

Agile BI - Was ist das eigentlich?

Volker Herbort, M. Sc. & Prof. Dr. Reinhold von Schwerin
Hochschule Ulm
Ulm

Schlüsselworte:

Agile BI, Agile Data Warehousing, BI Sandboxing, Scrum, Agile Analytics

Einleitung

In letzter Zeit werden immer häufiger Artikel über agile Business Intelligence (BI), Agile Data Warehousing und Agile Analytics in Fachmagazinen und auf Konferenzen veröffentlicht. Leider kommt es immer wieder zu Vermischungen und Verwechslungen der Begriffe. So stellen sich viele die Frage in welcher Beziehung diese Begriffe zu den schon länger bekannten agilen Vorgehensweisen aus dem Bereich der Softwareentwicklung stehen? Der Vortrag möchte den BI-Anwendern dabei helfen die Begriffe zu ordnen und ein Gesamtbild zu erstellen. Der Schwerpunkt des Vortrags liegt auf der Veranschaulichung der Anwendbarkeit der agilen Projektmanagement-Ansätze aus der Softwareentwicklung in BI-Projekten sowie der Begründung für die Notwendigkeit neuer BI-Vorgehensmodelle und -Architekturen. Außerdem werden die Methoden für agiles Datawarehousing beleuchtet und dargestellt wie Sandboxing BI-Systemen Agilität verleihen kann.

Der Bedarf nach Agilität in der BI

BI-Projekte haben viele Gemeinsamkeiten mit Software (SW)-Projekten. Die klassischen Projektmanagementmethoden lassen sich in BI-Projekten genauso schwierig anwenden wie in Softwareprojekten. Dies hat mehrere Ursachen.

Einer der Hauptgründe für die Einführung der agilen Vorgehensmodelle (wie z.B. Scrum) in der Softwareentwicklung war, dass die traditionellen Modelle, etwa das Wasserfallmodell, darauf basieren, dass bereits zu Beginn eines SW-Projekts alle Anforderungen an das System bekannt sind. Diese Voraussetzung hat sich in der Praxis als nicht gegeben erwiesen und dazu geführt dass sehr viele SW-Projekte weder das eingeplante Budget noch den vereinbarten Termin einhalten konnten.

Viele BI Projekte stehen vor dem gleichen Problem, da sich die Anforderungen an Auswertungen während des Projekts ändern und andere, anfangs noch unbekannte Auswertungen wichtiger werden, als die ursprünglich geplanten. In diesem Fall hilft es den Anwendern wenig, wenn die geplanten Auswertungen umgesetzt werden. Es muss also eine Möglichkeit gefunden werden, neue Anforderungen oder geänderte Anforderungen zu berücksichtigen.

SW-Projekte mit Scrum

In SW-Projekten mit Scrum wird das Problem der sich ändernden Anforderungen wie in Abbildung 1 dargestellt gelöst. Alle Anforderungen werden von einem fachlichen Verantwortlichen, dem *Product Owner*, in einem *Product Backlog* gesammelt. Die Anforderungen werden oft als sogenannte *User Stories* im Format „Als *Rolle* kann ich *Funktion* x ausführen um *Nutzen* y zu haben“ erfasst. Das Product Backlog ist vom Product Owner in der Art priorisiert worden, dass die Anforderungen, die

den höchsten Nutzen bringen, an oberster Stelle stehen. Sobald neue Anforderungen identifiziert wurden, werden sie in das Product Backlog aufgenommen und je nach Priorität eingeordnet. Scrum arbeitet mit Iterationen, den sogenannten *Sprints* im Zyklus von meistens 1 - 4 Wochen. Vor einem Sprint werden in einem *Sprint Planning Meeting* die im Sprint umzusetzenden User Stories ausgewählt und vom Team im Detail ausgearbeitet. Die Auswahl geschieht in Abstimmung mit dem Product Owner und beinhaltet üblicherweise die am höchsten priorisierten Anforderungen. Anschließend werden die User Stories ins *Sprint Backlog* aufgenommen und herunter gebrochen auf Aufgaben, deren Umfang nicht größer als ein Arbeitstag sein sollte. Außerdem wird für jede Anforderung und auch jede daraus resultierende Aufgabe eine sogenannte *Definition of Done (DoD)* festgelegt. Die DoD legt fest, wann eine Aufgabe beendet ist, sie beinhaltet unter anderem die Festlegung von funktionalen Tests und User Acceptance Test. Auf diese Weise werden also immer nur die Anforderung detailliert, umgesetzt und getestet, die aktuell am wichtigsten sind.

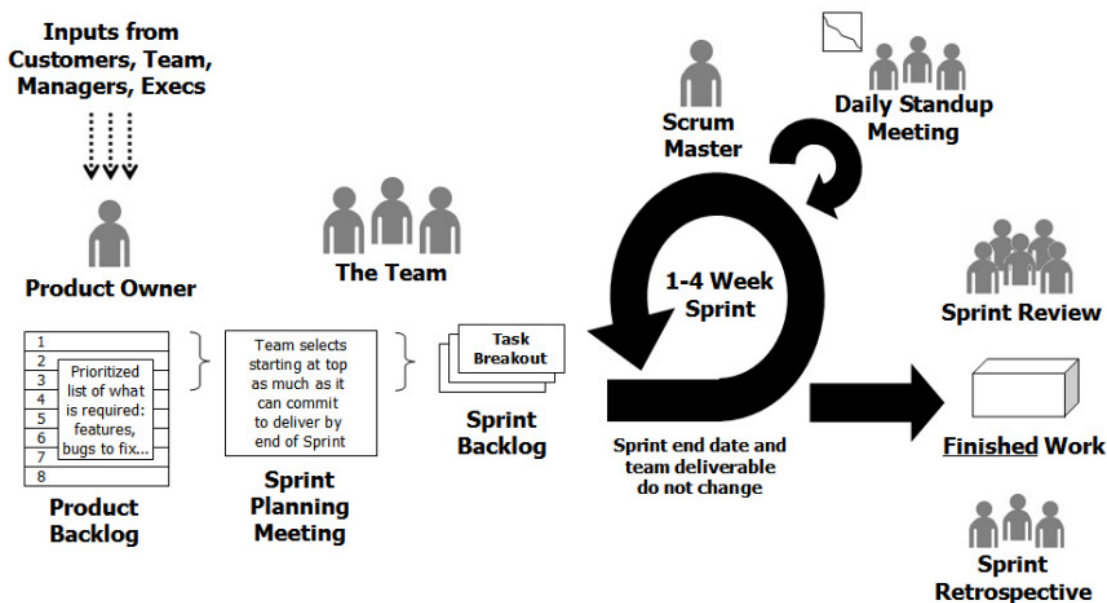


Abb. 1: Scrum (Quelle: Scrum Primer, Pete Deemer & Gabrielle Benefield, Yahoo, 2006)

Während des Sprints arbeitet das Team die Aufgaben ab und stimmt sich in fest terminierten täglichen kurzen Treffen, dem *Daily Scrum* ab. Am Ende des Sprints entsteht ein Produktinkrement, das der Product Owner während des *Sprint Reviews* abnimmt und das bereits getestet und nutzbar ist (*Potentially Shippable Product Increment*). Auf diese Weise bekommt der Product Owner bereits nach wenigen Iterationen ein Teilsystem, das ihm einen Nutzen bringt. Im Gegensatz dazu findet eine „Auslieferung“ in klassischen Projekten meist erst gegen Ende des Projektes statt.

Innerhalb eines Scrumprojektes gibt es keinen Projektmanager mehr. Dafür gibt es eine Art „Schiedsrichter“, den sogenannten *Scrum Master*. Er kümmert sich um organisatorische Probleme und überwacht die ordnungsgemäße Zusammenarbeit im Team und zwischen Team und Product Owner. Außerdem leitet er die *Sprint Retrospective*, während der Probleme des Entwicklungsprozesses aufgedeckt und Lösungen erarbeitet werden. Die Retrospektive sorgt für eine kontinuierliche Verbesserung des Vorgehens und eine Überwachung der Einhaltung der Vorgaben des Scrum Prozesses.

Der große Vorteil der agilen Vorgehensmodelle ist, dass nur das entwickelt wird, was der Endanwender wirklich benötigt. Außerdem können Teile der Funktionalitäten schon während des Projektes vom Endanwender genutzt werden. Aber sind auf diese Weise überhaupt noch die weithin üblichen Festpreisprojekte möglich?

In der Tat: Festpreisprojekte mit Scrum basieren auf einem initialen Product Backlog, dessen Umfang als Grundlage für die Schätzung des Projektaufwands genommen wird. Sollten sich nun neue Anforderungen ergeben, müssen diese ins Produkt Backlog aufgenommen und priorisiert werden. Am Ende des Projektbudgets oder zur Zeit des vereinbarten Liefertermins werden daher nicht alle initialen Anforderungen umgesetzt worden sein. Es ist daher, analog zum Baurecht, ratsam, nach jeder Iteration eine Teilabnahme durchzuführen.¹ Aber durch die ständige Priorisierung kann sichergestellt werden, dass die wichtigsten Funktionen bereits nutzbar sind. Nun kann entschieden werden, ob sich der weitere Aufwand für die noch verbliebenen Anforderungen rentiert oder ob die erstellte Software ausreichend ist und das Projekt beendet werden kann.²

Agile Data Warehousing

Die Übernahme der aus der SW-Entwicklung bekannten Ansätze findet sich im sogenannten *Agile Data Warehousing* wieder.³ Hierbei soll das Data Warehouse, welches die Datenbasis der meisten BI-Systeme hält, mit Hilfe der agilen Vorgehensweisen entwickelt und weiterentwickelt werden.

Die User Stories in BI-Projekten beschreiben in den meisten Fällen Auswertungen von Daten. Diese Auswertungen sollen beim Agile Data Warehousing auf iterative Weise, Schritt für Schritt, in einem

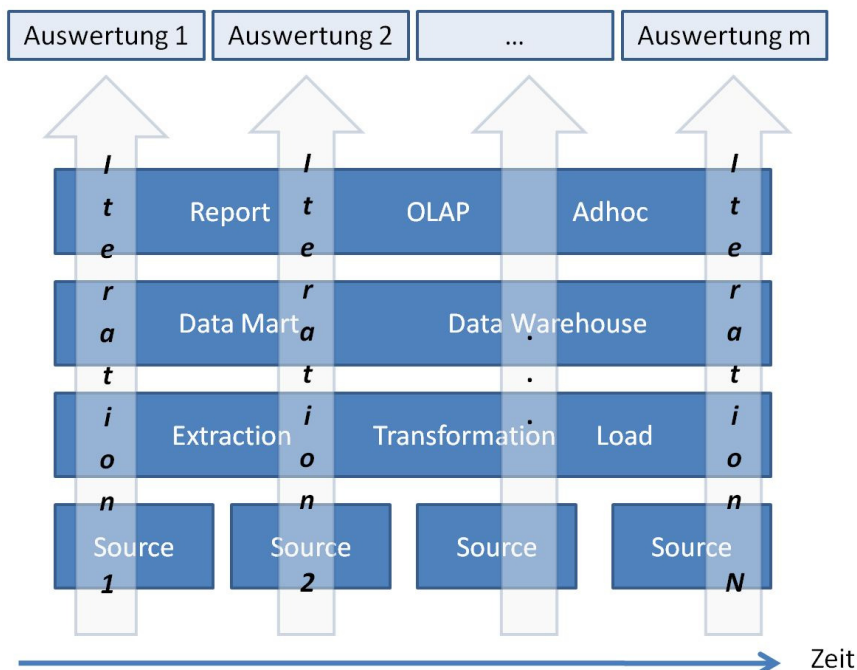


Abb. 2: Agile Data Warehousing

¹ Die „eingebaute Rechtssicherheit“ von Scrum, M.A., Hofmann, Scrum-Day 2010

² Der agile Festpreis, A. Roock, Javamagazin 6/2010

³ Agile Data Warehousing-Delivering World-Class Business Intelligence Systems using Scrum and XP, R. Hughes, iUniverse 2008

Data Warehouse abgebildet und die notwendigen Datenstrukturen gegebenenfalls entwickelt werden. Abbildung 2 zeigt wie Agile Data Warehousing im Bezug zur üblichen BI-Architektur steht.

Die Idee vom Agile Data Warehousing ist, dem Auftraggeber möglichst früh bereits nutzbare Auswertungen zur Verfügung stellen. Diese Idee steht im Widerspruch zur klassischen Vorgehensweise, bei der zunächst die Quelldaten für alle benötigten Analysen in einem zentralen Data Warehouse zusammengefügt werden und erst im Anschluss daran die Auswertungen implementiert werden. Die Zusammenführung der Daten benötigt den Großteil der Zeit in BI-Projekten. Viele Projekte geraten gerade bei der Konzeption eines Data Warehouses ins Stocken, da keine Einigung über Namensgebungen und/oder Details der Implementierung erzielt werden kann.⁴ Diese Details werden bei Agile Data Warehousing Just-In-Time entschieden.

Aufgrund der Analogie zur Softwareentwicklung eignet sich Scrum sehr gut zur Durchführung von agilen BI Projekten. Unter Agile Data Warehousing wird also die Anwendung von agilen Methoden zur Erstellung von BI Architekturen verstanden.

Agile Analytics

Agile Analytics hat das gleiche Ziel wie Agile Data Warehousing, unterscheidet sich jedoch in der Umsetzung. Bei Agile Analytics stehen sehr kurzfristige Analysen im Vordergrund, für die eine Iteration im Agile Data Warehousing bereits viel zu lange dauern würde. Die Idee hierbei ist es bestimmten BI-Anwendern, den sog. *Knowledge Workern*, zu ermöglichen, innerhalb kürzester Zeit von Quelldaten zu einer einfachen Repräsentation in Form eines Dashboard oder Reports zu kommen. Alternativ wird hierfür auch der Begriff *Self-Service BI* verwendet. Abbildung 3 erläutert die Eingliederung von Agile Analytics in das BI Umfeld. Um den Prozess von den Daten bis zur Auswertung durchlaufen zu können, müssen im Wesentlichen zwei Voraussetzungen gegeben sein.

Als erstes muss die BI-Infrastruktur es dem Anwender erlauben, möglichst schnell den Weg von Daten zur Repräsentation der Daten in einer Auswertung gehen zu können. Die Struktur der Datenhaltung muss also flexibel genug sein, um zusätzliche Auswertungen zuzulassen. Die Verwendung einer solchen flexiblen Datenstruktur wird teilweise auch mit Agile Data Warehousing bezeichnet. Da sich diese Datenstrukturen allerdings teilweise Grundlegend von den Ansätzen eines Data Warehouse unterscheiden können, wäre sicherlich ein Begriff wie „Agile Data Provision“ sinnvoller.

Ein Lösungsansatz zur Flexibilisierung von Auswertungen bietet die Hinzunahme von Metadaten über Geschäftsprozesse, die es dem Endanwender ermöglicht zumindest einen Großteil der mit dem aktuellen Datenbestand möglichen Auswertungen, anhand der Metadaten flexibel zusammenzustellen.

Für Analysen, die auf neuen Daten basieren, hat sich der sogenannten Sandboxing Ansatz⁵ bewährt. Eine Sandbox kann z.B. ein parallel zum produktiven Data Warehouse aufgebautes System sein, welches Datenänderungen durch den Endbenutzer erlaubt. Die Trennung vom produktiven Data Warehouse ist aus mehreren Gründen sinnvoll. Neben der Sicherung des aktuellen Standes, kann auf diese Weise auch eine zu hohe Belastung des produktiven BI-Systems vermieden werden, die z.B. durch ETL-Prozesse entstehen kann. Idealerweise ist das Sandboxsystem eine im Umfang eingeschränkte Kopie der produktiven BI-Umgebung, denn dadurch ist es im Anschluss an eine

⁴ Agile Best Practices for Data Warehousing (DW)/Business Intelligence (BI) Projects, S. Ambler, <http://www.agiledata.org/essays/dataWarehousingBestPractices.html>

⁵ Agile Data Warehousing with Integrated Sandboxing, S. Brobst, M. Mcintire & E. Rado, Business Intelligence Journal, Volume 13, No 1, 1st Quarter 2008

erfolgreiche Auswertung einfacher möglich die Auswertung in das produktive BI-System zu übernehmen. Diese Kopie sollte regelmäßig synchronisiert werden, damit nicht benötigte Strukturen wieder entfernt werden.

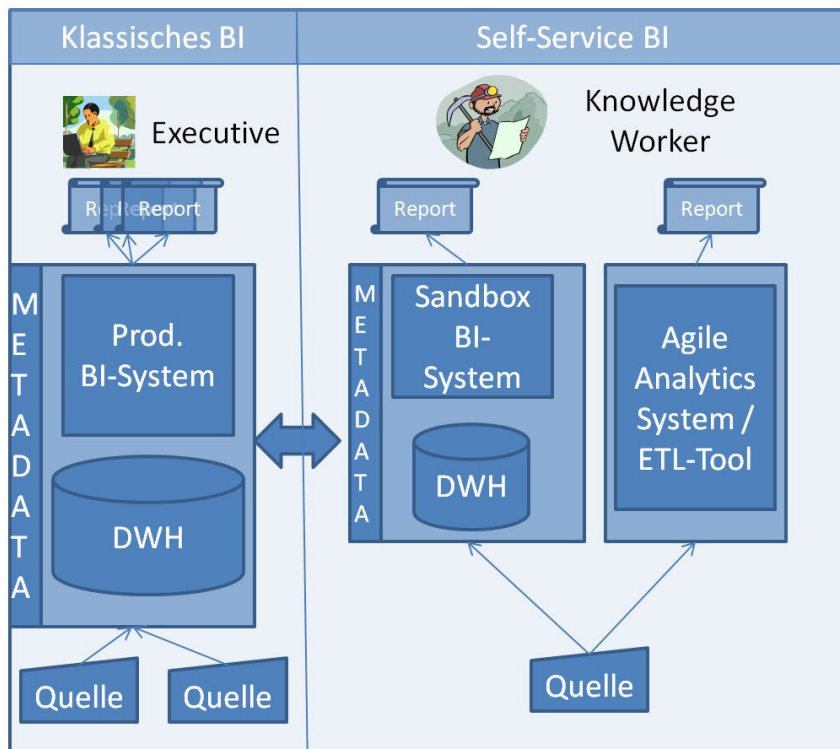


Abb.3: Agile Analytics / Self-Service BI

Eine Alternative stellen BI-Systeme dar, die es ermöglichen Auswertungen ohne ein Data Warehouse auf Basis von Dateien zu machen. Diese Systeme basieren meist auf ein ETL-Werkzeug, das gleichzeitig die Möglichkeit bietet erste Auswertungen vorzunehmen.⁶

Die zweite Voraussetzung für die Anwendung von Agile Analytics betrifft den Knowledge Worker. Für die Durchführung von Analysen ist es unerlässlich, dass der Anwender sowohl tiefgehende IT-Kenntnisse als auch eine detaillierte Kenntnis der Geschäftsprozesse und derer Daten hat. Laut einem Bericht von Forrester Research⁷ ist diese Situation durchaus gegeben. Die klassische Trennung von IT und Geschäftsprozessen gibt es in vielen Unternehmen nicht mehr. Dies könnte unter anderem an der Einführung von interdisziplinären Studiengängen wie z.B. der Wirtschaftsinformatik liegen. Um den Prozess von der Datenbereinigung bis zur Auswertung durchführen zu können, muss der Anwender neben dem fachlichen Verständnis sowohl statistische Kenntnisse als auch Datenbank- und SQL-Kenntnisse aufweisen.

Zusammenfassung

Die Begriffe Agile Data Warehousing und Agile Analytics werden oft gleichwertig verwendet. Es ist allerdings richtiger diese beiden Begriffe unter dem Sammelbegriff *Agile Business Intelligence*

⁶ Enabling Agile BI with a Compressed Flat File Architecture, W. Sunna, P. Agrawal, Business Intelligence Journal, Volume 15, No 2, 2nd Quarter 2010

⁷ Agile BI – Best Practices for Breaking through the BI Backlog, Forrester Research, April 2010

zusammenzufassen und eine klare Trennung für die Begriffe Agile Data Warehousing und Agile Analytics einzuhalten wie sie in Tabelle 1 dargestellt ist.

Agile Business Intelligence	
Agile Data Warehousing / Vorgehensweise	Agile Analytics / Architektur
Anwendung agiler Vorgehensmodelle	Datenbasierte Architektur
Iterative Erstellung von Auswertungen	Flexible Auswertungsmöglichkeiten
Erstellung von Prototypen und nutzbaren Teillösungen	Schnelle Aufbereitung von Daten zur Darstellung von Auswertungen
Details werden durch Kommunikation festgelegt	Sandboxing

Tab.1: Agile Data Warehousing vs. Agile Analytics

Agile Data Warehousing beschreibt die Vorgehensweise bei der Erstellung einer BI-Lösung und Agile Analytics beschreibt die Werkzeuge und Techniken, die eingesetzt werden können, um einem Endbenutzer möglichst flexible Analysen zu ermöglichen. Beides gemeinsam macht also Agile BI aus und erlaubt auf schnellen Nutzen ausgerichtete, flexible aber dennoch disziplinierte⁸ Abwicklung von BI-Projekten; insgesamt also eine bedarfsorientierte professionelle Herangehensweise.

Kontaktadressen:

Volker Herbort
Hochschule Ulm
Prittwitzstrasse 10
D-89075 Ulm

+49 (0) 731-50 28495
herbort@hs-ulm.de
<http://www.hs-ulm.de>

Reinhold von Schwerin
Hochschule Ulm
Prittwitzstrasse 10
D-89075 Ulm

+49 (0) 731-50 28259
r.schwerin@hs-ulm.de
<http://www.hs-ulm.de>

⁸ Balancing Agility and Discipline: A Guide for the Perplexed .B. Boehm, R. Turner. Addison-Wesley, 2008.