

Scrum für Business Intelligence und Data-Warehouse Projekte

Thomas Löchte
Informationsfabrik GmbH
Münster

Schlüsselworte

Scrum, Agile BI, Agile DWH, Vorgehensmodell,

Einleitung

Agile Vorgehensweisen sind in der klassischen Softwareentwicklung seit vielen Jahren fest etabliert. Seit ca. zwei Jahren nutzen auch mehr und mehr Data Warehouse und BI Projekte agile Entwicklungsmethoden, mehr noch Agile BI ist inzwischen neben Big Data DAS Trendthema auf Kongressen, in Blogs und anderen Veröffentlichungen.

In diesem Vortrag werde ich eine kurze Einführung des agilen Rahmenwerks Scrum geben und den Scrum Prozess, die Scrum Artefakte und Rollen erläutern.

Darauf aufbauend werden die Besonderheiten von Data Warehouse und BI Projekten hinsichtlich beschrieben und die damit einhergehenden Maßnahmen mit Scrum erläutert.

1 Was ist Scrum?

Scrum ist ein agiles Projektmanagement Rahmenwerk, das eine Vorgehensweise für Projekte beschreibt. Scrum wird häufig für Produktentwicklungen und insbesondere in der Softwareentwicklung eingesetzt.

Mit Scrum wird auf die wichtigsten Geschäfts-Anforderungen fokussiert. Das Business priorisiert die umzusetzenden Funktionen und legt somit die Reihenfolge der Entwicklung fest. Selbst-organisierende Entwicklungsteams bestimmen das beste Vorgehen für die Umsetzung.

Mit Scrum wird in kurzen und regelmäßigen Abschnitten (von 2-4 Wochen) lauffähige Software erstellt.

Der Scrum Prozess ist so einfach, dass er auf einem Bierdeckel beschrieben werden kann.

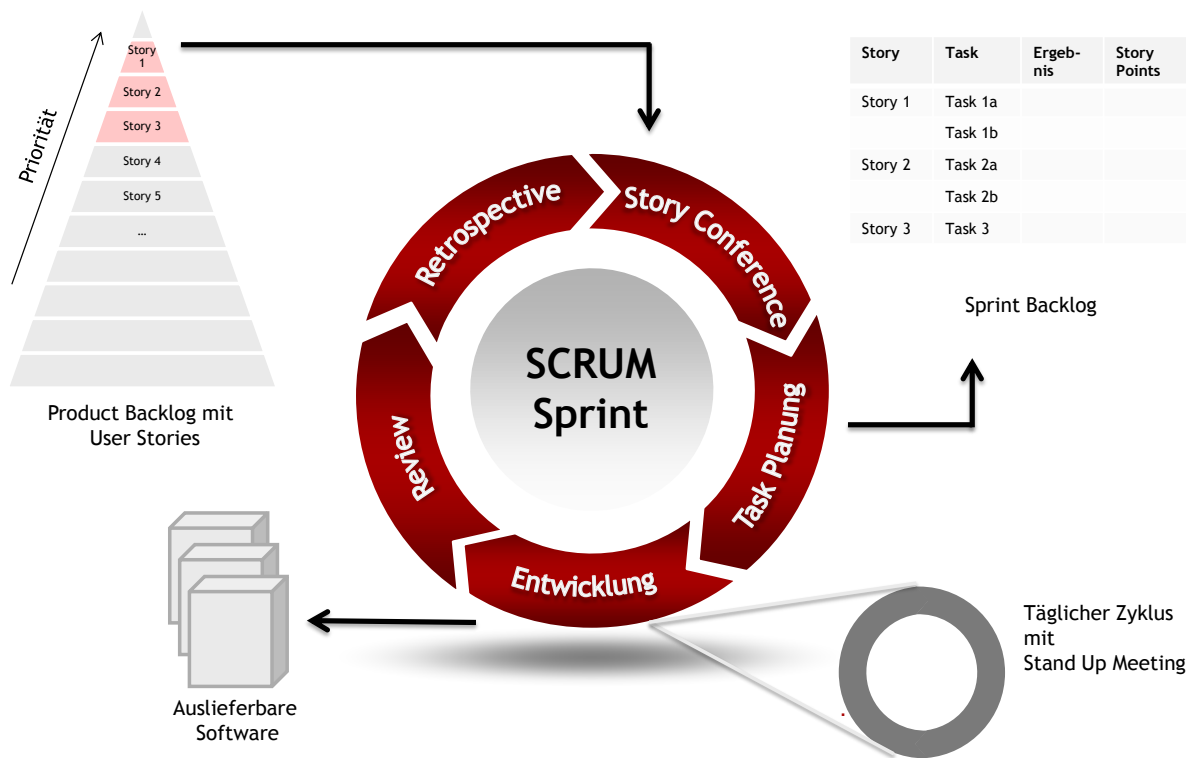


Abbildung 1: Scrum Prozess

1.1 Rollen

In Scrum gibt es nur drei Rollen:

Der **Product Owner (PO)** erstellt die Product Vision, also das übergreifende Ziel des Projektes. Weiterhin definiert er die Produkt-Features, priorisiert diese und akzeptiert die Umsetzung der Funktionen. Er ist verantwortlich für den Erfolg (ROI) des Projektes.

Das **Team** besteht typischerweise aus 5-9 Personen und ist funktionsübergreifend besetzt. Alle für das Projekt benötigten Skills sollten im Team vorhanden sein. Das Team agiert selbstorganisiert und arbeitet so weit wie möglich autonom.

Der **Scrum Master** ist verantwortlich für die Einhaltung von Scrum-Werten und -Techniken. Zu seinen Aufgaben gehört die Beseitigung von Hindernissen. Er muss dafür sorgen, dass das Team ohne Störungen arbeiten kann. Der Scrum Master ist Coach für das Team und für den PO. Darüber hinaus vertritt er die Scrum Werte gegenüber dem Management und hilft dem Management, den PO einzusetzen.

In jedem Projekt gibt es natürlich noch weitere Personen (**Stakeholder**), die berücksichtigt werden müssen. Dies sind auf der einen Seite die Anwender des Produktes. Auf der anderen Seite sind dies Abteilungen und Bereiche, die Abhängigkeiten zum Produkt oder Projekt haben sowie das Management.

Die Bedürfnisse der Stakeholder zu befriedigen ist Aufgabe des Product Owners. Der Scrum Master ist dafür verantwortlich, den Scrum-Prozess an die Stakeholder zu kommunizieren und bei Bedarf im Detail zu erklären.

1.2 Artefakte

Scrum gibt sehr wenige Artefakte vor. Diese wenigen Artefakte werden genutzt, um die Anforderungen an das Produkt zu beschreiben und zu priorisieren. Darüber hinaus dienen Sie dem Team als Hilfsmittel für die Umsetzung.

Die **Product Vision** ist eine kurze Beschreibung des Vorhabens und enthält die wesentlichen Ziele, die mit dem Vorhaben erreicht werden sollen.

Das **Product Backlog** ist eine priorisierte Liste der Anforderungen an das Produkt. Anforderungen werden in User Stories beschrieben. Idealerweise soll jeder Eintrag wertvoll für Benutzer des Produktes oder für den Kunden sein. Das Product Backlog wird laufend um neue Anforderungen (User Stories) erweitert und vor Beginn eines jeden Sprints durch den Product Owner neu priorisiert. Der Aufwand der User Stories im Product Backlog soll durch das Team initial geschätzt werden. Das Product Backlog gehört dem Product Owner.

Eine User Story ist kurze Spezifikation einer Anforderung. Die Umsetzung einer Story soll in wenigen Tagen erfolgen können. Manchmal wird zwischen fachlichen und technischen User Stories unterschieden.

Eine User Story wird wie folgt beschrieben:

Als User (Wer?) möchte ich diese Funktionalität (Was?), damit ich folgenden Nutzen habe (Wozu?).

Beispiel:

Als Controller möchte ich in der Deckungsbeitragsrechnung bis auf den Einzelkunden herunterdrillen können, um Kunden mit negativem Deckungsbeitrag zu identifizieren.

Das **Sprint Backlog** enthält diejenigen User Stories, die im Rahmen des aktuellen Sprints umgesetzt werden. Die umzusetzenden User Stories werden i.d.R. in kleine, umsetzbare Aufgaben (Tasks) heruntergebrochen. Das Sprint Backlog gehört dem Team.

Zur Unterstützung des Teams kann ein **Burndown Chart** genutzt werden, um die Fertigstellung der Aufgaben im Sprintverlauf zu visualisieren. Das Burndown Chart sollte täglich aktualisiert werden und gehört dem Team.

Das **Impediment Backlog** listet alle Hindernisse (Impediments) auf, die im Rahmen der Sprints aufgetreten sind. Die aufgeführten Hindernisse müssen so schnell wie möglich beseitigt werden.

Die **Definition of Done** beschreibt, wann eine Story, ein Sprint oder ein Release tatsächlich fertig gestellt ist. Die Definition of Done kann im Laufe des Projektes angepasst oder erweitert werden.

1.3 Phasen des Scrum Prozesses und Meetings

1.3.1 Sprint Planning I und II

Am Anfang eines Sprints wird das Sprint Planning durchgeführt. Es ist zweigeteilt. Zuerst findet die Story Conference statt, in der die umzusetzenden User Stories besprochen und geschätzt werden. Diese Stories werden dann ins Sprint Backlog übernommen.

Hierzu stellt der Product Owner die Stories vor, die umgesetzt werden sollen. Er erläutert die Ziele und Anforderungen zu einer Story. Das Team klärt Details und bewertet die Anforderungen mit Story Points. Hierzu kann z.B. Planning Poker genutzt werden.

Wichtig ist, dass das Team Anforderungen und Ziele versteht.

Am Ende einigt sich das Team auf eine Menge von Stories, die im nächsten Sprint umgesetzt werden.

Darüber hinaus wird ein Sprint Ziel abgestimmt, dem sich alle Teilnehmer verpflichten.

In der darauffolgenden Task Planung (Sprint Planning II) werden die User Stories in umsetzbare Aufgaben heruntergebrochen.

1.3.2 Entwicklung mit Daily Scrum

Während der Entwicklungsphase wird täglich ein 15 minütiges Meeting mit dem Team durchgeführt. Ziel ist ein übergreifendes Verständnis über den Stand der Aufgaben herzustellen sowie die Identifizierung von Hindernissen.

Jeder Teilnehmer beantwortet die folgenden drei Fragen:

- Was habe ich gestern erledigt?
- Was werde ich heute erledigen?

- Welche Hindernisse müssen beseitigt werden?

Das Meeting dient lediglich zur Identifizierung von Problemen, nicht zu deren Lösung.

1.3.3 Sprint Review (Demo)

Am Ende des Sprints wird die erstellte Lösung dem Product Owner und anderen Stakeholdern vorgestellt. Das Team zeigt die umgesetzten Ergebnisse. Alle Anwesenden geben Feedback zur Umsetzung. Das Meeting dient zum übergreifenden Dialog zwischen Teams, PO und Stakeholdern.

Am Ende akzeptiert der PO die umgesetzten Stories oder lehnt sie ab.

1.3.4 Retrospektive

Mit Scrum soll sich das Team während des Projektes laufend verbessern. Hierzu ist die Retrospektive eine wichtige Veranstaltung.

Es wird besprochen was war gut war, was verbessert werden sollte und welche Hindernisse aus dem Weg zu räumen sind.

Das Team legt fest, welche Verbesserungen am wichtigsten sind und im nächsten Sprint umgesetzt werden.

In der Retrospektive muss der Scrum Master seine methodischen Fähigkeiten einsetzen um bis zum Kern der Probleme vorzudringen und eine stetige Verbesserung zu erzielen.

1.3.5 Backlog Refinement / Grooming

Bevor eine User Story in der Sprint Planung bearbeitet wird, ist es häufig sinnvoll, dass diese Story vorbesprochen wird. Hierzu können Backlog Refinement Meetings oder Backlog Groomings abgehalten werden.

2 Besonderheiten im Business Intelligence Umfeld

In Business Intelligence und Data Warehouse Umfeld gibt es verschiedenen Punkte, die im Vergleich zu klassischer Softwareentwicklung besonders hervorzuheben sind.

Architektur

Die Architektur eines BI oder DWH Projektes ist oft komplexer als in der klassischen Softwareentwicklung. Aufgrund der Tatsache, dass verschiedene Vorkomponenten in unterschiedlichen Formaten in ein Data Warehouse integriert werden, gibt es Verbindungen zu zahlreichen Systemen. Darüber hinaus werden oft viele unterschiedliche Werkzeuge eingesetzt (Datenmodellierung, ETL, Reporting, Data Quality, OLAP usw.).

Beteiligte

Ein Business Intelligence Projekt hat i.d.R. viele Beteiligte in unterschiedlichen Abteilungen oder unterschiedlichen Unternehmen. Es gibt Schnittstellen sowohl hinsichtlich der Datenversorgung als auch hinsichtlich der Nutzung der Business Intelligence Lösungen.

Komplexität für Automatisierung

In klassischen Softwareentwicklungsprojekten ist die Automatisierung der Stützprozesse wie Paketierung, Deployment und Test relativ einfach möglich. Die Toolunterstützung ist hier sehr gut. Es gibt heute kaum ein Java Projekt, das kein automatisches Deployment und keinen automatischen Build durchführt.

Diese Toolunterstützung fehlt im Business Intelligence Umfeld oder ist oft nicht ausgereift. Die Automatisierung der Stützprozesse muss meist im Rahmen des Projektes entwickelt werden.

Datengetriebener Test

Tests im BI Umfeld sind i.d.R. datengetrieben. Die Erstellung von Testdaten ist aufgrund der Integrationsaufgabe des DWH häufig aufwändig. Darüber hinaus sind die (anonymisierten) produktiven Datenmengen für Unit Tests zu umfangreich.

Entwickler sind eher Spezialisten

Bei Scrum wird eine kooperative Entwicklung mit interdisziplinären Teams durchgeführt. Hierzu sind Generalisten, die mehrere Tätigkeiten durchführen können wichtig. Durch die Heterogenität und Komplexität sind diese im DWH Umfeld seltener als in der klassischen SW Entwicklung. Oft ist ein Entwickler entweder Spezialist für ETL oder Reporting oder Datenmodellierung usw.

Refaktorisierung

Durch die Heterogenität und die komplexe Architektur können umfangreiche Refaktorisierungsmaßnahmen im Data Warehouse Umfeld sehr aufwändig und komplex werden.

3 Wie Scrum auch für Business Intelligence Projekte funktioniert

Scrum ist sehr gut für Business Intelligence Projekte geeignet. Damit ein Projekt erfolgreich ist, sollten aber einige Punkte berücksichtigt werden.

3.1 Anforderungsanalyse

Nach der reinen Scrum Lehre werden die umzusetzenden Anforderungen im Product Backlog aufgeführt. Das Product Backlog wird laufend erweitert, so dass zu Beginn eines Projektes nur die Anforderungen beschrieben sein müssen, die in den ersten Sprints umgesetzt werden.

Für Business Intelligence Projekte ist es allerdings wichtig bereits zu Beginn so viele Anforderungen zu kennen, dass fundierte Architekturentscheidungen getroffen werden können.

Es kann sinnvoll sein, eine fachliche Architekturskizze mit den verschiedenen Data Marts und BI Front Ends zu erstellen.

Weiterhin ist bei der Beschreibung der Anforderungen zu beachten, dass die User Stories nicht zu groß geschnitten werden. Eine Anforderung wie in unserem Beispiel kann, wenn die Datenbasis noch nicht zur Verfügung steht, möglicherweise nicht innerhalb eines Sprints umgesetzt werden. Zu große Stories sollten aufgeteilt werden. Um eine Bandbreite für die Anzahl der Stories pro Sprint zu nennen: die Zahl sollte zwischen 3-10 Stories liegen.

(Beispiel: Als Controller möchte ich in der Deckungsbeitragsrechnung bis auf den Einzelkunden herunterdrillen können um Kunden mit negativem Deckungsbeitrag zu identifizieren.)

3.2 Architektur

Aufgrund der Tatsache, dass Architekturänderungen im Business Intelligence Umfeld sehr aufwändig sein können, ist es wichtig, sehr früh eine grundlegende Architekturbeschreibung anzufertigen. Zur Architektur gehören die verschiedene Datenschichten, die Art der Datenversorgung (Full Load, Delta, Batch, Real-Time), das Historisierungskonzept und eine erste Übersicht über die Datenquellen. Darüber hinaus sollten die verschiedenen Arten der Front Ends beschrieben sein. Es ist ein Unterschied, ob Sie die Daten „nur“ in Standardberichten oder auch für Data Mining, OLAP und Self-Service BI zur Verfügung stellen möchten.

Themen wie Datenqualität und Metadaten gehören ebenfalls in die Architekturbeschreibung.

Es gibt Architekturkomponenten, die sich für agile Projekte besser eignen als andere, da Veränderungen hier einfacher durchzuführen sind.

Die Architekturbeschreibung muss kein umfangreiches Dokument sein. Viel wichtiger ist, dass die Entscheidungen von allen verstanden wurden. Die Beschreibung ist auch nicht in

Stein gemeißelt und soll im Projektverlauf an die neuen Anforderungen angepasst bzw. ergänzt werden.

3.3 Frühzeitig automatisieren

Eine wesentliche Eigenschaft der agilen Entwicklung ist die regelmäßige Auslieferung funktionierender Software innerhalb kurzer Iterationen. Dies setzt voraus, dass die Entwickler sich auf die Weiterentwicklung konzentrieren können und Prozesse wie Build und Deployment weitgehend automatisch ausgeführt werden. Auch die Regressionstests müssen automatisiert werden.

Mit der Automatisierung sollte schon früh im Projekt begonnen werden, da ansonsten ganze Sprints nur für diese Themen benötigt werden.

3.4 Technische Stories oder technische Sprints

Trotz der erläuterten Punkte zum Thema Anforderungsanalyse, Architektur und Automatisierung kann es passieren, dass ein umfangreiches Refactoring notwendig wird. Z.B. weil neue, vorher nicht bekannte Anforderungen hinzugekommen sind oder weil sich viel technische Themen (technische Schulden) aufgrund der Priorisierung des Fachbereichs aufgebaut haben. Es ist wichtig, diese Themen zu bearbeiten. Aus diesem Grund sollten regelmäßig technische Stories oder gar ganze technische Sprints eingeplant werden. Auch wenn diese entgegen der Scrum Regeln im ersten Schritt keinen geschäftlichen Mehrwert bringen. Der Mehrwert ergibt sich aus der schnelleren Entwicklung oder aus der einfacheren Wartung.

3.5 Einbeziehung der Stakeholder

Eine Business Intelligence Lösung wird i.d.R. von verschiedenen Fachanwendern aus unterschiedlichen Abteilungen genutzt. Die Anwender haben häufig völlig unterschiedliche Anforderungen an das BI System.

Oft werden Berichte und OLAP Analysen direkt von den Fachanwendern erstellt. Diese sind nicht zwingend in den Scrum Prozess eingebunden und nicht Teil des Scrum Teams.

Es gibt also starke Abhängigkeiten zwischen dem Scrum Team, den einzelnen Abteilungen und den erstellten Lösungen.

Aus diesem Grund sollte für BI Projekte eine aktive Information aller Anwenderbereiche eingeplant werden. Ob dies durch regelmäßige Anwendertreffen, durch aktive Unterstützung der Anwender oder durch die Einladung zur Sprint Demo durchgeführt wird, ist vom Product Owner zu definieren. Der Product Owner ist verantwortlich für die Kommunikation mit den Stakeholdern und für die Aufnahme neuer Anforderungen.

3.6 Zuschnitt der Teams bei größeren Projekten

Die Größe von BI Projekten unterscheidet sich sehr stark. Kleine BI Projekte können von nur 1-2 Entwicklern realisiert werden, in Großprojekten sind mehr als einhundert Entwicklermöglich.

Bei großen Projekten ist es wichtig, den Zuschnitt der Teams so zu gestalten, dass diese möglichst unabhängig agieren können (was natürlich nie ganz möglich ist).

Darüber hinaus ist es wichtig, dass die einzelnen Teams zusammen sitzen sollten, optimalerweise in einem Büro. Die Arbeit mit verteilten Teams ist mit Scrum eher schwierig und erfordert eine besondere Aufmerksamkeit.

Im klassischen Scrum ist es so, dass bei mehreren Teams jedes Team einen Scrum Master und einen Product Owner hat. Die Product Owner aller Teams arbeiten im Product Owner Kreis zusammen. Darüber hinaus gibt es ein tägliches Scrum of Scrums Meeting, in dem die Abhängigkeiten zwischen den Teams aufgelöst werden.

3.7 Integrationstest

Insbesondere bei größeren Projekten kann es sinnvoll sein, vor einem Release (also i.d.R. nach mehreren Sprints) einen separaten Integrationstest durchzuführen.

4 Fazit

Scrum ist als Rahmenwerk für Business Intelligence Projekte hervorragend geeignet.

Der Prozess ist sehr einfach zu lernen und es gibt eine klare Rollenaufteilung. Die regelmäßige Zusammenarbeit mit dem Fachbereich und Priorisierung der Aufgaben nach fachlichem Nutzen sorgen dafür, dass Lösungen erstellt werden, die einen hohen Mehrwert für das Unternehmen bringen.

Scrum macht Spaß, das Team wird durch selbstbestimmtes arbeiten sehr motiviert. Durch die Zusammenarbeit interdisziplinärer Teams findet ein sehr guter Know-How Transfer statt und es werden Kopfmonopole abgebaut.

Insbesondere die flexible Reaktion auf Änderungen, die gerade im BI Umfeld gegeben und gewollt sind, macht Scrum zu einem sehr guten Rahmenwerk für BI Projekte.

5 Quellenangaben

Agileatlas.org

Agilemanifesto.org

It-agile.de

Do-better-scrum

Ralph Hughes, Agile Data Warehousing: Delivering World-Class Business Intelligence Systems Using Scrum and XP

Kontaktadresse:

Thomas Löchte
Informationsfabrik GmbH
Scheibenstr. 117
48153 Münster

Telefon: +49 (0) 251 - 919979-63
Fax: +49 (0) 251 - 919979-9
E-Mail: tloechte@informationsfabrik.com
Internet: www.informationsfabrik.com