

ADF Software Factory - Software aus der Werkstatt

Ulrich Gerkmann-Bartels
enpit consulting OHG
33102 Paderborn

Schlüsselworte

Oracle ADF, Software Factory, Asset

Einleitung

Nach einer gewissen Anlaufphase konnten erste Oracle ADF 11g Projekte zwischenzeitlich von vielen Entwicklungsteams erfolgreich umgesetzt werden. So manche Hürde bei der Einführung einer neuen Technologie wurde überwunden und neben der Anwendung selbst, sind wertvolle Erfahrungen entstanden. Doch der Schatz an Möglichkeiten, die in diesem umfangreichen Oracle Application Development Framework (ADF) und der Fusion Middleware Plattform liegen, ist noch lange nicht vollständig geborgen.

Viele Entscheider und Projektleiter fragen sich, ob das nächste Projekt wieder genauso teuer wie das Erste sein wird. Gibt es in weiteren Projekten einen Mehrwert, der sich in ADF Projekten gemeinsam nutzen lässt? Können die Kosten in Nachfolgeprojekten durch die Verwendung von zuvor entstandenen "Standardmodulen" reduziert werden?

Der Vortrag skizziert den Aufbau einer Fertigungsstraße für ADF Projekte im Sinne einer Software Factory. Dargestellt werden alle sinnvollen Fertigungsschritte bis zur Auslieferung eines Softwareproduktes aus der Werkstatt. Unter anderem wird die Organisation und Ablage von Artefakten zur Wiederverwendung als Zulieferprodukt in weiteren Projekten dargestellt. Die Sinnhaftigkeit spezialisierter Tools - Unterstützung für die Generierung von Code und die Vereinfachung in der Entwicklung, ergänzen die Darstellung der Software Factory.

Begriffsdefinition

Der Begriff Software Factory ist im Wesentlichen heute von den Autoren J. Greenfield, K. Short, Cook S. und Kent S. durch Ihr Buch „Software Factory“ geprägt.

„...a **software product line** that configures extensive tools, processes, and content using a **software factory template** based on a **software factory schema** to automate the development and maintenance of variants of an archetypical product by adapting, assembling, and configuring framework-based components.“^[1]

Entnimmt man wesentliche Kernelemente der Aussage der Autoren, erhält man folgende Rahmendefinition, die innerhalb des Vortrages im Vordergrund stehen:

- a.) Eine Software Factory ist eine organisatorische Struktur mit dem Ziel, Applikationen und Komponenten durch einen „Montage“- Prozess zu produzieren.
- b.) Software wird erstellt durch Verwendung und Zusammenstellen von vorher definierten Komponenten. Neue Komponenten und Services werden von Teams oder Lieferanten in einen Produktionsprozess geliefert.
- c.) Das zusammengestellte Endprodukt ist das produzierte Ergebnis der Software Factory.

Bewertet man diese drei Kernaussagen in Bezug zu der Plattform Oracle Fusion Middleware und insbesondere des Oracle Application Development Frameworks, so wird jeder Software Architekt bestätigen, dass sich diese Leitgedanken einer Software Factory in der Plattform wiederfinden.

Dies gilt insbesondere für das Zusammenstellen und Produzieren des Endproduktes einer Software Factory.

Elemente der ADF Software Factory

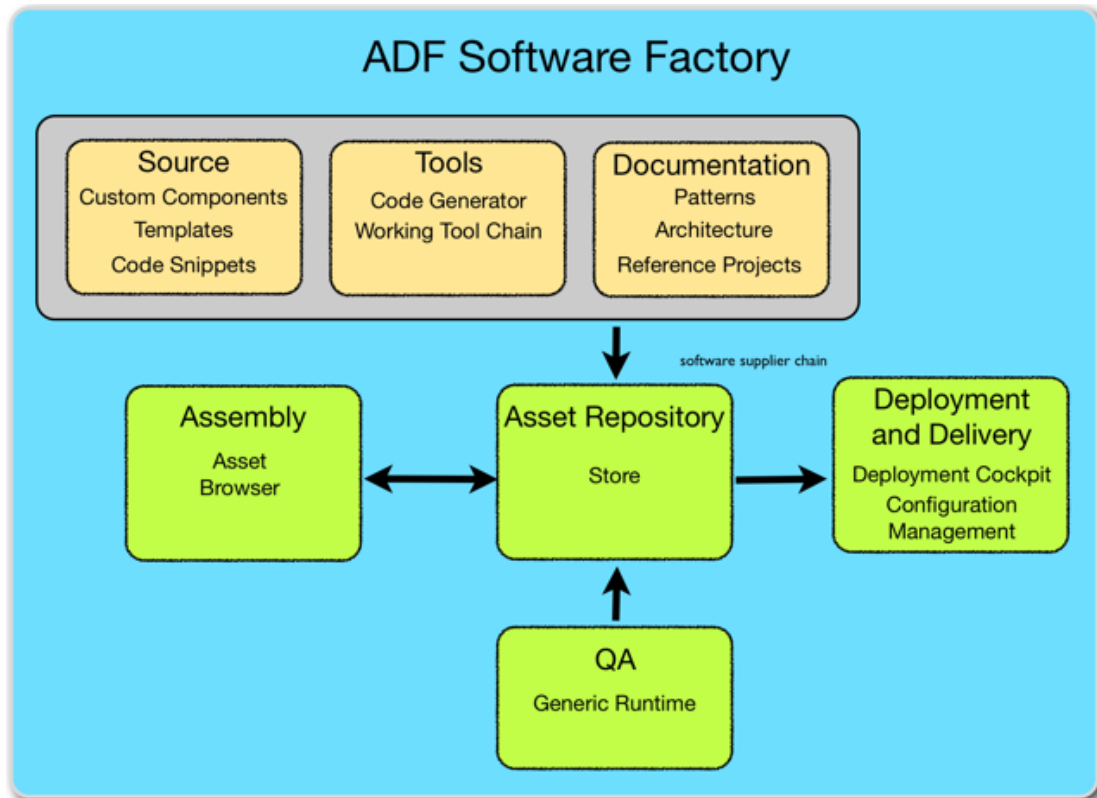


Abb. 1: ADF Software Factory

Ausgehend von der Darstellung Abb. 1, die den Rahmen der Software Factory vorgibt, gilt es festzulegen, welche Komponenten verfügbar und nutzbar bzw. in der unternehmenseigenen Fabrik zu definieren sind.

Source

Um eine hinreichend gute Ausgangsposition bei der Fertigung einzelner neuer Module zu erhalten, gilt es Sourcen für die individuelle Implementierung innerhalb der Werkstatt zur Verfügung zu stellen. Dies ist notwendig, um eine Divergenz innerhalb der zu entwickelnden Komponenten und äußeren Erscheinungsbildern zu verhindern. Insbesondere sind hier folgende Oracle ADF Artefakte zu sehen:

- ADF (Custom) Declarative Components, um wiederkehrende Darstellung und Funktionen auf der kleinsten User Interface - Ebene zu bündeln. Beispiele: Eingabe von Telefonnummern mit Auswahl der Kategorisierung (Privat, Arbeit, Mobile), Verfahrenseingabe mit mehreren Kriterien (Aktenzeichen, Jahr, Dienststelle) oder Funktionsleisten, die je nach Kontext an entsprechende Daten gebunden werden.
- JSF Page Templates, für die Gestaltung des Seitenaufbaus mit integrierten Inline Popup - Dialog für die Nutzung von Bestätigungsabfragen.

- ADF Task Flow Templates, für Steuerungsflüsse in der Applikation.
- ADF Skin
- Bibliothek von Code Snippets, um jeden Entwickler bei der Realisierung notwendiger Implementierungsaufgaben einfach und verifiziert zu unterstützen. Typische Beispiele sind hier Funktionen auf der Ebene: Ermittlung der selektierten Datensätze, Ausklappen eines Teilbaum im ADF TreeTable oder Loop durch eine Menge von Elementen.

Tools

Die Auswahl des Entwicklungswerkzeugs ist zur Zeit bei der Verwendung von Oracle ADF unter Berücksichtigung des Fusion Development Stack durch den Oracle JDeveloper besetzt. Durch den Funktionsausbau der Oracle Enterprise Pack for Eclipse ist sicherlich in Zukunft ein weiteres Kernentwicklungswerkzeug einsetzbar. Da die Software Factory vom Konzept her an dem Zulieferprodukt interessiert ist, wird es sicherlich in Zukunft auch Teilfertigungen geben, die nicht mit dem Oracle JDeveloper entwickelt werden.

Der Wunsch, die wesentlichen Elemente einer Fertigungstraße im Sinne der Software Factory in einer integrierten Entwicklungsumgebung darzustellen, ist nicht Out-Of-The-Box zu realisieren, da der Oracle JDeveloper nicht auf die jeweiligen spezifischen Ausprägungen in den einzelnen Entwicklungsteams eingehen kann. Um hier den Fertigungsprozess im Hauptwerkzeug zu unterstützen, ist es sinnvoll, die vorhandene Funktionen durch eigene Plugin bzw. Extensions zu erweitern, die auf die individuellen Kontexte der Organisation abgestimmt sind. Benötigt werden hier in erster Linie folgende Erweiterungen:

- Ein Wizzard, der eine Modulapplikation mit allen benötigten unternehmenseigenen Bibliotheken, Konfigurationen und Verzeichnisstrukturen anhand eines vorgegeben Namensraums erstellt. Die vorhandene Lösung der Application Templates ist nicht ausreichend.
- Code Generatoren, die auf Basis vorhandener Metadaten Teilaspekte generieren. Beispiel: Aufruf von PL/SQL Store Procedures oder CRUD - Module für Master Data Management Anwendungen.

Wünschenswert wäre die Möglichkeit, die vorhandenen Code Templates im Oracle JDeveloper offen zu legen. Dies würde eine individuelle Anpassung von Drag&Drop - Operationen ermöglichen, um eigene UI Componenten oder Attribute individuell zu belegen.

Ein Erfolgsfaktor bei der Auswahl der Werkzeuge ist das Zusammenspiel und die Austauschbarkeit der einzelnen Tools. Diese Faktoren sind auf die Verwendbarkeit und Lebenszeit der Software Factory höher zu bewerten, als die Funktionsvielfalt der einzelnen Produkte. Zu berücksichtigen ist hier, dass neue Technologien, die es gilt zu unterstützen, meist auch neue Werkzeuge erfordern.

Beispiele: Entwicklung von Applikationen auf Smartphones oder der Wechsel vom Java EE Server OC4J auf Weblogic Server.

Beispielhafte Werkzeuge einer ADF Software Factory für den Bereich Tools

- Visual Vision, Innovator, ArgoUML
- Oracle JDeveloper, Eclipse mit OEPE
- SQL Developer, Toad, PLSQL Developer, SQL*PLUS
- Velocity, Eclipse Modeling

- CVS, Subversion, Git
- Hudson, Jenkins, TeamCity
- Bugzilla, Jira, Redmine
- Ant, Maven
- Scripte und Plugins für die einzelnen Werkzeuge, um einen reibungslosen Werkzeugübergang und Integration derer zu gewährleisten.

Documentation

Ein weiterer wichtiger Faktor innerhalb der Software Factory ist die Dokumentation und Verfügbarkeit von „Produktionsanweisungen“ und Referenzlösungen. Sie dienen dazu, dass jedes Entwicklungsteam oder Zulieferer ein gemeinsames Bild von der gewählten Architektur und Vorgehensweisen bei der Implementierung hat. Für eine erfolgreiche Software Factory ist es nicht ausreichend, die Architektur vorzugeben und die Realisierung den Teams zu überlassen. Benötigt wird eine Interaktion. Durch verschiedene Fachanforderungen in einzelnen Modulen erscheinen immer wieder neue Herausforderungen, für die meistens noch keine dokumentierte Vorgehensweise oder Best Practices entwickelt worden ist. Diese sollten durch ein qualifiziertes ADF Kernteam angegangen und dokumentiert werden, um den Fertigungsprozess für die gesamte Werkstatt zu optimieren. Diese Vorgehensweise sollte durch ein Verfahren etabliert werden, das auch eine Priorisierung (Voting) der erfassten Aufgabenstellungen beinhaltet. Die Priorisierung zeigt durch die Anzahl der abgegebenen Stimmen, welche Herausforderungen in der Fertigung in technischer Hinsicht noch existieren und den Fertigungsprozess behindern. Eine solche Dokumentation sollte folgende Elemente beinhalten:

- Design Guide
- Style Guide
- Development Guide
- Best Practices und Referenzlösungen (Vgl. Nimphius F., ADF Code Corner)
- Plattform, um Herausforderungen in der Implementierung der fachlichen Anforderungen zu erfassen, zu bewerten und Lösungsmuster zu dokumentieren.
- Suchfunktionalität

Asset Repository

Alle realisierten Komponenten und Module werden in der Software Factory in ein Asset Repository eingelagert.

Dieses Repository dient dazu, beispielhaft folgende Informationen festzuhalten:

- Namensraum
- Version
- Kompatibilität (Minimum Version bis Maximum Version)
- Abhängigkeiten

- Verwendungshäufigkeit
- Typ (Application, ADF Library, WLS Shared Library, Java Library, JDeveloper Extension)
- Verweis auf ein Artefakt
- QA - Status (Entwicklung, Test, Integration, Produktion)

Des Weiteren kann das Asset Repository im Zusammenspiel mit dem Konfigurationsmanagement für die einzelnen Komponenten und Langzeitumgebungen (Entwicklung, Test, Integration, Produktion) genutzt werden, um den Auslieferungsprozess (Deployment und Delivery) oder das Aufsetzen eines neuen Fertigungsprozesses eines weiteren Moduls zu beschleunigen.

Ansatzweise stehen folgende Werkzeuge zur Realisierung zur Verfügung:

- Oracle Enterprise Repository
- Maven + Custom Solution
- Subversion + Custom Solution

QA

Um die Qualität der im Asset Repository befindlichen Module unabhängig von einem Endprodukt zu überprüfen, ist es sinnvoll diese Teilkomponenten in einem Akzeptanztest zu verifizieren. Dazu wird eine generische Laufzeitumgebung benötigt, die in einer einfachen Art und Weise in einem bestehendem manuell oder automatisierten Testprozess eingebunden werden kann.

Werkzeuge:

- ADF - EMG Task Flow Tester
- Custom Solution

Assembly

Basierend auf dem Asset Repository werden die Endprodukte der Software Factory konfiguriert und zusammengestellt. Die zusammengestellte Konfiguration wird in einem Ausführungsrahmen einer sogenannten Master Application ausgeführt.

Realisiert wird dieses durch ein Enterprise Archiv (EAR), mit zugeordneten Modulen in Form von enthaltenen ADF Libraries oder Abhängigkeiten zu eigenständig ausgelieferten Weblogic Server Shared Libraries.

Diese Abhängigkeitsinformationen und Deployment Profile werden im Asset Repository abgelegt.

Werkzeuge:

- Custom Solution

Deployment and Delivery

Durch die Kombination der Informationen, die im Asset Repository und Konfigurationsmanagement vorliegen, kann ein teilautomatisiertes Deployment und Delivery Prozess etabliert werden.

Durch die Einführung eines Deployment Cockpits werden zwei zentrale Punkte der Software Factory im Auslieferungsprozess gesteuert.

1. Welche Applikationen, Bibliotheken müssen auf welche Laufzeitumgebung ausgeliefert werden bzw. zum Download für den Endkunden bereit gestellt werden.
2. Darstellung der aktuellen ausgelieferten Produkte und der Abhängigkeiten zwischen den Komponenten und Modulen.

Werkzeuge:

- Custom Solution

Ausblick

Erkundigt man sich aktuell über den Stand der Visionen und Produkte von Microsoft zum Thema Software Factory, so stellt man verblüffenderweise fest, dass das Produktteam Pattern & Practice die vorhandenen Produkte wie Web Client Software Factory nicht weiter in Eigenregie vorantreibt.

The screenshot shows the Visual Studio Marketplace page for the extension 'Web Client Software Factory 2010'. The page features a header with the product name and a 'Free' badge. Below the header is a brief description: 'The Web Client Software Factory is a Visual Studio extension that helps software teams create composite web applications.' The main content area is divided into several sections: 'CREATED BY' (patterns and practices (Microsoft)), 'LAST UPDATED' (10/15/2011), 'REVIEWS' (4 stars, 3 reviews), 'VERSION' (3.0), 'SUPPORTS' (Visual Studio 2010), 'LICENSE' (View), 'DOWNLOADS' (Download button, 21,089), 'SHARE' (social media icons), 'FAVORITES' (Add To Favorites button), and 'TAGS' (GAX, web development, Web, patterns & practices, Software Factory, WCSF). Below this is a 'DESCRIPTION' tab, which is currently selected. The description area contains a large image of the product box and a text block: 'Web Client Software Factory 2010. The Web Client Software Factory 2010 helps software development teams create Web applications in VS 2010 according to predefined templates and patterns.' At the bottom of the page, there is a red warning message: 'This project is no longer being maintained and no future releases are planned.' and a 'System Requirements' section listing 'Visual Studio 2010 Ultimate, Premium, or Profession editions (32 or 64 bit)'.

Abb. 1: Web Client Software Factory 2010

Warum Microsoft diese Produkte und damit auch vielleicht die Vision nicht weiterführt, entzieht sich der Kenntnis des Autors. Ein wesentlicher Punkt bei Betrachtung der Software Factory, gilt der

Schaffung der organisatorischen Struktur und dem Zusammenspiel der einzelnen Produzenten und Zulieferer. Dies ist sicherlich unabhängig vom Werkzeug. Aber gerade hier gilt es, aktuell die entsprechenden Rahmenbedingungen zu setzen, um Herausforderungen wie SaaS, Cloud Computing, Continuous Delivery oder der Einführung einer neuen Software Development Plattform gerecht zu werden.

Referenzen

1. Greenfield, Jack; Short, Keith; Cook, Steve; Kent, Stuart (2004). *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*. ISBN 0-471-20284-3
2. Müller T., Web Client Software Factory, ASP .NET Professional 01/2008
3. Web Client Software Factory, <http://visualstudiogallery.msdn.microsoft.com/8fc2c9d6-5282-404b-9c58-e72b326b1e26>, abgerufen am 13.09.2012
4. Nimphius F. , ADF Code Corner

Kontaktadresse:

Ulrich Gerkmann-Bartels
enpit consulting OHG
Theodor-Heuss-Str. 17c
D-33102 Paderborn

Telefon: +49 5251 20277 93
E-Mail: ulrich.gerkmannbartels@enpit.de
Internet: www.enpit.de