

Agile Softwareentwicklung mit Oracle ADF

Robert Szilinski
esentri consulting GmbH, Ettlingen

Andreas Koop
enpit consulting OHG, Paderborn

Schlüsselworte:

Agilität, SCRUM, Oracle ADF, Continuous Integration

Einleitung

Agile Methoden, wie beispielsweise Scrum, haben sich längst etabliert um Qualität zu sichern, Arbeitskraft optimal einzusetzen und die Komplexität von Projekten durch iterative Vorgehensweise beherrschbar zu gestalten. Dabei ist Scrum grundsätzlich sehr flexibel einsetzbar und kommt mit nur wenigen Regeln aus.

Dennoch muss neben der Methodik auch die Verbindung zur Technik funktionieren, damit effektiv agil entwickelt werden kann. Oracle ADF eignet sich zunächst nicht optimal für agile Projekte und es stellt sich die Frage, wie sich sowohl die Qualität als auch die Produktivität bei Oracle-Webprojekten steigern lässt und man gleichzeitig auf bewährten Industriestandards aufbauen kann.

Agilität mit Scrum ist heute vielfach „gesetzt“

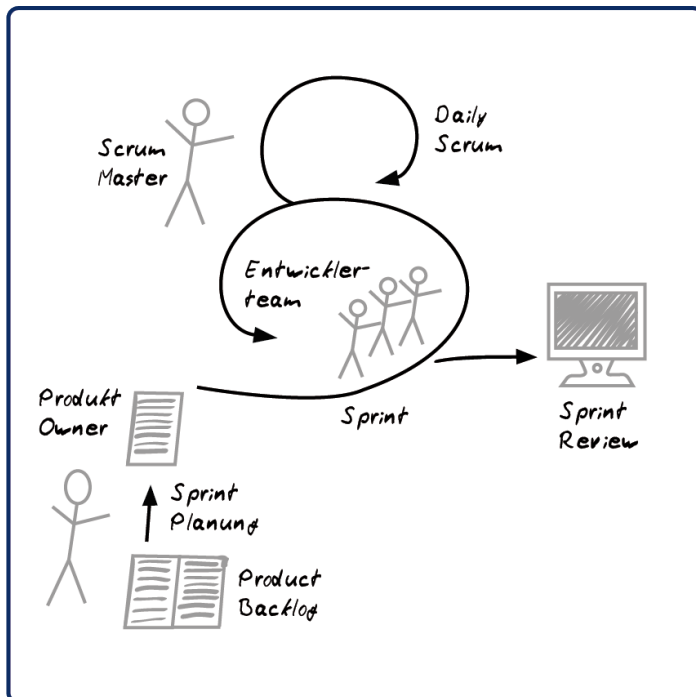


Abbildung 1: Typischer Scrum Prozess

Der oben abgebildete SCRUM-Zyklus stellt den Kern der SCRUM-Vorgehensweise dar. Alle Anforderungen, Features und Beschreibungen zu dem zu Implementierenden Projekt werden in dem so genannten „Produkt-Backlog“ gesammelt.

Ausgehend davon werden diese Anforderungen als so genannte „Sprints“ im „Sprint-Backlog“ gesammelt. Ein Sprint ist ein bestimmter zum Projekt gehörender Teilbaustein, welcher bearbeitet werden muss. Dabei muss bei der Definition der Sprints darauf geachtet werden, dass die Bearbeitung in einem Zeitraum von 2-4 Wochen erfolgen kann.

Weiterhin werden die gesammelten Sprints priorisiert. Anschließend können die wichtigsten zuerst abgearbeitet werden. Damit vermeidet man das Scheitern der essentiellen Bausteine durch Zeitknappheit gegen Ende des Projekts, weil diese im schlechtesten Fall nicht oder nur oberflächlich implementiert werden können.

Die SCRUM-Methodik setzt auf eine inkrementelle Abarbeitung des Projekts. Oft kommt es vor, dass die Gesamtkomplexität eines Projekts und die Fülle der Aufgaben das Projekt scheitern lassen, oder erhebliche Kompromisse eingegangen werden müssen, da die Strukturen zu verworren sind. Dieses Problem wird durch die Aufteilung des Projekts in kleine, abgeschlossene Teilaufgaben, den Sprints, gelöst.

Ziel nach der Bearbeitung jedes Sprints ist die Erstellung eines lauffähigen Release. Konkret bedeutet dies, dass nach jeder inkrementellen Verbesserung eine testfähige Version des Projekts entsteht. Dadurch vermeidet man böse Überraschungen gegen Ende des Projekts, wenn das System zum ersten Mal getestet wird und strukturelle Fehler auftreten. Die Beteiligten, vor allem die Auftraggeber, können so zu jedem Zeitpunkt den Erfolg und das Fortkommen des Projekts betrachten.

Bei jedem Sprint werden folgende Schritte durchlaufen:

1. Plan
2. Do
3. Check
4. Act

Die Qualität und ein stets lauffähiges System steht immer an oberster Stelle und durch diese Vorgehensweise wird nach jeden Teilbaustein geprüft, ob die Zielvorgabe erreicht wurde. Sollte diese nicht der Fall sein, geht es zurück zur Fehlerbehebung und die Schritte werden von vorne durchlaufen.

Ein großer Vorteil von SCRUM ist die Transparenz des Projektgeschehens. Alle Sprints werden auf dem SCRUM Board festgehalten und sind für jeden Beteiligten einsehbar. Durch diese Vorgehensweise wird SCRUM den ständig ändernden Anforderungen und Entwicklungen in heutigen Projekten gerecht.

Alle Beteiligten sind besser informiert, haben das „große Ganze“ im Blick und können ihren Standpunkt täglich dem Team mitteilen. Dies verbessert nicht nur das Arbeitsklima, sondern steigert die Produktivität nachhaltig und führt das Projekt zum Erfolg.

ADF - produktiv mit Oracle im Standard, jedoch mit Schwächen bei agiler Vorgehensweise

Oracle hat mit ADF und der perfekten Integration in die IDE JDeveloper Maßstäbe bei der JEE Entwicklung gesetzt. Seit geraumer Zeit können mit dem Oracle ADF Framework produktiv und konform zum JEE Standard komplexe Webanwendungen realisiert werden. Der Hauptvorteil von ADF liegt in der Toolunterstützung und der Integration in den Oracle Technologiestack. Dies gilt insbesondere für die Möglichkeit die komplette Persistenzschicht für bestehende Datenbankstrukturen zu generieren und mittels Data Bindings entsprechende Oberflächen per Drag & Drop zu designen. Die generelle Herangehensweise lehnt sich an der Entwicklung von Forms Anwendungen an und bietet zunächst auch unerfahrenen Java Entwicklern die Möglichkeit aufwendige JEE Anwendungen zu implementieren. Dabei werden große Teile des Codes durch Assistenten generiert und es ist dem Entwickler nicht immer sofort ersichtlich, wo sich XML Dateien oder Code-Bausteine geändert haben.

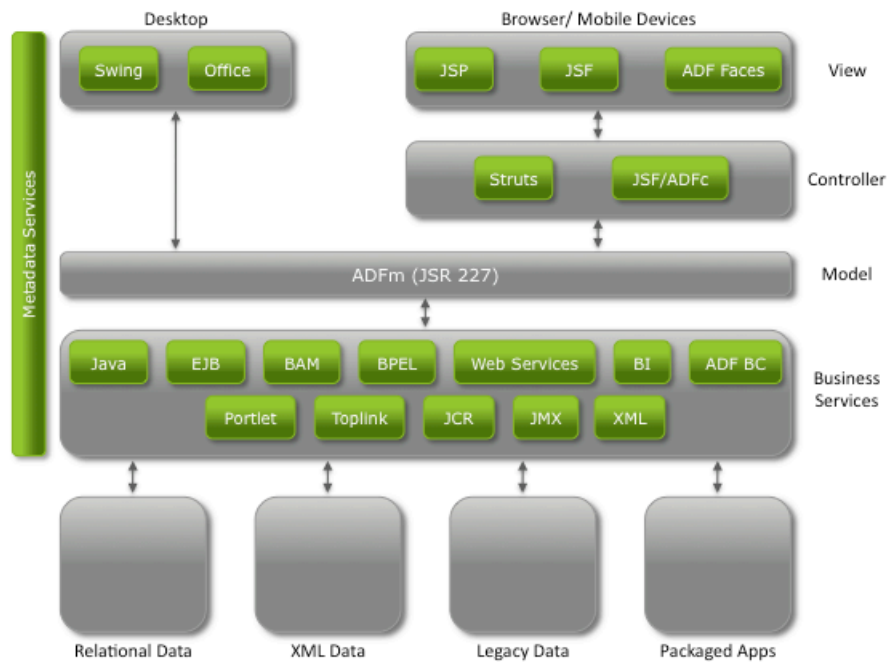


Abbildung 2: Die Architektur von Oracle ADF

Genau hier liegt jedoch auch das Problem bei einer agilen Vorgehensweise, die typischerweise nicht auf bestehenden Datenstrukturen aufbaut und Anforderungen häufig erst im Laufe des Projektes bekannt werden.

ADF ist für dieses Vorgehensweise nicht ausgelegt und spielt seine Stärken typischerweise genau dann aus, wenn analog klassischer Wasserfallmodelle die Anwendung im Vorfeld durch die Analyse und das Softwaredesign bereits ausreichend beschrieben wurde und entsprechende Datenstrukturen vorhanden sind.

Dennoch lässt sich beim Einsatz von Oracle ADF nach agilen Methoden entwickeln, indem man zusätzliche Tools nutzt und auf die Vorgehensweise im Projekt abstimmt. So wird die agile Entwicklung in optimaler Weise unterstützt, wenn primär sichergestellt ist, dass es jederzeit möglich ist ein lauffähiges Deployment des aktuellen Entwicklungsstands zu erzeugen.

Continuous Integration mit Oracle ADF als Add-On

Dies ist das Hauptziel der Continuous Integration (CI). Hierbei ist ein CI-Server, der nach jeder Änderung des Quellcodes die komplette Anwendung baut und testet, von zentraler Bedeutung. Dadurch wird sowohl die syntaktische und mittels automatische Unit-Tests, die funktionale Korrektheit der Anwendung geprüft. Darüber hinaus kann durch weitere automatisierte Checks (z.B. Sonar) die Qualität des erstellten Quellcodes analysiert und dokumentiert werden.

Es wird bei CI empfohlen, dass jeder Entwickler seine Änderungen so frühzeitig und so häufig wie möglich eincheckt. Man motiviert dabei vor allem große Änderungen über mehrere kleine Schritte umzusetzen und dabei gleichzeitig die Lauffähigkeit des Systems zu erhalten. Dies erfordert eine gewisse Disziplin seitens aller Entwickler, aber die Vorteile liegen auf der Hand:

- Integrationsprobleme werden frühzeitig entdeckt
- Unit-Tests entdecken Fehler durch Änderungen sehr früh
- Man erreicht eine hohe Verfügbarkeit des aktuellsten Systems, was in vielerlei Hinsicht hilfreich ist (Demonstrationen, Tests, Vertriebszwecke, ...)
- Fehler wie z.B. vergessene Dateien beim Check-In werden sofort erkannt
- Man kann jederzeit vor einem Release gut abschätzen, wie weit fortgeschritten der aktuelle Stand ist
- Probleme bei zu weit auseinanderlaufenden Entwicklungssträngen werden vermieden

Zusammenfassung

Viele Projekte werden heute agil implementiert und SCRUM hat sich dabei als erfolgreiches Konzept etabliert. Oracle ADF ist das Entwicklungsframework für die Oracle Middleware und der de facto Standard für Webanwendungen im Oracle Umfeld. Leider ist Oracle ADF nicht für die agile Entwicklung konzipiert worden und auch der JDeveloper bietet von Haus aus keine ausreichende Unterstützung für agile Methoden.

Durch zusätzliche Tools, wie z.B. Hudson/Jenkins gemeinsam mit Maven/Ant können die Schwächen von ADF teilweise ausgeglichen werden und auch größere Teams werden in die Lage versetzt auch agile Projekte zu implementieren, auch wenn dies nach wie vor mit einigen Einschränkungen verbunden ist.

Kontaktadresse:

Robert Szilinski
esentri consulting GmbH
Pforzheimer Straße 132
76275 Ettlingen

Telefon: +49 (0) 7243-354 90 0
E-Mail robert.szilinski@esentri.com
Internet: www.esentri.com

Andreas Koop
enpit consulting OHG
Theodor-Heuss-Straße 17c
33102 Paderborn

Telefon: +49 (0) 5251-202 77 91
Email andreas.koop@enpit.de
Internet: www.enpit.de