

Life Cycle Management für Apex Applikationen im agilen Projektumfeld

Sven-Uwe Weller
syntegris information solutions GmbH
Neu-Isenburg

Schlüsselworte:

Apex, agile development, deployment, Hudson

Einleitung

APEX ist ein Rapid Application Development (RAD) Tool, mit dem sich sehr schnell erste Erfolge bei der Entwicklung kleinerer Applikationen einstellen. Sobald die Entwicklerteams größer werden oder spätestens, wenn die Anforderungen aus dem Unternehmensumfeld an die Entwicklungsabteilung eine strukturierte Vorgehensweise verlangen, dann wird es Zeit, über die Versionierung der Applikation und die Automatisierung von Tests nachzudenken. Hierbei werden oft Schwächen von APEX sichtbar.

Die Präsentation gibt Erfahrungen weiter und zeigt anhand von Best Practices auf, wie APEX Applikationen gemanagt werden können ohne an Agilität zu verlieren. Der Schwerpunkt liegt dabei nicht auf dem erstmaligen Erstellen einer Applikation, sondern auf der kontinuierlichen Weiterentwicklung und Verbesserung, insbesondere auf Vorgehensweisen beim Übergang von Entwicklung in Test.

Theorie Softwarelebenszyklus

Eine Applikation ist ein Softwareprodukt. Als solche unterliegt sie auch dem typischen Produktlebenszyklus (Einführung, Wachstum, Reife, Sättigung, Degeneration) wie er aus der Betriebswirtschaftslehre bekannt ist.

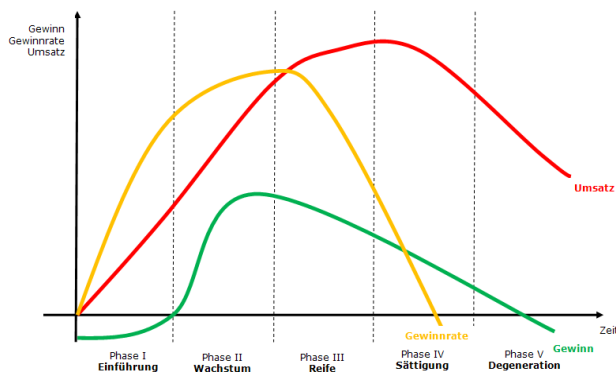


Abb. 1: Produktlebenszyklus (Quelle: wikipedia)

(<http://upload.wikimedia.org/wikipedia/commons/4/48/Produktlebenszyklus2.png>)

Reifephase bedeutet meist, dass das Produkt immer weiter verbessert wird. Bei der Entwicklung von Softwareprodukten wird oft unterschätzt, wie wichtig die Reifephase eines Produktes ist. Agile Methoden helfen durch schnelle Anpassungs- und Erweiterungsmöglichkeiten immer neue Softwareversionen herauszubringen und damit die Lebensdauer einer Applikation zu verlängern. RAD

ist also nicht nur nützlich um schnell eine Applikation zu erstellen (time-to-market), sondern auch um schnell auf weitere Anforderungen zu reagieren und diese in der Software umzusetzen.

Manuelles Deployment

Deployment bedeutet, dass eine fertige Software oder auch eine neue Version in eine neue Umgebung ausgerollt/installiert wird. Es gibt zwei typische Punkte wo das auftritt: Beim Übergang aus der Entwicklungsumgebung in die Testumgebung und beim Übergang von Test in die Produktivumgebung.

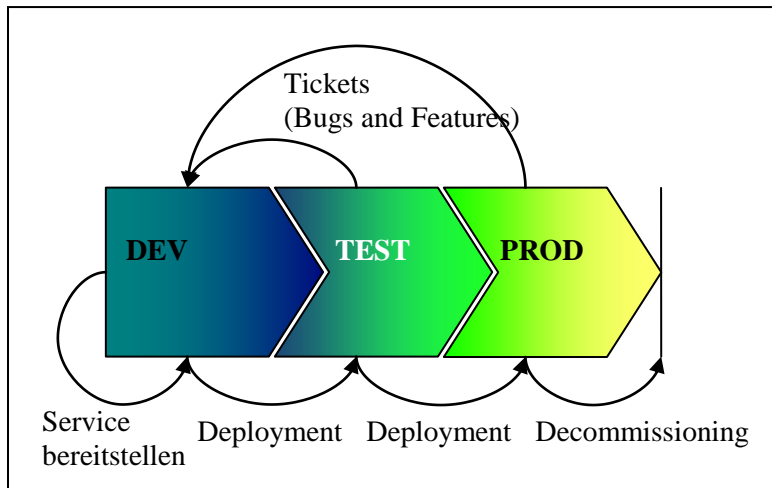


Abb. 2: Phasenübergänge

Eine typische APEX Applikation hat mehrere Komponenten, die kopiert werden müssen

- die APEX Applikation selbst (hauptsächlich die Pages)
- Zugehörige Objekte im Workspace (Shared Components, Themes, Templates, Images, CSS)
- Datenbankobjekte
- Daten
- Sonderfälle, wie User Settings für Interaktive Reports, unterschiedliche Logos/Styles für Dev, Test und Produktion.

Für alle diese Objekte muss man überlegen, wie die Versionierung bzw. der Übergang von einer Umgebung in die andere erfolgt.

Eine Strategie kann sein, dass die Objekte als Installations- und/oder Upgrade Skripte in APEX hinterlegt werden. Dies ist aber nicht besonders zuverlässig. Es ist ein manueller Schritt, der gerne vergessen wird, z.B. weil der Entwickler das nicht kennt. Aber auch Fälle, wie das Anlegen eines Index, werden von DBAs vorgenommen und nicht als APEX Upgrade Script in der Anwendung hinterlegt. Dies ist jedoch ein generelles Problem, nicht nur für APEX Applikationen.

Empfehlung ist die Datenbankseite (Objekte + Daten) komplett von der Applikationsseite zu trennen. D.h. die APEX Applikation zu exportieren, ohne Rücksicht auf geänderte Datenbankobjekte zu nehmen. Alles, was auf der Datenbank liegt, sollte dann gesondert betrachtet werden, so wie es in anderen Projekten auch gemacht wird. Dabei werden Änderungen über das normale Versionsverwaltungstool als DB Skripte abgespeichert. Die APEX Applikation wird exportiert und das Export Script ebenfalls im SVN gesichert. Namenskonventionen sind für die korrekte Erkennung der DB Objekte hilfreich. Eine Empfehlung ist, dass alle Objekte mit einem Kürzel versehen werden,

welches klar macht, zu welcher APEX Applikation sie gehören. Das ist vor allem dann wichtig, wenn mehrere Applikationen sich den gleichen APEX Workspace und die gleichen DB Schemas teilen.

Wichtig ist, dass die verschiedenen DEV/TEST/PROD Workspaces möglichst die gleiche Workspace ID haben. Das ermöglicht es später nicht nur die komplette Applikation neu einzuspielen, sondern mittels Page Import/Export auch einzelne Seiten zu erneuern. Beim Aufsetzen der Workspaces sollte der Administrator deshalb darauf achten, einen Import des Workspaces vorzunehmen und die IDs zu übernehmen.

Eine sehr nützliche Funktion ist es, dass man in APEX Applikations-Variablen definieren kann, die beim Export neu belegt werden können. Damit lassen sich einige Komponenten ändern, die zwischen den verschiedenen Zieldatenbanken nicht gleich sein dürfen. Beispiel wäre ein Image auf der Login Page. Dazu müssen drei kleine Schritte gemacht werden:

- 1) Unter „Shared Components>Edit Application Definition“ wird ein Substitutionsstring angelegt „LOGOFILE“. Dort steht der Name des Images, z.B. myApp_logo_Test.PNG.
- 2) Auf der Login Page wird dieser Substitutionsstring verwendet.
``
- 3) In den Installationsanweisungen wird hinterlegt, dass dieser Substitutionsstring während der Installation extra abgefragt wird. „Supporting Objects>Edit Substitutions“. Checkbox setzen.

Automatisches Deployment

Automatisches Deployment bietet sich vor allem dann an, wenn man schnell eine saubere Testumgebung neu aufsetzen möchte. Es gibt eine relativ unbekannt und schlecht dokumentierte Funktion, mit der man den Export einer APEX Applikation auch mittels Kommandozeile ausführen kann. Dies läuft über Java-Klassen.

In APEX 4.0 gibt es für diesen Export neue zusätzliche Parameter, die sich auf die Erweiterungen bei den Interactive Reports und Team Development beziehen. Die wichtigste Neuerung bei den Interactive Reports ist, dass man zwischen den privaten Reports und den für alle anderen User freigegebenen Standard Alternativen unterscheiden kann. Die bei APEX ebenfalls mitgelieferte Java-Klasse „APEXExportSplitter“ erlaubt es ein ExportFile in viele Einzelfiles (entweder als Directory-Baumstruktur oder mit strukturierten langen Filenamen) zu zerlegen.

Es empfiehlt sich diesen Splitter einzusetzen, denn Einzelfiles sind für die Versionierung natürlich informativer.

Hudson kann in dem Zusammenhang als etwas komplexerer Scheduler betrachtet werden, der alles ausführen kann, was in ein Shellscript passt. Ein automatisches Deployment ergibt sich, wenn man regelmäßig (täglich!) den aktuellen Stand der Applikation exportiert und in das Testsystem einspielt.

Testautomatisierung

Analog zum automatischen Deployment sollten auch die Tests weitestgehend automatisiert werden. Dies gilt insbesondere für Unit-Tests. Nicht oder nur schwer automatisiert werden können User Acceptance Tests, bei denen die Anwender prüfen müssen, ob eine bestimmte Funktionalität so umgesetzt wurde, wie definiert und erwartet.

Einfache Standardtests können und sollten aber bei jeder neuen Version wiederholt ablaufen, um sicherzustellen, dass sich die existierende Funktionalität nicht verändert hat. Sie können auch gemacht

werden um definierte Massentests vorzunehmen, die allein wegen des Umfangs schwer für den User durchzuführen sind.

Ein Tool, welches sehr einfach ermöglicht Testfälle für APEX Applikationen zu erzeugen und durchzuführen, ist SELENIUM. Dies ist eine weitverbreitete Open Source Software zum Testen von Webanwendungen. Die Komponente SELENIUM-IDE ist ein Plug für den Firefox Browser, die es erlaubt, Testfälle aufzuzeichnen und abzuspielen.

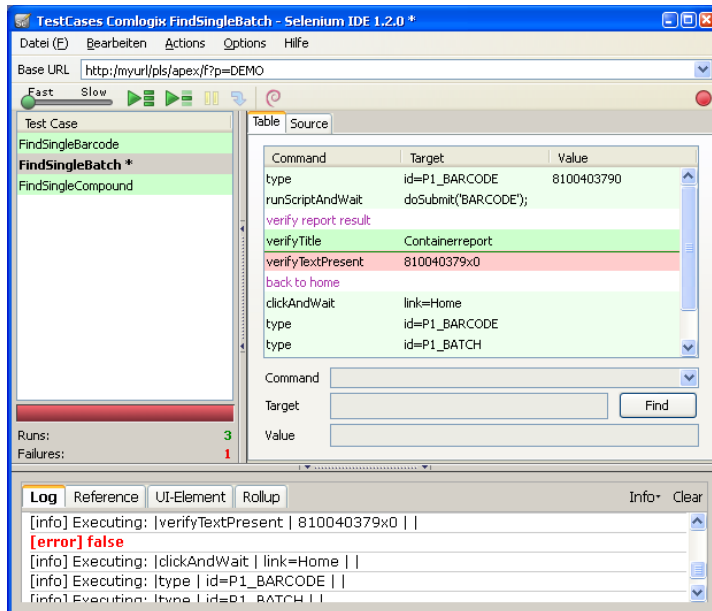


Abb. 3: Beispiel Selenium IDE- Testsuite mit drei Testcases

Wenn diese Testfälle, bzw. Testsuiten einmal angelegt sind, können sie auch automatisch mittels Hudson abgespielt werden.

Die Kombination aus automatisiertem Deployment und automatisierten Tests kann man schon als Continuous Integration Testing bezeichnen.

Weitere Tipps

Es gibt noch weitere Empfehlungen, wie man den Lebenszyklus von APEX Applikationen positiv beeinflussen kann.

Ein wichtiger Punkt dabei ist das Account Management. Insbesondere bei größeren Unternehmen sollte zur Authentifizierung niemals das Rad neu erfunden werden. Entweder man koppelt die APEX Applikation an eine existierende andere Anwendung oder man verwendet LDAP. Gut ist eine Kopplung an eine spezielle LDAP/AD Gruppe. Sobald ein Anwender in diese Gruppe aufgenommen wird, hat er damit automatisch Zugriff auf die Applikation. Da Entwickler allerdings normalerweise nicht die gleichen Gruppenrechte haben, empfiehlt es sich dafür die normalen APEX Rechte zu verwenden. Ich habe in meinen Projekten eine Package geschrieben, die beide Authentifizierungsmethoden miteinander kombiniert.

Damit die Applikationen kontinuierlich verbessert werden können, empfiehlt sich auch die Verwendung des Feedback Features. Dieses ist ab APEX 4.0 verfügbar. Der Anwender kann damit schnell und einfach in die Lage versetzt werden, Rückmeldungen zu geben. Im Vortrag wird kurz demonstriert, wie so ein Feedback aussieht und welche Informationen dabei automatisch mitgesammelt werden.

Kontaktadresse:

Sven-Uwe Weller
syntegris information solutions GmbH
Hermannstraße 54-56
D-63263 Neu-Isenburg

Telefon: +49 (0) 6102-298668
Fax: +49 (0) 6102-558806
E-Mail sven.weller@syntegris.de
Internet: www.syntegris.de