

Zwischenbilanz eines Oracle ADF basierten Softwareentwicklungsprojektes im Raumfahrt Umfeld

Dipl. Inform. Robert Schweser
megatel GmbH
Bremen

Schlüsselworte

Oracle ADF 11g und 12c, JDeveloper 12c, ADFdi, ADF Desktop Integration

Einleitung

Halbzeitstand eines ADF Softwareentwicklungsprojektes im Raumfahrt Umfeld. Dieser Vortrag ist sowohl ein praktischer Erfahrungsbericht als auch eine Retrospektive über ADF als Entwicklungsframework. Nach gut 2 Jahren Entwicklung einer Satelliten Referenzdatenbank als Enterprise Anwendung, die als unternehmensweite zentrale Plattform für die Satelliten-Entwicklung als Referenz zur Sicherung von Satelliten-Konfigurationsständen dienen soll, wird eine Zwischenbilanz gezogen.

Das Kernentwicklerteam betritt die Welt der ADF Entwicklung. Es etablieren sich Vorgehensweisen, Umsetzungsstrategien, das Verständnis von ADF und dem JDeveloper als Tool wächst, neue und bessere Lösungswege werden gefunden. Das ADF Framework schenkt einerseits viele Funktionen, fordert aber auch andererseits einiges vom Entwickler ein.

Zielgruppe:

- Software Entwickler und Architekten
- Technische Führungspersonen
- Projektplaner und -manager
- Vom Einsteiger bis zum Erfahrenen

Was bietet dieser Vortrag?

- Einen Einblick in unsere persönliche Erfahrung mit Oracle ADF
- Unsere Software Development War Story mit ADF als Entwicklungsframework
- Unsere Lösungswege und Herangehensweisen werden beschrieben
- Erfahrungsaustausch
- Mitnahme der einen oder anderen guten Idee (Best Practice)
- Ideengeber für die eigene Planung eines ADF Projektes

Das Projektumfeld – Unternehmenskontext

- Die OHB AG ist ein europäischer Raumfahrt- und Technologiekonzern und eine der bedeutenden unabhängigen Kräfte der europäischen Luft- und Raumfahrtindustrie.
- Mit ihrer über 30-jährigen Erfahrung in der Entwicklung und der Umsetzung innovativer Raumfahrtsysteme und -strukturen, dem Angebot von spezifischen Luft-, Raumfahrt- und Telematikprodukten hat sich die OHB-Gruppe herausragend positioniert und für den internationalen Wettbewerb aufgestellt.
- Die Wertschöpfungskette reicht von Satellitenentwicklung, -bau und -betrieb, über Datenübertragung und -verarbeitung, die Entwicklung und den Bau wissenschaftlicher Nutzlasten bis hin zu Strukturen für die Luft- und Raumfahrt.
- Die megatel GmbH als Teil der Aerospace + Industrial Products Sparte der OHB AG bietet IT Consulting, Datenbanklösungen, Projektmanagement, Softwareentwicklung, IT Services als Dienstleistung und managed die IT Infrastruktur des Konzerns und seit über 25 Jahren auch Oracle Partner

- Eigene Projekte wurden bereits mit ADF als Framework durchgeführt im Raumfahrt Umfeld allerdings noch nie

Projektrahmen – Aufgabenstellung

- Teil im Gesamtprojekt Meteosat Third Generation (MTG)
- Kunde/Auftraggeber: EUMETSAT / ESA
- Ziel: Verbesserte Wettervorhersage durch neue Techniken der Bildaufnahme
- Programm besteht aus 4 Imager-Satelliten und 2 Sounder-Satelliten
- OHB System in Bremen verantwortlich für beide Sounder-Satelliten und vier weitere Plattformen für die Imager-Satelliten
- OHB System in München ist Hauptauftragnehmer für die IRS-Nutzlast und größter Unterauftragnehmer für das FCI-TA.
- Betriebsphase des gesamten MTG-Programms für 2017 bis 2037 vorgesehen (Starts ab 2017)
- Spezialbereich hier: Satellitenkommunikation
 - Datenaustausch via Telekommandos von der Bodenstation, Telemetrie Daten vom Satelliten, Nutzdaten vom Satelliten
 - Komplexe Protokolle und Inhalte
 - Es muss gesichert sein das Kommandos passend sind für die Satellitenkomponenten und deren Konfiguration
 - Es muss gesichert sein das die Rückläufigen Telemetrie- und Nutzdaten verstanden werden
- Absicherung erfolgt durch ausgiebige Simulation und Tests am Boden
 - Getestet werden alle Telekommandos und Telemetrie Formate
 - Simulation und Hardwaretests
 - Abgleich mit den erwarteten Daten
 - Validierung der Steuerdateien
 - Einsatz spezieller Ausrüstung wie Electrical Ground Support Equipment (EGSE)
 - Dazu Rückgriff auf die Daten der Ingenieure, die die Modulgruppen, Komponenten und System Elemente erstellen
- Zusätzlich gibt es bei der Satellitenentwicklung unterschiedliche Lebenszyklen:
 - Definition: „Auf Papier“ vordefiniert und als Vorlage für das Electrical Engineering
 - Realisation: Electrical Engineering und Konstruktion, die praktisch Umsetzung
 - Released: Der eingefrorene Zustand für ausgelieferten Satelliten
- Benötigt wird: Satellite Reference Database (SRDB)
 - Zentrale Sammelstelle und Referenz für die Entwicklung
 - Speichern aller Lifecycles
 - Verwalten aller Parametern der System Elemente, Komponenten und Module eines Satelliten
 - Support für Datenpopulation
 - Telekommandos und Telemetrie Referenz
 - Erzeugen von Steuerdateien für die Tests
 - Change Management
- SRDB als Enterprise Tool
 - Erweiterbares, flexibles Design
 - Nutzbar für andere Satellitentypen
 - Nutzbar für alle Anwender in der Entwicklung
 - Rechtemanagement und Zugriffssicherheit auf Modulebene wird benötigt
- SRDB Endbenutzer
 - Aus dem Entwicklungsumfeld
 - Ingenieure

- Tester
 - Qualitätssicherer
 - Projektleiter
- Anwender verteilen sich über Deutschland und Europa
 - Kunden, Auftraggeber
 - Betreiber
 - Projekt-Partner anderer Unternehmen
 - Zulieferer
- MTG-SRDB Roadmap
 - Projektstart 2011
 - Start Anwendungsentwicklung Mitte 2012
 - Start Datenpopulation Mitte 2014
 - Ab Ende 2015 Maintenance
 - Bereithaltung der Daten bis Ende der Mission

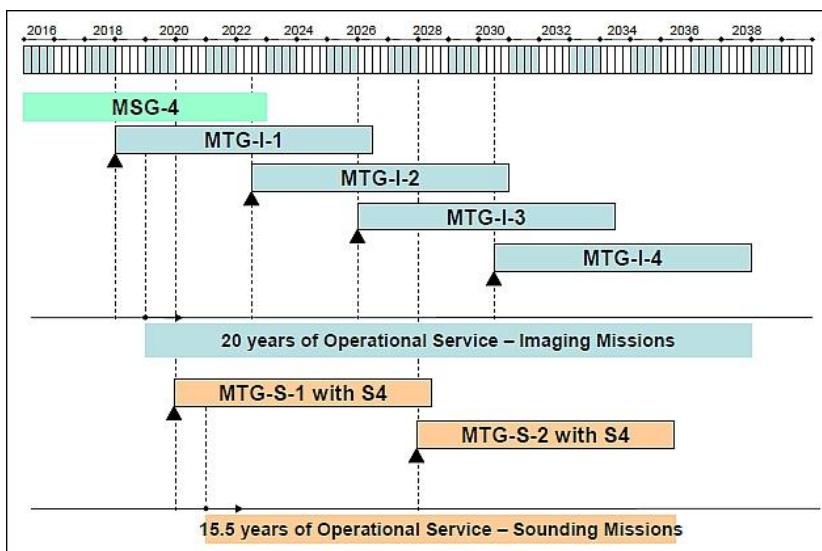


Abbildung 1 - MTG mission notional deployment scenario (image credit: EUMETSAT, ESA)

Projektrahmen – MTG-SRDB Kernanforderungen

Fachlich:

- SRDB bildet sämtliche Satelliten-Lebenszyklen ab
- Change Management auf Datenfeldebene
- Steuerung der Zugriffsrechte auf Modulebene
- Sicherung der kompletten Daten
 - mit Änderungsverfolgung auf Datenfeldebene
- Zugriff für alle Nutzer der Satellitenentwicklung
- Referenz für sämtliche Konfigurationsdaten
- Erzeugen von Steuerdateien für Testzwecke
- Import-/Exportfunktionen relevanter Daten-Entitäten

Andere wichtige Aspekte:

- Extrem lange Laufzeit der Software (20-30 Jahre)
- Basiert auf komplexen Datenmodell
- Import/Export von Massendaten mittels MS Excel

- Erweiterbarkeit für andere Satellitentypen
- Komplexe Validierungen und Abhängigkeiten
- Volle Transparenz über Änderungen
- Wartbarkeit
- Komplexe Datenexporte

Projektrahmen – Das Entwicklungsteam

Beteiligte Personen und ihre Rollen:

- 1 Senior Software Architekt / DB Spezialist
- 1 Senior System Ingenieur / Projektleiter
- 2 Senior Softwareentwickler
- 1 Junior Softwareentwickler
- 1 Testmanager (V&V)
- 1 Testentwickler
- IT Service: Administration
 - des Produktionsservers
 - WLS Spezialisten
- 1 QA Manager

Methodik:

- Schwergewichtiger Softwareentwicklungsprozess, SPICE for SPACE als Kontext und Requirement der ESA
- Innerhalb dieses Umfeldes wurde versucht iterativ in Zyklen zu entwickeln, was aber schnell in eine Art Kanban Modus übergang aufgrund von Projekthindernissen weil bestimmte Requirements erst schrittweise in der Entwicklung klar formuliert werden konnten
- Agile Methoden aus XP wie Daily Stand Up Meeting, mit bspw. Kurzinformationen des Teams über ADF Erfahrungen, Pair Programming, Code Review

Projektdurchführung - Projektvorlauf

Vorgelagerte Tätigkeiten:

- Analyse der Kundenanforderung
- Anforderungsdesign
- Logische Datenmodellierung
- Physikalische Datenmodellierung
- Aufsetzen der Tools
- Aufsetzen der Infrastruktur
- Evaluierung des Entwicklungsframeworks

Dies sind teilweise iterative Tätigkeiten, die sich im Laufe des Entwicklungsprojektes wiederholen, aber natürlich auch initial erbacht werden mussten. Dabei ist Change Management Bestandteil der Entwicklung.

Die Entwicklung-Infrastruktur:

- Lokale Entwicklungsumgebung, damit autonome Entwicklung möglich
 - Build Server Umgebung:
 - CI mit Unit Tests
 - Nightly Build mit Selenium Unit Tests
- Entwicklungsserver

- Produktionsserver (Skalierbar)
 - Produktionssystem
 - Testsystem
 - Entwicklungssystem

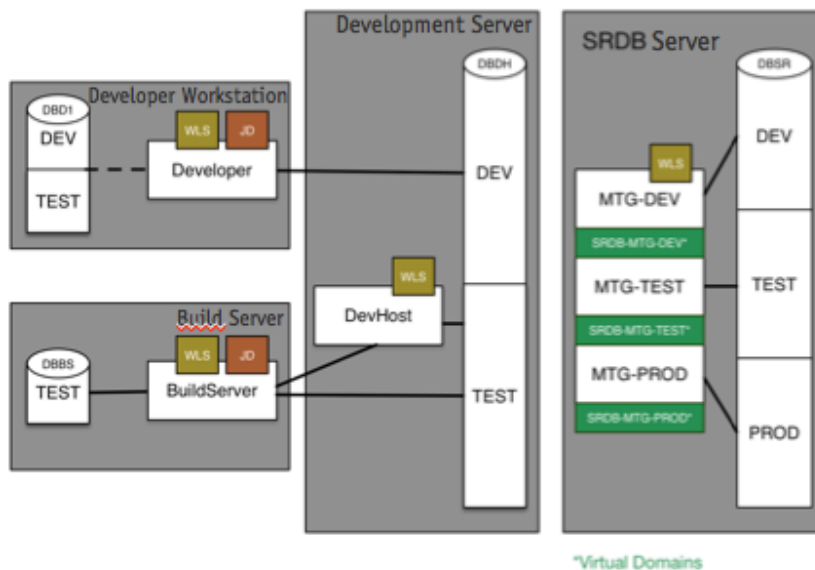


Abbildung 2 - Skizze mit den unterschiedlichen Entwicklungs-Systemen und deren DB Schematas



Auswahlkriterien für ADF als Framework:

- Relationales Datenbank Design ist Vorgabe
 - ADF integriert sehr gut Tabellen als Entity Objects
 - Schichtentrennung
 - Gute Business Modell Schicht (BC)
- Passt zu unserer Oracle Datenbank
- Einfache Umsetzung MVC und GUI
- Modernes und anerkanntes Web Framework
- Große Community, Support, Dokumentation
- ADFdi unterstützt direkt Excel Datenaustausch
- Langzeitperspektive



Evaluierung von ADF / Ziele der Prototypentwicklung:

- Web Anwendung prototypisch bauen
- Einfacher Login ohne Rechtebeschränkung
- JSF Template für die GUI erstellen mit
 - Oben: Anwendungsmenü
 - Links: Tree View (Explorer) zum Navigieren durch die Satelliten Modelle und deren Komponenten
 - Hauptbereich mit der Funktionalität für die ausgewählte Komponente
- Implementierung eines einfachen Master/Details Dialog für eine Daten-Entität

Projektdurchführung - Anwendungsentwicklung



Wie erfolgt die Einarbeitung?

- Schulung ganz am Anfang
- Oracle ADF Tutorials, Videos
- Hands on, praktische Erfahrung sammeln
- Schulungsunterlagen
- Recherche im Netz
 - ADF Community
 - Oracle ADF Development Support Seiten
- Lesen von Dokumente, Bücher über ADF
 - ADF Architecture Square
 - Oracle JDeveloper 11gR2 Cookbook
- Nachschlagen in Büchern

Man landet immer wieder bei den gleichen Leuten aus der ADF Gemeinde, die „Gurus“ wie wir sie nennen. Zum Beispiel:

- The Groundside Blog by Duncan Mills - <https://blogs.oracle.com/groundside/>
- ADF Code Bits by Nick Haralabidis - <http://adfcodesbits.blogspot.de>
- Timo Hahns JDev & ADF Goodies Blog - <https://tompeez.wordpress.com/author/tompeez/>
- Frank Nimphius - <https://twitter.com/fnimphiu>
- Chris Muir - <https://blogs.oracle.com/onesizedoesntfitall/>

Entwickler Absprachen am Anfang:

- Gemeinsame vertikale Anwendungsentwicklung
- DB Modellierung hoheitlich vom Spezialisten
- Identische JDeveloper Projektkonfiguration und Struktur
- Namenskonvention für
 - DB Objekte,
 - Java Code,
 - ADF Objekte,
 - ADF JSF Komponenten
- Hervorzuheben sind hier von Chris Muir die Vorlagen aus der ADF Architecture Square für „Code Conventions“ und „Naming and Project Layout Guidelines“
- Namenskonventionen und Projektstrukturvorgaben werden eigenständig erweitert
- Nachschlagewerk mit ADF Erfahrungen im Wiki
 - Checklisten für ADF Implementierungsabläufe
 - Umsetzungsbeispiele
 - ADF und JD Tipps und Tricks



Einheitliche Konfiguration des JDeveloper Projektes für alle Team Member.

Nach der Standard Installation lohnen sich:

- Konfiguration des ADF Loggers
- Konfiguration des WLS
- Konfiguration Code Formatter
- Sichern der „System“ Umgebung als Backup nach jeder Konfigurationsanpassung
 - Initial mit der frischen Erstinstallation



Kodierungsalltag mit der Umgebung

- Ungewohntes visuelles und deklaratives Entwickeln
 - Support durch Checklisten im Wiki vom Team
- Code Commits in kleinen Schritten („Baby Steps“)
 - Gerade am Anfang sehr wichtig
- Immer Review der Code Änderungen vor einem Commit
 - Source Files könnten unnötig verändert sein
- Ausweichoption auf lokale DB Umgebung hilft sehr
 - Man stört sich nicht gegenseitig
- Nach Abruf von Source Code Updates aus dem Repository
 - immer die JDeveloper Caches löschen
 - alles neu durchkompilieren



Entwickeln eines Software Features

- Review der nötigen DB-Tabellen und der Anforderungen
- Aufbau BC Objects: EOs, VOs, ViewLinks, Acessors, AMs
- Schrittweise Aufbau der Geschäftslogik: Validierungen, Einschränkungen, LOVs, ViewCriterias etc.
- Test des Modells mit dem ADF Model Tester
- Dann erst Aufbau der JSF
 - Wenn möglich in Taskflows zerlegen
 - Schrittweises Einbauen der GUI Elemente
- Unit Tests und manuelle Entwickler-Tests
- Übergabe an das Test Team zur Test-Automation

Schritt 2, 3, 4 und 5 sind fließend und iterativ



Extras bei der Entwicklung von ADFdi :

- Ev. eigene VO Objekte und ein eigenes AM
- Erstellen einer einfachen JSF Seite zum testen
 - hier werden die Bindings definiert
- Manuelles Testen der JSF
- Am Ende wird das Excel Workbook erstellt
- Manuelles Testen des Workbooks



Testen eines Software Features:

- Unit Tests die den Zugriff auf Tabellen prüfen
 - Zur Sicherstellung der Kompatibilität mit dem DB Schema
- Durch Selenium Skripte werden die Dialoge geprüft und Test Cases durchgeführt
 - Test Daten werden in YAML-Dateien gesichert
- Excel Tests Manuell
- CI-Testautomation, durchführen aller Modell Unit Tests
- Nightly Build, zusätzlich alle Dialog-Tests
 - Ergebnisse in ein Report und in das QM Tool



Framework Update

- Einige Updates wurden bereits durchgeführt
 - Bspw. 11g auf 12c
 - Bspw. 12.1.2 auf 12.1.3
- Aufsetzen der neuen Umgebung auf Entwicklersystem
 - Dieser entwickelt die Anpassungen im Try-Branch
 - DB Modell Änderungen auf lokaler Test-DB
- Danach Tests bis alle Probleme ausgeräumt sind
 - Resync mit Trunk, ev. Anpassungen
 - Dann Reintegration in Trunk und Update für alle

Projekt Zwischenbilanz / Fazit

Stand der Anwendung heute:

- DB Tabellen: 210
- DB History Tabellen: 96
- GUI Dialoge: 47
- Bounded Task Flows: 32
- Unbounded Task Flows: 6
- Lifecycle Definition: 100%
- Lifecycle Realisation: 80%
 - Für Daten Entitäten Import via Excel: 100%
- Bereits Produktiv mit getesteten Funktionen

The screenshot displays the 'Realised System Elements Details For Platform-I' window. The main table lists TC Packets with columns for Mnemonic, Identifier, Short Description, Long Description, TC Packet Header, Application Process, Verification Profile, Command Type, High Prio, and Prio I. The selected row is TCPA_01 (P88C0001).

Below the table, the 'Attributes' section for 'TC Packet 01' (P88C0001) is shown, including fields for Index, Mnemonic, Short Description, Long Description, TC Packet Header, Command Type, and various acknowledgment and verification settings.

The 'Telecommand Elements' section at the bottom lists parameters for the selected TC packet, such as P88_P_CMP_001 (P88P0015) and P88_E_ACP_004 (P88E004K).

Abbildung 3 - Screenshot eines Hauptdialogs mit Master / Detail Datenansicht



Lessons Learned

- Schrittweise verbessern und mit Altlasten leben
- Diszipliniert nach den eigenen Vorgaben entwickeln
- Overhead durch Technische Risiken, Forschung
 - Einfache Standardaufgaben gehen schnell
 - Spezielle Anforderungen dauern mitunter unverhältnismäßig
- Nicht durch seltsames Verhalten des JDevelopers täuschen lassen
- In längeren Iterationen planen ist schwer, da Abschätzungen zur Umsetzung zu ungenau sind
- Gegenseitig helfen, Erkenntnisse teilen
- Experten zum Review und Support einbinden



Zukunftsaussichten

- Mehr Unit Testautomation im Modell
- Excel Testautomation
- ADF Funktionsaufrufe besser verstehen
 - Call Stack
- Weiter ADF Wissen ausbauen
 - LifeCycle, Modell Update Cycle und Transaktionen
- Ungelöste Probleme beantwortet bekommen
- Sinnvolle Codemetriken finden und mittels Sonar darstellen



Unsere Best Practices Sammlung

- Gemeinsamer Code Formatter
- Sicherung System Verzeichnis
- Baby Steps Commit
- Immer Code Review vor Commit
- Unit Tests zum Absichern der Schema Kompatibilität
- Erst das einfache Model aufbauen, dann die Details, dann Model Testen, dann erst GUI aufbauen
- Bounded Task Flows verwenden
- CI und Nightly Testing
- Framwork Updates gut testen

Kontaktadresse:

Robert Schweser
megatel Informations- und Kommunikationssysteme GmbH
Lise-Meitner-Straße 2
D-28359 Bremen

Telefon: +49 (0) 0421-22095-0
Fax: +49 (0) 0421-22095-16
E-Mail robert.schweser@megatel.de
Internet: www.megatel.de