

# The Path to operational Excellence: DevOps Erfahrung in Enterprise BPM

Sven Bernhardt, Benjamin Huskic  
OPITZ CONSULTING Deutschland GmbH

## Schlüsselworte

DevOps, Enterprise BPM, Oracle BPM Suite, Oracle Service Bus, Prozessautomatisierung

## Einleitung

Weltweit führende Unternehmen stehen vor der Herausforderung die Erwartungen der Endbenutzer an BPM Plattformen zu erfüllen. Zwei wesentliche Erfolgsfaktoren dabei sind ein transparenter Betrieb und möglichst kurze Reaktionszeiten im Fehlerfall.

In diesem Vortrag möchten wir mit Ihnen die Erfahrungen teilen, wie es ein deutscher Großkonzern geschafft hat, die Erwartung von 350.000 Benutzern zu erfüllen. Dabei werden wir den Mehrwert darstellen, welcher sich durch die enge Zusammenarbeit von Betrieb und Entwicklung ergibt. Dieser kann deutlich erhöht werden, indem zusätzlich der Hersteller, in unserem Fall Oracle, frühzeitig mit einbezogen wird.

Die größte Herausforderung des Projektteams war es Probleme in einer proaktiven Weise zu behandeln. Unterstützt durch Oracle gelang es dem Betriebs- und Entwicklungsteam Hand in Hand ein Rahmenwerk zu entwickeln, welches Möglichkeiten bot, fehlerhafte Zustände überschaubar und handhabbar darzustellen und in großen Teilen automatisiert zu beheben. Darüber hinaus half die Implementierung unterschiedliche Design-Patterns, die Prozesse robuster und wartbarer zu machen. Dieses Vorgehen führte zu einer signifikanten Erhöhung der Transparenz, einer deutlich reduzierten Anzahl von Fehlern und ermöglichte schließlich ein proaktives Monitoring sowie Fehlerhandling. Letztlich war dies der Schlüssel zum Erfolg und führte auf Kundenseite zu Zufriedenheit sowie Akzeptanz hinsichtlich der implementierten Prozessplattform.

## Ausgangssituation

Vom Mengengerüst her umfasst der Betrieb circa 30 Personalprozesse mit unterschiedlicher Komplexität, beginnend bei Self-Services für Mitarbeiter, die beispielsweise für die Änderungen ihrer Personalstammdaten verwendet werden können, bis hin zu komplexen Prozessen die von Führungskräften für deren Mitarbeiter angestoßen werden, zum Beispiel um standortübergreifende Mitarbeiterversetzungen abzuwickeln.

Das System bietet dabei deutschlandweit circa 140.000 Mitarbeitern die Möglichkeit, über ein entsprechendes Prozessportal an HR Prozessen teilzunehmen. Es werden pro Monat bis zu 40.000 Geschäftsvorgänge abgewickelt.

Aufgrund der hohen Menge von Prozessinstanzen entsteht unter anderem auch ein erwarteter Prozentsatz an fehlerhaften Instanzen, welche durch den Betrieb ausgewiesen und besonders behandelt werden müssen. Als Ursache für die auftretenden Fehler lassen sich die folgenden Hauptkategorien festhalten:

1. Implementierungsfehler während der Entwicklung
2. Fehlfunktionen der Prozess-Engine
3. Fehler in der fachlichen Modellierung
4. Infrastrukturell bedingte Fehler

Die Handhabung der Fehler stellt den Betrieb entsprechend vor unterschiedliche Herausforderungen.

### **Herausforderungen im Betrieb**

Im Folgenden möchten wir auf die zuvor erwähnten Kategorien eingehen und beleuchten welche Erfahrungen gemacht wurden und welche Schritte wir eingeleitet haben, um Effektiver mit den Herausforderungen umzugehen.

### **Implementierungsfehler während der Entwicklung**

Eine wichtige Grundeinstellung, wenn es um Implementierungsfehler geht ist, dass das alle Projektbeteiligten die Einstellung teilt, dass Fehler grundsätzlich etwas Gutes sind. Natürlich führen Fehler beim Kunden oder den Endbenutzern zunächst zu Verstimmungen, da beispielsweise eine notwendige Aktion nicht funktioniert oder ein Prozessschritt nicht beendet werden kann. Aber welche Software ist schon hundertprozentig fehlerfrei? Die Antwort: Keine! Fehler passieren, werden nicht absichtlich eingebaut und tragen im Nachgang zur Gesamtqualität einer Software bei. Deshalb ist es wichtig Kunden und Endbenutzer dahingehend zu sensibilisieren und bei auftretenden Fehlfunktionen diese zu dokumentieren, analysieren und schnellstmöglich zu beheben. Um dies sicherzustellen ist es notwendig, dass alle Stakeholder an einem Strang ziehen und das auf Schuldzuweisungen („Fingerpointing“) verzichtet wird. Schafft man es eine solche „Don't shoot the Pianist“-Kultur zu etablieren, können Probleme sachlich und in kürzester Zeit gelöst werden. Dies trägt dann zur allgemeinen Zufriedenheit sowie letztlich zur Robustheit und Akzeptanz einer Softwarelösung bei.

Bei Implementierungsfehlern geht es in erster Linie darum, möglichst zeitnah einen Fix einzuspielen. Dabei müssen Service- und Changemanagement beziehungsweise Entscheidungsträger im Allgemeinen ihre Flexibilität unter Beweis stellen und die organisatorischen Angelegenheiten für die Auslieferung klären. Die Etablierung und Kommunikation eines eindeutigen Changemanagement-Prozesses ist, der die entsprechenden Schritte und Verantwortlichkeiten sollte hierbei selbstverständlich sein.

Eine ebenso wichtige Rolle spielen Betrieb und Entwicklung. Von der Entwicklungsseite muss der Lebenszyklus der Anwendung so aufgestellt sein, dass ein Fix minimalinvasiv und ohne Auswirkungen auf nicht beteiligte Elemente der Anwendung bereitgestellt werden kann. Gleiches gilt auch für die Betriebsseite. Ein kritisches Update sollte sich ebenfalls mit geringstmöglichem Aufwand in das Produktivsystem bringen lassen. Dazu ist es unabdingbar die entsprechenden Vorbedingungen zu schaffen. Einfache Beispiele dafür wären auf Entwicklungsseite darauf zu Achten, dass in Composites sich häufig ändernde Komponenten wie z.B. Rules in ein eigenes Composite gegeben werden. Für den Betrieb muss es möglich sein, nur die auszutauschende Komponente und die direkten Beteiligten vom Netz nehmen zu können. Es ist schlichtweg fahrlässig mit der Annahme zu starten, dass es in Produktivumgebungen keine oder sehr wenig Fehler geben wird. Die Erfahrungen aus unterschiedlichsten Projekten zeigen Anderes. Daher ist es strategisch klug hinsichtlich der Flexibilität optimal aufgestellt zu sein.

### **Fehlfunktionen der Prozess-Engine**

Um Fehler in der Prozess-Engine zu adressieren ist es Notwendig, dass eine saubere Kommunikationsstruktur mit dem Hersteller aufgebaut ist. Bei Enterprise Projekten sollte der

Hersteller von Beginn an mit involviert sein um schnellstmöglich auf eventuelle Probleme in der Plattform reagieren zu können. Dies ist unbedingt notwendig, da es bei komplexen Enterprise-Produkten zu längeren Implementierungszeiten kommen kann.

Ein wichtiger Punkt von der Betriebsseite ist es, stets über vorliegende Patches sowie Updates informiert zu sein, um eventuellen Fehlfunktionen proaktiv vorbeugen zu können. Development und Betrieb müssen gemeinsam entscheiden, welche Patches für die eigene Landschaft notwendig sind. Wird die Entscheidung getroffen Patches einzuspielen, ist die Rolloutplanung gemäß dem definierten Changemanagement Prozesses durchzuführen.

Des Weiteren ist es relevant, dass klare Eskalationsstrukturen etabliert und in allen beteiligten Parteien, in unsrem Fall Kunde, Hersteller und Dienstleiter, klar kommuniziert sind. Abhängig von der Priorität eines Fehlers kann es notwendig sein, über die regulären ticketbasierten Eskalationsprozesse agieren zu müssen. Kommt es zu betriebskritischen Störungen oder sind Antwortzeiten von Service Requests (SR) zu lang beziehungsweise bleibt eine Lösung aus, kann die etablierte Eskalationsstruktur genutzt werden, um z.B. kurzfristig eine Task Force ins Leben zu rufen, welche sich dem Problem zeitnah und mit aller Aufmerksamkeit widmet. So können Spezialisten aus allen Lagern auf kurzen Dienstwegen zusammenarbeiten und das Problem lösen oder zumindest entschärfen. Existiert die angesprochene Eskalationsstruktur nicht beziehungsweise wurde versäumt diese zu etablieren, kommt es aufgrund der bestehenden Supportprozesse, die für den Regulärbetrieb definiert sind, zu hohen Zeitverzögerungen welche auf allen Seiten zu Anspannungen führen.

### **Fehler in der fachlichen Modellierung**

Fehler in der fachlichen Modellierung sind eher selten, ziehen sich aber wenn vorhanden unabwendbar über die Entwicklung bis in den Betrieb durch. Dies ist typisch für Fehler die sich bei der Aufnahme der Anforderungen beziehungsweise bei nicht dokumentierten Spezifikationsänderungen ergeben. In der Regel fallen diese auch nicht direkt auf, nach einer gewissen Laufzeit jedoch und nach Prüfung der ersten auftretenden Ungereimtheiten durch die Fachseite, werden diese aber dann aufgedeckt. In der Regel ist dann guter Rat teuer, denn die Beseitigung dieser fachlichen Differenzen ist meist nicht trivial. Fragen die sich bei der Feststellung von fachlichen Differenzen stellen sind:

- Welche Auswirkungen hat die fachliche Differenz? (eventuelle Compliance-Verletzung)
- Welches Gewicht haben die Auswirkungen? (Regel- oder Ausnahmefall)
- Was passiert mit bereits falsch gelaufenen Prozessinstanzen?

Wichtig ist an dieser Stelle, dass die fachliche Abweichung zeitnah behoben und dies dementsprechend nachvollziehbar dokumentiert wird. Dazu ist eine enge Kooperation von Fachbereich und Technik notwendig. Dies ist insbesondere in Großkonzernen eine Herausforderung, da dies in der Regel mit vielen Beteiligten und dementsprechend vielen Diskussionen zusammenhängt. Verantwortlichkeiten und Zuständigkeiten müssen, sofern diese klar sind, dabei beachtet werden. Aus diesem Grund ist es auch für die Beschleunigung der Ursachenbeseitigung in solchen Fällen ratsam, einen eindeutigen und an alle Beteiligten kommunizierten Ablauf zu definieren, der die entsprechend einzuhaltenden Schritte und Verantwortlichkeiten regelt.

In einer automatisierten Landschaft sollte es einem Nutzer nicht möglich sein fachliche Fehler zu verursachen. Dies bedeutet, dass die Implementierung alle fachlichen Möglichkeiten berücksichtigt und es nur zu technischen Fehlern kommen kann. Sollte sich dennoch Unzulänglichkeiten des Codes ergeben, sollten solche Fälle stets mit einem Change des Anwendungscodes versehen, um das Wiederholte auftreten dieses Fehlers für die Zukunft auszuschließen.

(In Form von Validierungen o. ä)

Sollte dies programmatisch nicht zu bewerkstelligen sein, müssen Prozesse in Zusammenarbeit mit dem Betrieb etabliert werden, welche die Vorgehensweise bei solchen Situationen ganz klar und eindeutig definieren.

### **Infrastrukturell bedingte Fehler**

BPM Projekte haben die Eigenschaft, dass die Einzelkomponenten in der Regel massiv verteilt sind, weil die Prozesse mit anderen Enterprise Systemen, beispielsweise dem ERP System, interagieren müssen. Die Fachlogik ist dabei, den Prinzipien service-orientierter Architekturen folgend, zumeist in kohäsive und voneinander unabhängige Services gekapselt, um im Falle von fachlichen Änderungen maximal flexibel zu sein und die Wiederverwendbarkeit der Fachlogik zu ermöglichen, was die Entwicklung neuer Prozesse beschleunigen kann. Auf der anderen Seite erhöht sich durch die verteilte Architektur auch die Komplexität der Gesamtlösung und stellt Entwicklung und Betrieb vor neue Herausforderungen, die es zu bedenken und beachten gilt.

In den Bereich der infrastrukturell bedingten Fehler, welche sich unweigerlich durch die verteilte Lösungsarchitektur ergeben, zählen:

- Timeouts von Transaktionen und Service-Calls
- Aufruffehler bedingt durch Netzwerkfehler oder Nicht-Verfügbarkeit von Systemen und Services
- Fehlkonfigurationen, wie falsche JVM Parameter oder zu kleine Connection Pools

Um solche Fehler zu vermeiden beziehungsweise im Fehlerfall diese schnellstmöglich auszumerzen ist eine enge Zusammenarbeit von Betrieb und Entwicklung notwendig; Im Idealfall bereits zur Entwicklungszeit.

Infrastrukturelle Fehler fallen vor allem in Hochlastszenarien beziehungsweise bei Lastspitzen ins Gewicht und können nicht nur Auswirkungen auf die Prozessplattform als solches haben, sondern vielmehr auch auf die entsprechend angebotenen Drittsysteme und Services. Um die Robustheit der Gesamtsystemlandschaft sicherzustellen ist es daher notwendig die erwarteten Mengengerüste gemeinsam mit dem Betrieb zu erörtern und mit diesen Erkenntnissen die involvierten Systeme hinsichtlich potentieller Schwachstellen zu analysieren. Klarheit darüber, ob entsprechende Maßnahmen notwendig sind, bringt allerspätestens dann ein entsprechend aufgesetzter Lasttest, der dringend erforderlich ist.

### **Mit Betrieb und Entwicklung gemeinsam zum Erfolg**

Bei Prozesslandschaften ist es vor allem zum Zeitpunkt der Produktivsetzung und auch noch die darauffolgenden Monate vorteilhaft, wenn der Betrieb und die Entwicklung eng zusammenarbeiten. Entscheidende Faktoren, um gemeinsam die neuen Herausforderungen bewältigen zu können, sind:

- kurze Kommunikationswege
- keine bürokratischen Hürden aufgrund von getrennten Abteilungen
- keine Kategorisierung von Probleme als Entwicklungs- oder Betriebsproblem; das hindert die Zusammenarbeit

Weiterhin ist es notwendig, dass der Betrieb von der Entwicklung eine High-Level Einführung über die zu betreuenden Prozesse erhält, um wesentliche Implementierungsdetails kennenzulernen und auch einen fachlichen Einblick in die entsprechenden Prozesse zu bekommen. Dies kann beispielsweise im Rahmen eines gemeinsamen Workshop erfolgen. Die Entwicklung für ihren Teil muss aber auch dazu angehalten werden Prozesse und deren involvierte Komponenten gut und auch für Außenstehende verständlich zu dokumentieren.

Der Betrieb muss seinerseits potentielle Probleme zeitnah an die Entwicklung kommunizieren, damit diese dann gemeinsam kategorisiert und behoben werden können. Damit ist nicht automatisch gewährleistet, dass der Betrieb reibungslos abläuft, man kann aber davon ausgehen, dass Lösungen wesentlich schneller gefunden werden, als wenn beide Bereiche autark voneinander arbeiten und nur über Mediatoren beziehungsweise gar nicht miteinander kommunizieren.

Ein weiterer wichtiger Punkt, dass ein einheitliches Verständnis über das Deployment-Vorgehen nach Änderungen an bestehenden Komponenten, aber auch bei der Bereitstellung neuer Komponenten und Prozesse vorhanden ist. Zudem muss im Vorfeld von anstehenden Deployments eine genaue Abstimmung von Entwicklung und Betrieb erfolgen. Dies gilt insbesondere dann, wenn Versionswechsel anstehen und in Verbindung damit die Migration von Altprozessen erforderlich ist.

## **Resümee**

Die wichtigste Erkenntnis ist, dass der Betrieb in Schieflage geraten kann, falls Implementierungs-, Prozess-Engine-, Infrastruktur und fachliche Fehler nicht von Beginn an adressiert werden. Durch die erhöhte Transparenz und die Zusammenarbeit sind die Zeiten für etwaige Fehlerfindung und Fehlerbehebungen stark herabgesetzt worden. Zudem ist durch die Zusammenarbeit aller Beteiligten die Qualität des Codes signifikant gestiegen, was zu einer Minimierung der Fälle in denen überhaupt agiert werden muss, deutlich gesenkt worden ist. Aus heutiger Sicht, läuft der Betrieb wie es alle Beteiligten von Beginn an erwartet haben. Man hat nun die Sicherheit, dass bei ungewöhnlichen Wahrnehmungen von Fehlern, eine Fehlfunktion des Systems unwahrscheinlich ist. Es muss hin und wieder mal ein Fehler behoben, was der Erwartungshaltung entspricht und der Endanwender überhaupt nicht mehr mitbekommt. Sowohl der Kunde, als auch die Kollegen vom Betrieb sind zufrieden, wozu vor allem die Transparenz über alle Ebenen bezüglich des Zustands der Applikation beigetragen hat.

## **Kontaktadressen:**

Sven Bernhardt  
OPITZ CONSULTING Deutschland GmbH  
Standort Gummersbach  
Kirchstrasse 6  
D-51647 Gummersbach

Telefon: +49 (0) 2261 6001 - 0  
Fax: +49 (0) 2261 6001 – 4200  
Mobil: +49 (0) 172 2193529  
E-Mail: sven.bernhardt@opitz-consulting.com  
Internet: www.opitz-consulting.com

Benjamin Huskic  
OPITZ CONSULTING Deutschland GmbH  
Standort Hamburg  
Butendeichsweg 2  
D- 21129 Hamburg

Telefon: +49 (0) 40 741122-0  
Fax: +49 (0) 40 741122-4300  
Mobil: +49 (0) 173 7235492  
E-Mail: [benjamin.huskic@opitz-consulting.com](mailto:benjamin.huskic@opitz-consulting.com)  
Internet: [www.opitz-consulting.com](http://www.opitz-consulting.com)