

# Forms goes APEX – wie man es richtig macht

Sven-Olaf Kelbert / Niels de Bruijn

MT AG

Ratingen

## Schlüsselworte

APEX, Forms, Migration, Modernisierung

## Einleitung

Immer mehr Unternehmen denken darüber nach, ihre teilweise schon sehr alten Oracle Forms-Anwendungen durch moderne Architekturen zu ersetzen. Dieser Schritt sollte allerdings wohl überlegt und gut geplant sein.

Der Vortrag zeigt, welche Überlegungen angestellt werden sollten, um Forms-Anwendungen zu modernisieren. Eine sorgfältige Analyse der Forms-Masken ist nötig, um möglichst viel des Know-hows, welches über Jahre in die Forms-Masken geflossen ist, in die neue Technologie zu überführen. Des Weiteren stellt sich die Frage, welche neue Technologie zum Einsatz kommen soll. Es werden verschiedene, alternative Technologien gegenüber gestellt und gezeigt, wann und warum eine Umstellung auf APEX sinnvoll ist.

Anschließend wird gezeigt, welche Schritte für eine Umstellung durchgeführt werden müssen und anhand einer Demo vorgeführt, wie eine komplexe Forms-Maske in APEX überführt werden kann. Dabei ist es wichtig zu wissen, welche Forms-Bestandteile in die Datenbank überführt werden sollten, welche Forms-Bestandteile mit Standard-APEX-Funktionalitäten ersetzt werden können, und für welche Funktionalitäten man Erweiterungen bzw. Plug-Ins benötigt.

Eine komplexe Forms-Maske wird dann ihrem APEX-Pendant gegenüber gestellt und verglichen. Daraus wird ersichtlich, auf welche Änderungen man sich einlassen muss, wenn man eine Anwendung von Forms auf APEX umstellt. Die verschiedenen Vor- und Nachteile des Technologiewechsels werden betrachtet und gewichtet.

## Forms-Modernisierung – warum?

Oracle Forms ist eine seit fast 30 Jahren bewährte Technologie, datenzentrierte Business-Applikationen zu entwickeln. Aufgrund des Alters gibt es verschiedene Aspekte, warum eine Forms-Applikation nicht mehr state of the art ist.

Als erstes ist hier das Look&Feel, also das Aussehen der entwickelten Masken zu nennen. Grundsätzlich sind Forms-Masken im Layout von vor mehr als 20 Jahren stehen geblieben. Sie entsprechen damit in keiner Weise Applikationen, mit denen Anwender in der heutigen Zeit vor allem im Internet mehr und mehr zu tun haben. Als Beispiele seien hier Facebook und Xing genannt, aber auch Plattformen wie Amazon oder Ebay. All diese Seiten sind sehr benutzerfreundlich aufgebaut, haben ein modernes Layout, sind mehr oder weniger intuitiv bedienbar, flexibel anpassbar, auf mobilen Geräten zu nutzen und erfordern kein Browser-Plugin.

Dies ist auch bereits ein weiterer Grund für eine Forms-Modernisierung. Heutzutage wollen immer mehr Anwender ihre Business-Applikationen auch vom Smartphone oder Tablet aus bedienen. Da Oracle Forms in einem Applet im Browser läuft, ist dies nicht sinnvoll möglich.

Außerdem gilt die Entwicklungsumgebung von Oracle Forms als altbacken. Studienabsolventen haben in den wenigsten Fällen bereits an der Universität oder Fachhochschule damit zu tun gehabt. Sie haben dort moderne objektorientierte Sprachen und Frameworks kennen gelernt und wollen diese nun auch in der Praxis anwenden. Forms hat also ein Entwickler-Nachwuchs-Problem. Deshalb ist es durchaus sinnvoll, eine Erweiterung einer Anwendung über eine neue Technologie abzubilden.

Ein weiteres Problem ist, dass man nicht auf die in den Forms-Masken vorhandene Business-Logik von außen zugreifen kann. Wenn im Unternehmen also neue, moderne Applikationen entwickelt werden, muss die gesamte Business-Logik aus den Forms-Masken extrahiert oder gar nachprogrammiert werden. Eine direkte Wiederverwendung des Codes ist nicht möglich.

Eine Einstellung des Supports durch Oracle, welche vor einigen Jahren ein großes Thema war und viele Kunden aufgeschreckt hat, ist allerdings mittlerweile kein akutes Thema mehr, da Oracle Forms zumindest ab dem Release 10gR2 unter den Lifetime Support fällt.

Trotzdem sollte man sich aufgrund der vorher genannten Themen überlegen, ob und wann eine Modernisierung von vorhandenen Forms-Anwendungen sinnvoll sein könnte.

### **Forms-Modernisierung – wohin?**

Bevor man sich Gedanken über eine Ablösung von Forms machen kann, sollte man sich darüber Gedanken machen, welche Plattform denn für die eigenen Zwecke die sinnvollste Variante sein könnte.

In den meisten Fällen werden folgende Optionen unterschieden:

- Beibehaltung von Forms (Migration auf neueste Releases)
- Ablösung durch ADF
- Ablösung durch APEX
- Ablösung durch nicht-Oracle-Produkte

Die nicht-Oracle-Produkte können dabei beliebige Frameworks sein, die sich nach den Vorlieben oder Gegebenheiten des Kunden orientieren können. Hier könnte .NET eine Rolle spielen oder Open Source Java Frameworks.

Normalerweise steht nun eine Technologie-Evaluation an, die beliebig aufwändig werden kann. Hier können nur grob die wichtigsten Merkmale verglichen werden oder ein komplettes Prototyping in den verschiedenen Technologien mit konkreten Business Cases durchgeführt werden. Ein Ergebnis könnte dann wie folgt aussehen:

	Forms	APEX	ADF	.NET
GUI	1	3	3	3
Wiederverwendbarkeit	2	2	3	2
Web-Tauglichkeit	1	3	3	3
Migration	3	3	2	1
Kommunikation mit Oracle DB	3	3	3	2
Standard-Komponenten	2	3	3	2
individuelle Komponenten	2	3	3	3
Integration	1	3	3	2
vorhandene Entwickler	3	2	1	1
neue Entwickler	1	3	3	2
Summe	19	28	27	21

Abb. 1: Technologie-Vergleich

Diese Matrix kann natürlich für jeden Kunden je nach Gegebenheiten unterschiedlich aussehen, sowohl von den Kriterien her als auch von der Auswertung her.

Manchmal ergibt sich direkt aus der Matrix bzw. aus den bei der Entstehung der Matrix gemachten Überlegungen direkt die sinnvollste neue Technologie. Ist dies nicht der Fall, können auch separate Aufwandsschätzungen von Spezialisten einen Hinweis geben. So haben wir für ein Projekt sowohl für eine Neuentwicklung mit APEX als auch für die Umsetzung mittels ADF bzw. Java Aufwandsschätzungen ermittelt. Bei der Entwicklung mit APEX lag der Aufwand bei ca. 380 PT, bei ADF wären es ca. 490 PT gewesen und bei reinem Java bei ca. 590 PT.

Zu den Aufwandsschätzungen kommen natürlich noch die durchaus wichtigen Kosten, was es bedeutet, die bestehende Entwickler-Mannschaft in der neuen Technologie fit zu bekommen, damit diese die Umstellung oder zumindest ab einem gewissen Punkt die Wartung und Betreuung der umgestellten Anwendung übernehmen kann. Die Entwickler im Oracle Forms Umfeld haben neben den Forms-Kenntnissen im Wesentlichen SQL- und PL/SQL-Kenntnisse. Dies sollte man nicht vernachlässigen, da in business-kritischen Applikationen meist sehr viel Geschäftslogik enthalten ist.

Bei einer Migration nach APEX kann diese Geschäftslogik meist zu großen Teilen übernommen werden. Entweder, sie ist sowieso in der Datenbank vorhanden. Dann ist kaum Anpassung nötig. Oder die Geschäftslogik muss zumindest aus den Forms-Masken in die Datenbank verlagert werden, was aber immer noch wesentlich einfacher ist, als diese komplett neu in Java oder C neu zu entwickeln. Außerdem ist Java objektorientiert im Gegensatz zu PL/SQL, was die Akzeptanz bei den Forms-Entwicklern erschwert.

### Forms-Modernisierung – wie?

Hat man sich nun dazu entschieden, die Forms-Applikation zu modernisieren (was noch nicht zwingend auch direkt eine neue Technologie bedeuten muss), stellt sich die Frage nach der

Vorgehensweise. Hier gibt es mehrere Möglichkeiten: eine komplette Neuentwicklung, eine automatische 1:1-Migration mit einem Migrationswerkzeug, ein einfaches Forms-Upgrade auf das neueste Release oder eine Kombination der Wege. Vor- und Nachteile der verschiedenen Wege werden in der Präsentation vorgestellt.

### **Forms goes APEX**

Für viele Kunden ist der Weg nach APEX der präferierte Weg, da es viele gute Gründe für APEX gibt. Ist diese Entscheidung gefallen, geht es darum, einen möglichst sanften Übergang in die neue Technologie zu finden. Die Anwendung sollte in Teilanwendungen aufgeteilt und priorisiert werden.

Diese Teilanwendungen werden nun modernisiert, indem zuerst möglichst viel Geschäftslogik aus den Forms-Masken in die Datenbank ausgelagert wird, da diese dann auch von APEX aus genutzt werden kann. Dann werden die Teilanwendungen Stück für Stück nach APEX überführt. Nach Bedarf kann eine Kommunikation zwischen der Altanwendung und der neuen APEX-Applikation hergestellt werden, so dass beide Systeme eine Zeitlang parallel laufen können und keine Big-Bang-Umstellung nötig ist.

Mit der Umstellung auf APEX erreicht man schnelle Ergebnisse und erhöht dadurch von Anfang an die Akzeptanz bei den Anwendern, da kein jahrelanges Vor-sich-hin-Entwickeln nötig ist. Kunden haben so recht einfach die Möglichkeit, ihre Anwender zufrieden zu stellen und auch die Entwickler mitzunehmen in die Zukunft der Anwendungsentwicklung.

### **Kontaktadresse:**

Sven-Olaf Kelbert, Niels de Bruijn  
MT AG  
Balcke-Dürr-Allee 9  
D-40882 Ratingen

Telefon: +49 (0) 2102-309610  
Fax: +49 (0) 2102-30961101  
E-Mail [sven-olaf.kelbert@mt-ag.com](mailto:sven-olaf.kelbert@mt-ag.com), [niels.de.bruijn@mt-ag.com](mailto:niels.de.bruijn@mt-ag.com)  
Internet: [www.mt-ag.com](http://www.mt-ag.com)