um den Prozessablauf der End-to-End-Geschäftsprozesse vollständig rekonstruieren zu können, muss die Operational Intelligence des rBPMS alle hierfür notwendigen Daten erhalten. Damit die operativen Systeme die Entkoppelung beibehalten, sollte dies asynchron erfolgen (Abb. 3).

Der BPMN-Standard könnte mit diesen asynchronen Prozessevents ebenso erweitert werden. Alle möglichen BPMN-Ereignisse, die als Event gemeldet werden sollten, könnten standardisiert werden. Damit könnten sowohl Individual- als auch Standardsoftwarehersteller rBPM-Systeme herstellen, die alle benötigten BPM-Anforderungen bereitstellen. Bereits eingesetzte rBPM-Systeme könnten durch Standardisierung austauschbar werden.

#### FAZIT

Es können weitere Vorteile des nach rBPM implementierten BPM-Systems zusätzlich zur Architekturkompatibilität zum Microservices-Ansatz aufgelistet werden. Als Erstes spricht die einfache, nicht invasive Einführung in die existierende IT-Landschaft dafür. Dadurch, dass ein rBPMS die Prozessausführung nicht direkt steuert, müsste diese nicht aus den bestehenden operativen Systemen herausgelöst und nach BPMS migriert werden. Ein rBPMS müsste als IT-Anwendung der eigenen unterstützenden Querschnittdomäne abgekoppelt von den operativen Domänen agieren. Dadurch entstehen keine Performanceeinbußen, da die operativen Systeme die Events asynchron verschicken können.

Wie dem Microservices-Ansatz selbst, mangelt es rBPM im Gegensatz zum klassischen BPM an Eleganz und Handhabung, wie allen dezentralen gegenüber zentralen Ansätzen. Fein strukturierte Gesamtarchitektur der IT bei SOA soll durch eine auf den ersten Blick wild aussehende Architektur der Microservices ersetzt werden. Des Weiteren wäre keine unternehmensweite zentrale, vom BPMS bereitgestellte Taskliste für die Ausführung der Human-Tasks mehr vorhanden. Diese zusätzlich bereitzustellen, würde aber auch gegen den Microservices-Ansatz sprechen, da sie wiederum in mehreren fachlichen Domänen operativ eingreifen würde. Die Prozesstransparenz bei rBPM wäre nur dann gewonnen, wenn die Prozessinformationen auch eingesammelt würden. Die Gefahr, dass wichtige fachliche Schritte von den operativen Systemen nicht gemeldet werden, wäre immer vorhanden.

Beim Testen wäre die automatisierte Durchführung der Tests nicht ohne Erweiterungen möglich. Wie beim klassischen BPM wäre beim Reverse BPM die Auswertung der Testergebnisse der End-to-End-Prozesse in einer Testumgebung innerhalb des rBPMS umsetzbar.

Das Ausführen der Startevents und der Human-Tasks für Tests müsste gesondert erfolgen. Letztendlich könnte man mit rBPM ein weiterhin vollumfängliches Business Process Management mit allen Vorteilen in der neuen agilen IT der Microservices einführen.

#### Links & Literatur

- [1] Slama, Dirk: „SOA und BPM wachsen zusammen“, in: Computerwoche 08.05.2007: <https://www.computerwoche.de/a/soa-und-bpm-wachsen-zusammen,1219234>
- [2] Fowler, Martin: „Microservices“, 25.03.2014: <https://martinfowler.com/articles/microservices.html>
- [3] McLarty, Matt: „Learn from SOA: 5 lessons for the microservices era“, in: InfoWorld 09.06.2016: <https://www.infoworld.com/article/3080611/application-development/learning-from-soa-5-lessons-for-the-microservices-era.html>
- [4] Conway, Mel: „Conway's Law“: [http://www.melconway.com/Home/Conways\\_Law.html](http://www.melconway.com/Home/Conways_Law.html)
- [5] Wolff, Eberhard: „Microservices“, dpunkt.verlag, 2015
- [6] Amundsen, Mike; McLarty, Matt; Mitra, Ronnie; Nadareishvili, Irakli: „Microservice Architecture“, O'Reilly, 2016
- [7] Self-contained Systems: <http://scs-architecture.org>
- [8] Wikipedia: „Operational Intelligence“: [https://en.wikipedia.org/wiki/Operational\\_intelligence](https://en.wikipedia.org/wiki/Operational_intelligence)
- [9] Wikipedia: „Business Activity Monitoring“: [https://en.wikipedia.org/wiki/Business\\_activity\\_monitoring](https://en.wikipedia.org/wiki/Business_activity_monitoring)
- [10] Camunda BPMN Engine: <https://camunda.com/products/bpmn-engine/>



#### Azmir Abdi

ist Wirtschaftsinformatiker und arbeitet als Software Architect bei sidion. Er ist Oracle-zertifizierter Java-Developer und JEE-Architekt. Seine Schwerpunkte liegen in Integration, BPM und Enterprise Architecture. Er hat mehrjährige Erfahrung als Integrationsarchitekt in einer SOA-Landschaft und arbeitet gerade in Microservices-Projekten. Sein

aktuelles Anliegen ist es, bei der IT Bewusstsein sowohl für Geschäftsprozesse als auch für Microservices-Architekturen zu schaffen.