

Agile BI in der Praxis

DevOps4BI

Andreas Ballenthin

OPITZ CONSULTING Deutschland GmbH

Schlüsselworte:

Agile BI, SCRUM, Automatisierte Testen, DevOps, DevOps4BI

Einleitung

Traditionelle Vorgehensweisen, die eine strikte Trennung zwischen dem Entwickeln und dem Betreiben einer Plattform vorsehen, sind insbesondere bei der Anwendung agiler Vorgehensweisen nicht praktikabel. Die DevOps-Bewegung versucht, die Zusammenarbeit zwischen Entwicklung und Betrieb zu intensivieren und den Aufbau von Vertrauen und Transparenz auf beiden Seiten zu fördern. Der Weg führt über ein höheres Interesse füreinander und den regelmäßigen Austausch zu Ideen und Problemen im Projektalltag.

Wie sehe ich DevOps4BI?

Die Praxis hat gezeigt, dass das Zusammenspiel zwischen Entwicklung und IT-Betrieb im agilen Umfeld hauptsächlich durch gemeinsame Arbeitsweisen und Interessenslagen bestimmt wird. In vielen Kundenprojekten führte die reibungslose Zusammenarbeit an einer Fragestellung zu einer Lösung, welche die Teams mit gemeinsamem Augenmaß im Sinne des Kunden etablieren konnten. Die konsequente Zusammenarbeit, das erhöhte Verständnis füreinander und gemeinsame Problemlösungsprozesse sind der Kern der DevOps-Bewegung. Ich sehe DevOps, insbesondere DevOps4BI, als konsequente Fortsetzung einer agilen Entwicklung.

Um das zu verdeutlichen, betrachte ich ein Erfahrungen aus einem bereits 2 ½ Jahre andauernden Projekt, bei dem die Entwickler und das IT-Betriebsteam des Kunden in großer räumlicher Distanz vom System-IT-Betriebsteam sitzen. Zudem gehören die Mitarbeiter des System-IT-Betriebs einer anderen Tochtergesellschaft an. Es gibt also drei "Fraktionen" mit unterschiedlichen Zielsetzungen:

1. Das agile Entwicklungsteam möchte die Sprintergebnisse ausrollen und damit das Produktivsystem ändern.
2. Das System-IT-Betriebsteam möchte ähnlich wie das Kunden-IT-Betriebsteam ein stabiles Produktivsystem betreiben.
3. Das Kunden-IT-Betriebsteam betreibt hauptsächlich operative Systeme und hat für die Betreuung des Data Warehouse (DWH) nur wenig Zeit.

Der Schulterchluss zwischen Fachbereichen und Entwicklungsteam wird über agile Methoden weitestgehend erreicht. Die "Mauer" zwischen Entwicklungsteam und System- IT-Betriebsteam, wie in Abbildung 1 gezeigt, überwinde ich mit dem Leben des DevOps-Gedankens. Wenn wir uns nun die in Abbildung 2 gezeigte Erfahrungsverteilung zwischen Entwicklungsteam, Kunden-IT-Betriebsteam und System-IT-Betriebsteam im Kundenszenario anschauen, fällt folgendes auf:

1. Die Fähigkeiten für die Administration von ETL (hier: Informatica), der semantischen Schicht und des Frontends (hier: Business Objects) liegen innerhalb des Entwicklungsteams und sind dort auf mehrere Personen verteilt. Dieser Umstand ist nicht spezifisch für dieses Szenario, sondern kommt in Projekten regelmäßig vor. Das System-IT-Betriebsteam ist also für sich allein nur begrenzt in der Lage, diese komplexen Werkzeuge system- oder anwendungsseitig zu betreuen.
2. Das Entwicklungsteam verfügt nur über geringe Kenntnisse im Infrastrukturmilieu, wie in Storage-Systemen oder in der Netzwerk- und Firewall-Administration. Hier gilt es, durch geschicktes Fragen, Hinterfragen oder schlicht durch verstärktes Interesse zu Erkenntnissen zu gelangen.
3. Die Datenbankadministration, vor allem die Administration von Entwicklungs- und Testsystemen, stellt sowohl eine gemeinsame als auch eine geteilte Aufgabe dar. Die Erfahrung zeigt: Nach einigen Verhandlungen haben wir als Entwicklungsteam uns auf den maximalen Rechtfertigungsbereich für das Entwicklungsteam geeinigt.
4. Das Kunden-IT-Betriebsteam betreut hauptsächlich andere, operative Systeme und leidet unter einer chronischen Überlast. Dank intensiver Zusammenarbeit haben wir als Entwicklungsteam letztlich die Zustimmung erhalten, nur noch grundlegende Änderungen im DWH im Detail besprechen zu müssen. Heute kann das Entwicklungsteam im Regelbetrieb anfallende Eingriffe in das Produktivsystem und alle Vorarbeiten zu Betriebsübergaben selbst erledigen.

Automatisierte Deployments – Datenbankdeployments

Um die Datenbankinhalte auf den vom Entwicklungsteam gewünschten Stand zu bringen, kommt die hausinterne Schemaverwaltung zum Einsatz. Dieses auf "Ant" basierende Werkzeug vergleicht den Ist-Zustand einer Oracle-Datenbank mit dem in Skripten definierten Soll-Zustand. Sofern das Tool Unterschiede feststellt, führt es die erforderlichen DDL-, DML- und DCL-Befehle aus (Data Definition Language; Data Manipulation Language; Data[base] Control Language).

Das System-IT-Betriebsteam wünschte sich ein einziges, ablauffähiges Skript für das Deployment der Datenbankartefakte. Zu diesem Zweck verfeinerte das Entwicklungsteam die Schemaverwaltung so weiter, dass sie pro Datenbankbenutzer eine verkettete Datei aller notwendigen DDL-, DML-, DCL- und Datenmigrationsskripte auf einem Lieferserver bereitstellt. Im Gegenzug zeigte sich das System-IT-Betriebsteam bereit, die Algorithmen des Angleichens vom Soll- an den Ist-Zustand zu übernehmen.

Auf das Testsystem wird täglich ausgerollt, in das Abnahme- und später in das Produktivsystem derzeit etwa monatlich. Da das Deployment täglicher Inkremente in Einzelfällen ein anderes

Ergebnis liefert als ein Release- Inkrement, setzt das Entwicklungsteam das System auf den erfolgreich getesteten Daten- und DDL-Stand des vorigen Release zurück und installiert darauf das Release-Inkrement. Dies ermöglicht den beteiligten Teams das produktionsäquivalente Testen.

Automatisierte Deployments – ETL-Deployments

Die Inbetriebnahme der ETL-Prozesse führte das Unternehmen am Anfang des Projekts manuell durch. Das Entwicklungsteam erzeugte ein ETL-Paket und stellte es dem System-IT-Betriebsteam zur Verfügung. Das System-IT-Betriebsteam spielte das Paket daraufhin manuell auf der Zielumgebung ein. Das Client-Tool für das Deployment stellte sich jedoch als sehr leistungsschwach und fehleranfällig heraus, da während des Prozesses viele Interaktionen anfielen. Um ETL-Deployments schneller und sicherer durchzuführen, implementierte das Entwicklungsteam eine Kombination aus Skripten und ETL-Tool-spezifischen Ablaufsteuerungselementen. Eine solche Kombination ermöglicht nicht nur eine automatisierte Inbetriebnahme als Grundvoraussetzung für automatisierte Backend-Tests, sondern erlaubt es dem System-IT-Betriebsteam, ein Deployment mit einem einzigen Befehl durchzuführen.

Gemeinsame Administration von Entwicklungs- und Testsystemen

Für das Oracle RDBMS (Relational Database Management System) mit DWH-Schemata und ETL-Repository: Das System-IT-Betriebsteam hat SYSDBA-Rechte (Database System Administrator – höchstprivilegiertes Datenbankrecht sowie Root-Rechte auf den Datenbank- VMs (virtuelle Maschinen). Das Entwicklungsteam hat die DBA-Rolle sowie einen Betriebssystemzugang zu den Datenbank-VMs außerhalb der Gruppen oracle/oracledba. Dazu gehört auch, dass das System-IT-Betriebsteam beispielsweise für das Patching des Oracle RDBMS verantwortlich ist und das Wann und Wo in Absprache mit den anderen Teams durchführt. Systemrechtevergaben an User des DWH-Schemas werden im Vorfeld von System- IT-Betriebsteam und Entwicklungsteam hinreichend besprochen und sofort bis zum Produktivsystem ausgerollt.

Im Bereich ETL hat das Entwicklungsteam volle Administratorenrechte im ETL-Werkzeug und den Betriebssystemzugang als ETL-Eigentümer. Sachverhalte, die die Kolleginnen und Kollegen auch im Abnahme- und Produktivsystem durchführen, die sie aber nicht sinnvoll automatisieren können, bereitet das Entwicklungsteam vor. Das System-IT-Betriebsteam übernimmt diese später hoheitlich. Ein Beispiel wäre die Verschiebung des ETL-Repositorys aus einer separaten Oracle-RDBMS-Instanz in die DWH-Instanz. Das System- IT-Betriebsteam migriert die Hotfix-, Entwicklungs- und Testsysteme noch mit Hilfe des Entwicklungsteams, während es das Abnahme- und Produktivsystem später in alleiniger Zuständigkeit überführt.

Bei der Storage- und Netzwerkadministration verfügt das Entwicklungsteam nur über geringe Kenntnisse. Deshalb trägt das System-IT-Betriebsteam hier den überwiegenden Teil der Verantwortung und insbesondere Innovationen ein. Das Backup der Umgebungen gestaltete sich aufgrund des zu sichernden Datenvolumens und der Änderungsraten der Daten immer schwieriger. Eine reine Sicherung mit dem Recovery Manager (RMAN) konnte nicht mehr sinnvoll durchgeführt werden. Das Speichersystem bietet jedoch die Möglichkeit, innerhalb von Minuten einen Snapshot des Speicherumfangs zu erstellen. Mit Hilfe der Snapshots ist es daher möglich, in 15 Minuten einen Restore und Recovery durchzuführen.

Bei den Front-End-Systemen hat das Entwicklungsteam ebenfalls volle Administratorenrechte im Business- Objects-Werkzeug und den Betriebssystemzugang als Eigentümer.

Die gemeinsame Administration wird besonders anspruchsvoll, wenn die Teams ein Release-Inkrement im Testsystem vollständig testen müssen. Das Zurückrollen auf den Code und den Datenstand des letzten Release und das Deployment und die Verarbeitung zum initialen Release belasten alle Systemressourcen (Oracle RDBMS, ETL-Server und Storage-Systeme) so stark, dass die Teams die Systeme gemeinsam „monitoren“ sowie bei Auffälligkeiten zügig reagieren sollten.

Betreuung von Produktivsystemen

Insbesondere bei der Betreuung von Produktivsystemen wird deutlich, wie wichtig ein reibungsloses Zusammenspiel von Entwicklungsteam, System-IT-Betriebsteam und Kunden- IT-Betriebsteam ist. Die heterogene Systemlandschaft und das stetige Wachstum des Systemumfangs machen zunächst ein funktionierendes Monitoring unabdingbar. Zu diesem Zweck haben wir als Entwicklungsteam unterschiedliche Maßnahmen konzipiert und implementiert:

- Echtzeit-Hardware-Monitoring der Systeme, bereitgestellt durch SysOps (CPU, RAM, Netzwerkbelastung, Storage etc.)
- Versendung von E-Mails zur Nutzung von Tablespaces und verfügbarem Storage
- Versendung von E-Mails bei Erfolg beziehungsweise Misserfolg der ETL-Prozesse, unterteilt nach Schnittstellen
- Monitoring-Berichte im Frontend (Übersicht Dateischnittstellen, Status der Ladeprozesse)

Es ist nur selten notwendig, dass das Team Probleme vor der nächsten Inbetriebnahme direkt beheben muss. Diese Art von Eingriffen bezieht sich auf einen begrenzten Kontext, bei dem das Eingriffsrisiko als sehr gering eingeschätzt wird. Dazu gehören konfigurative Eingriffe wie:

- Änderungen in Steuertabellen
- Anlegen eines zusätzlichen Index, beispielsweise zur Unterstützung eines Anfrageverhaltens, das vom Product Owner oder vom Entwicklungsteam nicht erwartet wurde
- Detailkonfigurationen in ETL-Mappings oder ETL-Sessions, wie Größen von Lookups
- Änderungen der Scheduling der ETL-Strecken, beispielsweise bei Überbelastungen in der Nacht oder bei vorgesehenen Wartungsfenstern

Für alle Arten konfigurativer Eingriffe gilt, dass die Betriebsverantwortlichen sie mit dem nächsten Deployment des Systems bestätigen und damit alle Systeme gemeinsam rekonfigurieren müssen. So schließt sich der Kreis und die Teams erreichen das gemeinsame Ziel: eine mit Augenmaß im Sinne des Kunden beziehungsweise des Unternehmens etablierte Lösung.

Ich habe persönlich erlebt, wie wichtig ein Abbau von Hindernissen zwischen Entwicklungsteams und Betriebsteams für das Gelingen von DWH-Projekten sein kann. Agile Methoden erfordern von allen Beteiligten ein hohes Maß an Flexibilität, nur durch die reibungslose, teamübergreifende

Zusammenarbeit erreicht wird. Letztendlich kommt diese nicht nur dem Projektziel zugute, sondern auch dem Ausbau des persönlichen Erfahrungsschatzes jedes Einzelnen.

Kontaktadresse:

Andreas Ballenthin
OPITZ CONSULTING Gummersbach GmbH
Kirchstraße 6
D-51647 Gummersbach

andreas.ballenthin@opitz-consulting.com

Telefon: +49 (0) 2261-6001-0
Fax: +49 (0) 2261-6001-4000
Internet: www.opitz-consulting.com