

# Vorteile einer Tool gestützten Modernisierung von Forms und Reports

**Jan-Peter Timmermann**  
**Pitss GmbH**  
**Stuttgart**

## Schlüsselworte

Forms, Reports, Modernisierung, Tools, Weblogic

## Einleitung

Mit diesem Vortrag möchte ich mich nicht auf ein Tool konzentrieren, sondern es soll aufgezeigt werden, welche Vorteile sich durch ein Tool ergeben.

Ganz entscheidend für eine Modernisierung ist das Vorgehensmodell welches man nutzen möchte um erfolgreich eine Modernisierung zu tätigen. Dafür sind Aussagen wichtig die die Komplexität der Forms/Reports Anwendung betreffen.

Was sind das für Informationen. Wie gehe ich mit den Informationen um. Was muss ich anschließend wissen und machen um mit meiner Modernisierung weiter zu kommen.

Es werden aber auch die Stolpersteine aufgezeigt die einem auf dem Weg einer Modernisierung über den Weg laufen können. Es soll nicht beschrieben werden wie komme ich von Forms nach ADF/Apex sondern ein allgemeingültiges Vorgehen soll den Zuhörern näher gebracht werden mit dem Ziel ein Gefühl für die Komplexität der Modernisierung zu bekommen.

## Was genau ist IT - Modernisierung

Dazu möchte ich gerne Bernhard Düchting aus seiner Zeit bei Oracle zitieren.

Erhalten der Investitionen in bisherige Bestandsanwendungen bei gleichzeitiger Überführung in eine moderne, standardisierte Technologie

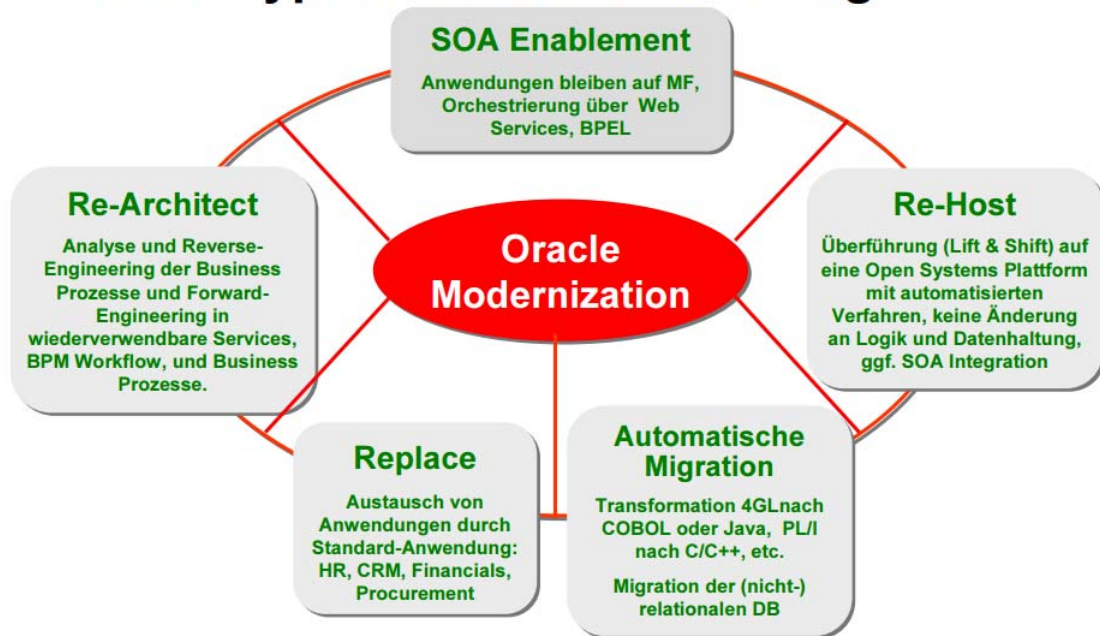
Verbesserung der Wirtschaftlichkeit Ihrer IT-Anwendungen

Hier kann man jetzt zurecht hinterfragen ob Forms sich dann überhaupt modernisieren lässt. Dies ist aber nicht Bestandteil meines Vortrages.

Auch die aktuelle Lüdendonk Studie für 2015 stellt den Bereich Modernisierung noch vor Mobil.

Auch andere Kollegen haben sich des Öfteren mit dem Thema Modernisierungen im Oracle Forms/Reports beschäftigt. Viele dieser Vorträge habe ich live miterleben dürfen. Alle diese Vorträge beschäftigen sich mit dem Thema Forms schöner zu machen. Es werden Vorschläge unterbreitet für einzelne Canvases Java-Bean Komponenten zu nutzen. Dabei sagt der Satz oben, das wir hier von einer Bestandsanwendung reden und nicht von einer Forms Anwendung. Zu einer Bestandsanwendung gehören mit Sicherheit mehr Komponenten als nur einer Forms-Maske. Aber der Reihe nach.

# Die 5 Typen der Modernisierung



**Die Auswahl erfolgt nach der Bestandsaufnahme (Assessment)**

ORACLE

Abb.: 1 Bernhard Düchting Oracle (2008)

## Umfang einer Oracle Forms/Reports Anwendung

Wenn wir über eine Bestandsanwendung reden, haben wir gerade im Oracle Forms/Reports Umfeld es nicht nur mit einer Forms Maske oder einem Report zu tun. In den meisten Fällen haben sich im Laufe der Jahre eine ganze Menge an Skripte/Datenbank Programme und Forms Libraries ergeben. Ebenso wird es mit Sicherheit Schnittstellen zu anderen Bereichen geben. Das können XML/Text Dateien sein oder aber auch Datenbank Routinen.

Im Laufe der Jahre ist die Anwendung immer mehr gewachsen. Es wurde häufig mit Copy/Paste gearbeitet. Das hat natürlich zur Auswirkung, dass in den Masken eine Menge an Logik hinterlegt ist.

Stolpersteine an dieser Stelle sind immer wieder die Themen, dass die Entwickler die das ganze einmal erstellt haben nicht mehr im Unternehmen sind. Die Dokumentation der Anwendung basiert auf diversen Kommentaren innerhalb der Anwendung oder aber auch die Sourcen liegen als \*.fmx vor.

## Warum möchte / muss ich Modernisieren?

Da sind zum einen natürlich die Anforderungen der Anwender. „Die Anwendung sieht ja alt aus“, Support muss bei einer Unternehmenskritischen Anwendung natürlich vorhanden sein. Schnellere Reaktionszeit auf Änderungen. Geringere Betriebskosten durch Konsolidierungen.

Der erste Schritt einer Modernisierung ist immer die Bestandsaufnahme.

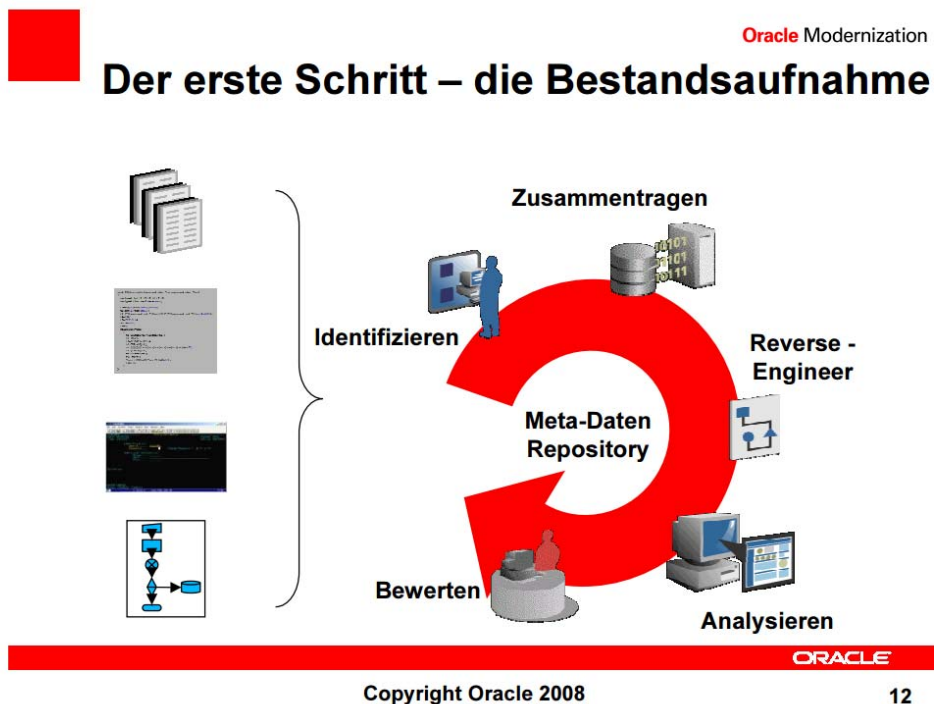


Abb.: 2 1 Bernhard Düchting Oracle (2008)

Dieses Schaubild veranschaulicht sehr schön die notwendigen Schritte einer Modernisierung. Es geht in diesem Falle nicht um Forms sondern um allgemeine IT-Modernisierung. Aber genau diese Schritte treffen natürlich auf für eine Forms/Reports Anwendung zu.

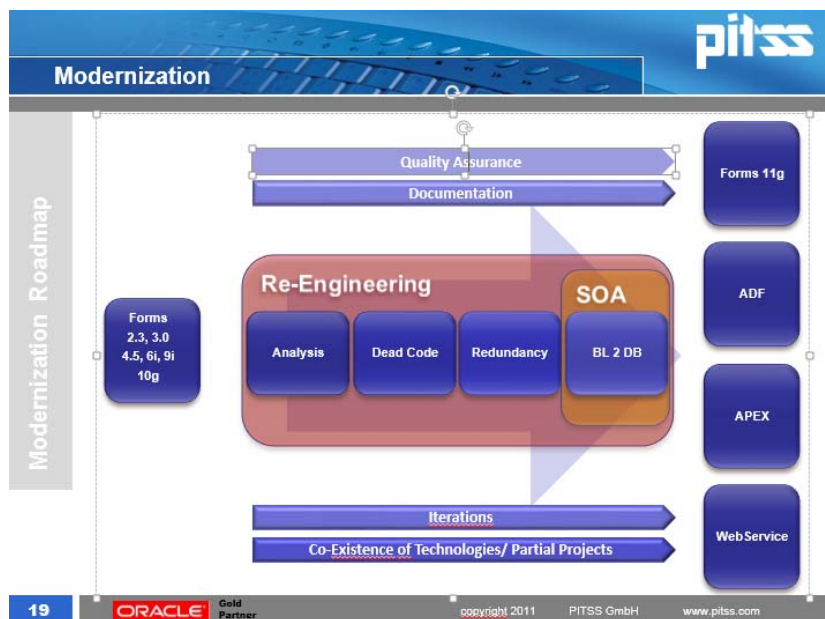


Abb.: 3 Pitss Modernisierungs Pfad

## **Analysen**

Schauen wir uns den Bereich der Analyse etwas näher an. Wenn ich eine Anwendung betrachten möchte kann ich das mit nur einem Modul tätigen. Dort finde ich dann so etwas wie einen Button der einen Trigger anstößt. Dieser wiederum ruft eine Pll Procedure auf, die dann eine Tätigkeit innerhalb der Datenbank ausführt.

Wie kann ich nun erkennen, ob der Button obsolete ist, die Pll obsolete ist oder sogar die Datenbank Procedure obsolete ist. Bei dem Button kann ich das noch händisch machen. Ich muss nur feststellen ob dem Button ein Canvas zugeordnet ist und ob der Button sichtbar ist. Ebenso kann ich mit search im Forms Builder nach einer Aktivierung des Buttons schauen.

Ob jetzt die Pll obsolete ist wird da schon etwas schwerer. Als erstes muss ich natürlich ausfindig machen an welchen Forms Modulen und Reports Modulen hängt diese Pll. Das wird händisch bei 1000 Modulen schon etwas schwerer. Dann kann ich dann Tools nutzen die mir diese Aussagen geben. Am einfachsten ist das allerdings, wenn ich die notwendigen Informationen innerhalb einer Datenbank habe und diese Informationen per sql-Select erhalten kann.(Abhängigkeits – Analyse).

Wenn ich mir nun die Procedure oder besser Funktion in der Datenbank anschau, habe ich das Thema dass diese Funktion innerhalb eine View, Procedure, Funktion, Package, Pll, Forms oder einem Reports genutzt wird. Um nun sicher zu stellen, das diese Funktion obsolete ist muss ich alle Datenbank Objekte anschauen, alle Reports anschauen sowie alle Pll's und alle Forms Module. Auch hier kann mir ein Tool die Arbeit abnehmen, indem ich nach dieser Funktion suche und mir eine Liste aller Objekte ausgeben lasse.

Wenn ich dann feststelle, dass diese Funktion nicht genutzt wird kann ich sie löschen, den Code aus der PLL austragen und den Button in der Maske entfernen. Alle diese Schritte kann ich natürlich händisch machen. Allerdings ist der Aufwand um ein vielfaches höher als wenn ich ein Tool Nutze welches mir die Informationen liefert bzw. ein Repository indem ich die Abfragen per SQL tätigen kann. Zusammen gefasst ist dieses Thema die Deadcode Analyse. Reduzierung des Umfanges der gesamten Anwendung.

Die nächste Analyse wäre dann sql-Statements zu prüfen ob es hier Sinn macht diese in die Datenbank zu posten. Aufgabe an dieser Stelle ist natürlich zu erkennen wo befindet sich SQL-Code. Das können natürlich Forms Procedures, Forms Funktionen und Trigger sein. Die Aufgabe ist es jetzt natürlich diese Routine in eine Datenbank Paket zu überführen und anschließend alle Aufrufe in der Maske auf diese neue Funktion zu überführen. Hier habe ich mit einem Tool natürlich die Möglichkeit alle diese Schritte halb – automatisch zu tätigen.

Stolpersteine bei der Analyse sind immer wieder das „vergessen“ von Modulen oder das nicht betrachten der gesamten Anwendung.

## **PJC - Pluggable Java Components**

Bei vielen Vorträgen auf der DOAG Konferenz wurde schon über die Möglichkeiten geredet PJC's in Forms für Modernisierungen zu nutzen. Schauen wir uns ein Beispiel einer Key-Filter Funktion an. Heute muss man nach der Eingabe das Item prüfen ob die Eingaben in dem gültigen Werte Bereich liegen. So zum Beispiel nur Nummer für ein numerisches Feld. Um bereits beim eintippen Buchstaben zu vermeiden kann man einen PJC – Keyfilter einsetzen.

Innerhalb meiner Forms muss ich den Key-Filter aktivieren und in dem Item die „Implementation Class“ eintragen. Davor muss aber das Item welches in Frage kommt gesucht werden. So wäre es zum Beispiel Sinnvoll alle Items automatisch zu bearbeiten die vom Typ Text-Item sind und einem Canvas zugeordnet sind und Numerischen Inhalt aufnehmen sollen. Bei nur einem Modul wäre der Aufwand für ein Tool sicherlich zu gross aber bei mehr als 100 Modulen kann man sich hier eine Routine bauen, die genau das für einen erledigt.

Ebenso das nutzen von Java Beans. Hier muss die Bean aktiviert werden. Die Bean muss in einem Block angelegt werden und sie muss auf einen Canvas gelegt werden mit einer Größe von 1 x 1 Pixel.

Das alles sind optimaler Weise Tätigkeiten die ich für mehr als nur eine Form per Batch- Lauf erledigen möchte. Ich kann mit Sicherheit über ein Tool festlegen welche Bedingungen erfüllt sein müssen um in den Genuss dieser Bean zu kommen.

### **Fazit**

Es gibt viele Schritte einer Forms Modernisierung. Wichtig ist hier die gesamte Anwendung zu betrachten. Ebenso muss man die Anwendung umfangreich Analysieren um darauf die notwendigen Schritte abzuleiten. Ebenso muss die Analyse natürlich den Aufwand widerspiegeln den die Modernisierung mit sich bringt. Nur so ist man in der Lage den Kosten/Nutzen abzuschätzen.

Es gibt auf dem Markt eine Reihe an Tools die hierzu herangezogen werden können. Es gibt Ansätze formst o XML und dann das Ganze in Apex einlesen. Ebenso gerne genutzt sind Tools wie der Forms Api Master der ein einzelnes Modul einlesen kann und dann darüber informieren kann wie das Modul aufgebaut ist.

Wie in der Eingangsfolie bereits von Oracle erwähnt ist hier der Repository Ansatz zu bevorzugen. Nur so bin ich in der Lage die gesamte Anwendung. Alle Forms Module sowie alle SQL Skript oder Reports zu analysieren und dann die richtigen Schlüsse daraus zu ziehen.

Für alle Tätigkeiten die sich wiederholen macht der Einsatz eines Tools Sinn, da man Fehler vermeidet.

### **Kontaktadresse:**

Jan-Peter Timmermann

Pitss GmbH

Industriestr. 3

D-70565 Stuttgart

Telefon: +49 (0) 172 215 10 43

E-Mail [jtimmermann@pitss.de](mailto:jtimmermann@pitss.de)

Internet: [www.pitss.de](http://www.pitss.de)