

Oracle APEX als Strategie

René Schneider
DB Systel GmbH
Frankfurt am Main

Schlüsselworte

APEX, Strategie, Best Practices, Vorgehensmodell, FraDD

Einleitung

Die hohe Geschwindigkeit in der sich heute Geschäftsprozesse verändern oder angepasst werden, bringt vor allem für die IT viele Herausforderungen mit sich. IT - Abteilungen können meistens dem Tempo der Fachbereiche nicht mehr schritthalten, was sie meistens zur Eigeninitiative zwingt. Hierbei wird schnell auf bekannte Bürosoftware wie Excel etc. zurückgegriffen. Mit Fokus auf die schnelle Lösung, werden schnell Themen wie Compliance, Corporate Design etc. vernachlässigt. Aktuelle Auswertung belegen, dass aktuell 2/3 der Prozesse auf selbst entwickelte Software basieren[1][2].

Dem Trend entgegen wirken

Das Ziel war klar. Es sollte eine Lösung geschaffen werden, welche den heutigen Fachabteilungen eine performante Alternative bietet. Der Scope wurde so festgelegt, das sich diese zukünftige Abteilung ausschließlich nur mit kleinen Anwendungen (Abteilungsbezeichnung: Small Solution) in einem definierten Kostenrahmen bewegt, um genau den bekannten Problemsektor der Microsoftwelt adressieren zu können. Große Prozessketten und bürokratische Schlachten sollten durch pragmatische/simple Ansätze vermieden werden. Dabei sollte aber auch darauf geachtet werden, dass neu entwickelte Anwendungen nach mehreren Jahren nicht den gleichen Stand erreichen, wie sie viele heute als "historisch gewachsene" Anwendung kennen. Ergo es sollten Standards geschaffen werden. Hierbei stellten sich mehrere Herausforderungen.

- Welche Technologie soll eingesetzt werden?
- Wie können die Konzernrichtlinien eingehalten werden?
- Wie organisieren wir die Entwicklung?
- Welches Mitarbeiterprofil benötigen wir?
- Weitere Themen waren Maintainance, Qualitätsmanagement, Dokumentation, LifeCycle etc.

Alle dieses Themen mussten berücksichtigt werden. Hauptanforderung war es ein System zu wählen, welches zentralisiert im RZ betrieben werden kann. Hierbei standen mehrere Lösungen zur Auswahl, wo nach mehreren Runden die Wahl auf Oracle APEX viel. Hauptgründe war die fast ausschließlich deklarative Programmierung, Export/Importfunktionen, Validierungsfunktionen, einfacher Betrieb und technische Architektur.

Die meisten Konzernrichtlinien, speziell Datenschutzthemen konnten durch die benannte zentrale Architektur durch Platzierung in einem entsprechenden Rechenzentrum gelöst werden. Themen wie Corporate Design, Passwortrichtlinien und jegliche weiteren Anforderungen innerhalb einer Anwendung wurde mit Hilfe eines Referenztemplates (packaged application) gelöst. Was heißt, es wurde eine APEX Anwendung entwickelt, welche alle im Vorfeld schon bekannten Richtlinien, sowie vordefinierten Funktionen enthält und auf Knopfdruck aufgespielt werden kann. Mit vordefinierten Funktionen ist zum Beispiel die vorgefertigte Benutzerverwaltung gemeint, welche unabhängig vom APEX Workspace ist usw.

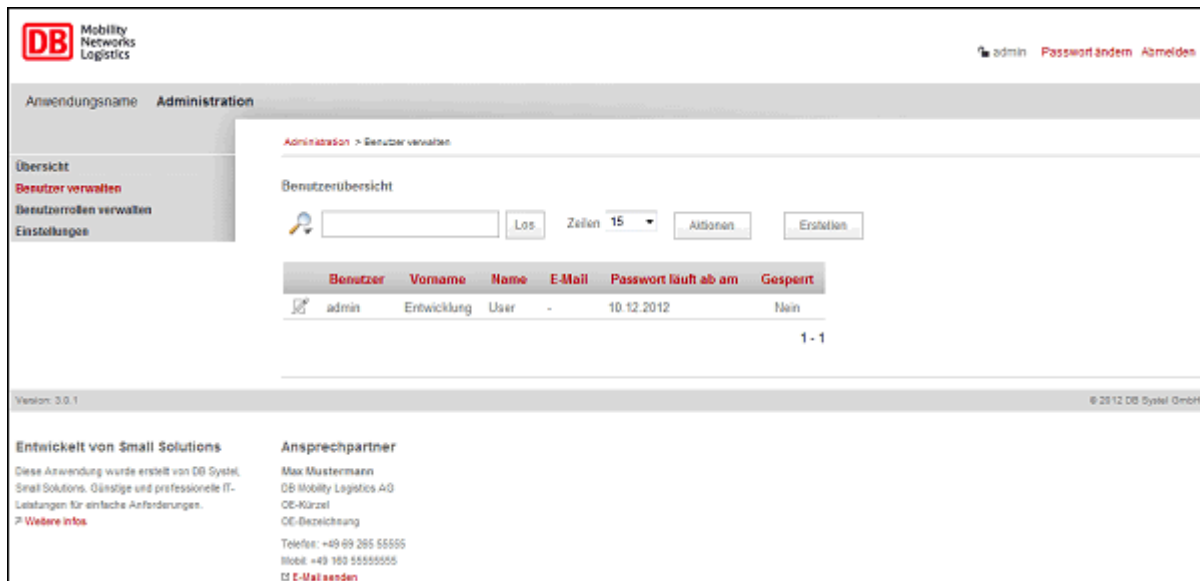


Abb.1 – Referenztemplate in der aktuellsten Version

Die nächste Herausforderung, war die Methodik/Vorgehensweise, wie wir zukünftig Anwendungen entwickeln. Hier kommt hauptsächlich der junge Rahmenmodell FraDD (Framework Driven Development) zum Einsatz. Dabei geht es darum, Anforderungen des Kunden auf vorhandene Features einer Technologie zu mappen bzw. weiche Anforderungen und Ideen prototypisch zu konkretisieren. Das heißt, für eine bestimmte Technologie wird eine Liste von Features erstellt, welche als Basis der Diskussionen für Anforderungsgespräche mit dem Kunden dienen. Sollte es Anforderungen geben die nicht einfach mit Features oder die Kombination von Features abgedeckt werden kann, ist die Aufgabe des Beraters zu schauen, wie Anforderungen angepasst/gesplittet werden, so dass sie doch wieder gemappt werden können. Oder sie werden aus Komplexitäts-/Kostengründen komplett weggelassen.

Ein einfaches Beispiel im APEX Umfeld sieht so aus, dass der Kunde von einer bestimmten Datentabelle eine Auswertung haben möchte, wo er sortieren kann und ggf. als Mail verschicken kann. Hierbei wäre der bekannte Interaktiv Report die richtige Wahl, was mit 5 Mausklicks und Eingabe eines Selects erledigt ist.

Da das Umfeld der Anforderungsanalyse zu Beginn meistens nur Ideen und Visionen bringt, soll mit Hilfe kurzfristig lauffähiger Prototypen die Vorstellungen geschärft werden. Prototypisches Vorgehen heißt, in einer bestimmten Iteration dem Anforderer/Ideengeber eine lauffähige Version zur Verfügung zu stellen und so gestellte Anforderungen zu verfeinern und auszuarbeiten. Dies bietet allen Parteien die Möglichkeit ein Verständnis für Problemstellungen zu entwickeln und die o.g. Vorgehensweise zu unterstützen.

In der Diskussion, in welchem Umfang das Team aufbaut werden soll, kamen wir schnell auf den Entschluss, dass hier ein breites Spektrum an Fähigkeiten vorausgesetzt ist. Zum einen jemand der technisch Erfahrung im Bereich der Anwendungsentwicklung, Test und Qualitätsmanagement mitbringt, also auch jemand, der fachliche Themen des Kunden verstehen kann, dementsprechend beraten und final das Projekt auch leiten kann. Ergo Generalisten statt Spezialisten. Um zeitliche Aspekte solcher Projekte zu optimieren, sollten Projekte max. aus 1 – 2 Mitarbeitern bestehen, um Fragen wie „Was setze ich Wie um“ immer zur richtigen Zeit beantworten werden können.

Rahmenbedingungen

Bekanntlich ist es so, dass wenn man dem Anforderer den kleinen Finger reicht, letzten endlich der ganze Arm gefordert wird. Um den Spielraum für die neuen Anforderungen während des Projektes einzugrenzen und nicht das eigentliche Ziel aus den Augen zu verlieren, wurden Leitplanken geschaffen, welche sich aus den definierten Features der Technologie (hier APEX) geschaffen haben.z.B.

- Keine autom. Schnittstellen ausser WebServices
- Kein Filesystemzugriff
- Niedrigstes Service level
- Wenig Komplexität
- Keine Offlinefähigkeit
- Kein Export/Import von formatierten Office Dokumenten
- Look&Feel wird vom Corporate Design vorgegeben

Erfahrungen sammeln

Nach Entscheidung, Ausarbeitung und Planung der ersten Punkte sollte nun erste Erfahrungen gesammelt werden. Um gemachte Erfahrungen langfristig zu konsolidieren, wurde festgelegt alle 3 Monate ein Rückblick auf Projektentwicklungen und Prozesse zu setzen. Hier sollte auch immer wieder reflektiert werden, dass der grundsätzliche pragmatische Ansatz (Simplify) erhalten bleibt.

Die größte Herausforderung der ersten Projekte war das bekannte Bewusstsein des Anforderers. "Ich bekomme alles so, wie es mir gerade in den Sinn kommt. Aber vielleicht sieht es nächste Woche schon anders aus." Hier musste zuerst ein Wandel beim Kunden stattfinden, dass zum Beispiel im letzten Meeting besprochene Ideen schon in einer lauffähigen Anwendung zu sehen waren. Der Effekt des prototypischen Vorgehens plus die transparente Diskussionen, Anforderungen auf Fähigkeiten von APEX zu mappen, war erstaunlich positiv.

Nach dem ersten halben Jahr und rund 20 fertigen Projekten verteilt auf verschiedenen Oracle Datenbanken kamen langsam Themen wie LifeCycle und Maintainance auf dem Plan.

Die LifeCycle-Themen kamen daher, inzwischen gab es eine neue Version von Oracle, der Betrieb möchte auf die neue Version, die ersten produktiven Anwendungen laufen jedoch noch auf der alten Version usw. Auch wenn man Erfahrungsgemäß weiß, dass alle APEX Versionen abwärtskompatibel sind, wurde hier auf Nummer sicher gegangen und sich auf zwei parallel laufende Versionen beschränkt. Dabei ist APEX selbst nicht immer die einzige Komponente, welche in die LifeCycle Betrachtung fällt. Addons (z.B. JavaScript Frameworks) werden schnell Teil der LifeCycle Diskussion. Wie kann man von ausgehen, dass Sourcen aus dem Internet wirklich weiterentwickelt werden. So kann man in diesem Bereich festhalten, bewusst darauf zu achten, wenn man Komponenten aus dem Internet einsetzt und sich bewusst auf einige wenige Addons beschränkt.

Im Bereich Maintainance standen Themen wie Dokumentation und Weiterentwicklung von bestehenden Verfahren. Durch die Vielzahl an neuen Projektaufträgen, steigender Zahl an Mitarbeitern, sowie der benannten generellen Anforderungen keine historisch gewachsenen Anwendung zu wiederholen, standen wir vor der Herausforderung der Weiterentwicklung einer Anwendung. Es musste die Möglichkeit gegeben sein, durch einen beliebigen Mitarbeiter der Abteilung das Projekt jeder Zeit weiterführen zu können. Einfache Aufgaben wie Zugangsdaten, sowie zentrale Sammlung aller bekannten/entstandenen Dokumente sowie Sammlung der Mails gehören zu den üblichen Sachen. Herausfordernder wird die Dokumentation der Anwendung. Durch die Struktur

von APEX Technologie, der ausschließliche Einsatz von definierten Features und Tools wie APEXlib (PageFlowDiagramm) plus Aufbau des Application Builders, bietet die APEX Entwicklungsumgebung eine Übersicht, welche durch jeden APEX erfahrenden Kollegen ohne Probleme verstanden werden kann. Darauf aufbauend wurde z.B. festgelegt bestimmte Namenskonventionen (z.B. Benennung eines Prozesse) einzuhalten und wenn möglich nur automatisch generierte Sourcen zu verwenden. So haben wir aus unserer Sicht einen Satz von Informationen geschaffen, um eine entsprechende schnelle Weiterentwicklung zu ermöglichen.

Nach 90 Projekten, ein volles Jahr später und eingespielter Vorgehensweise in der Abteilung kamen nach den ersten Lessons learned Workshops immer das Qualitätsmanagement zur Sprache.

Wer kennt es nicht, wenn man kurz vor dem Ende einer Entwicklung steht. Der Anforderer möchte nur noch Live gehen und immer wieder werden in letzter Minute Fehler gefunden, welche eigentlich schon gefixt waren. Dann wird endlich die Anwendung live geschaltet und doch kommen immer noch nachträglich Fehler auf. Um diesen Aufwand zu minimieren kamen wir eigentlich auf 2 Maßnahmen.

Der Einsatz von APEX Bordmitteln (APEX Advisor), welche bereits Kleinigkeiten, wie unbenutzter Code oder invaliden Statements usw. erkennt und so bereits technische Mängel behoben werden können. Für fachliche Themen wurde sich auf eine bekannte Lösung Selenium festgelegt, welche zum Beispiel auch für die Entwicklung von APEX selbst verwendet wird. Hiermit war es möglich auch fachliche Situation nachzustellen und automatisiert beliebig oft auszuführen. So besteht nun auch die Möglichkeit Tests des Anwenders selbst aufnehmen zu lassen und so die Zeitaufwände für Entwürfe von Testaufwänden in der Entwicklung zu sparen.

Zukünftige Herausforderungen

Nach zahlreichen internen Projekten stehen wir nun bei externen Kunden gerade beim Thema Zugriff aus dem Internet vor einer neuen Herausforderungen. Hierbei werden Themen wie automatische Penetrationstests, automatische Generierung von Sicherheitsregeln (mod_security WhiteLists) usw. betrachtet.

Viele weitere Themen, Provisioning, Verrechnung, Outsourcing, Migration von Office Anwendungen, etc. sind nur einige weitere Herausforderung, welche jedoch den Rahmen sprengen würden.

Resümee

Nach 2 Jahren und mehr als 100 Kleinprojekten haben sich einige Punkte herausgezeigt, wie man die Entwicklung mit einem rapid development Anwendung wie APEX und sein Umfeld optimieren kann. APEX in seinem vollen Umfang zeigt, dass es nicht nur als Entwicklungstool sondern durch seine Struktur und Begebenheiten auch als Guideline für aktuelle Herausforderungen in einem IT-Unternehmen dienen kann.

Kontaktadresse:

René Schneider
DB Systel GmbH
Jürgen-Ponto-Platz 1,
60329 Frankfurt

Telefon: +49 (0) 175 9328497
E-Mail: rene.schneider@deutschebahn.de
Internet: https://www.xing.com/profile/Rene_Schneider64

Quellen

- [1] <http://www.heise.de/ix/meldung/IT-verliert-Kontrolle-ueber-Geschaeftsprozesse-1244454.html>
- [2] <http://www.cio.de/strategien/2887431/>