

# Agile BI in der Praxis

## Warum SCRUM im DWH-Umfeld funktioniert

Andreas Ballenthin  
OPITZ CONSULTING Gummersbach GmbH

### Schlüsselworte:

Agile BI, SCRUM, Informatica, Business Objects, Projektmanagement, BI, DWH, Automatisiertes Deployment, Automatisierte Tests

### Einleitung

Das Thema Agile BI ist immer häufiger im Gespräch. Die letzten Jahre wurden methodische Ansätze aus der Anwendungsentwicklung (Schlagworte Scrum, Xtreme programming, Test driven Development) immer häufiger von der BI-Welt adaptiert.

Als wesentliche Idee dahinter steckt die Zielsetzung, die Schwächen der bestehenden BI-Vorgehensmodelle (typischerweise Wasserfall bzw. Weiterentwicklungen zu inkrementellen Varianten) zu korrigieren. Hier spielen verschiedene Schwächen eine Rolle. Für die Anforderer ist es sehr schwierig, bei der Anforderungsformulierung bereits alle Einzelheiten in ausreichender Tiefe zu formulieren. Auch besteht häufig eine gewisse Unsicherheit bezüglich der Steuerungskraft der formulierten Kennzahlen, Dimensionen und Berichte. Hinzu kommt, dass aufgrund der geschäftlichen Dynamik eine zeitliche Stabilität der Anforderungen und auch der Quellsysteme selten gegeben sind. Verschärft wird die Situation durch den prognostizierten Mangel an fähigen BI-Köpfen sowie den kontinuierlichen Kostendruck in der IT. In Summe versprechen agile Vorgehensweisen deshalb effizientere Ansätze zur erfolgreichen Entwicklung von BI-Lösungen.

Der Vortrag soll zeigen, welche Erfahrungen mit der agilen Entwicklung eines Enterprise-DWH gemacht wurden.

### SCRUM

In der Definition gibt es bei einem verhältnismäßig jungen Begriff wie SCRUM noch Spielraum. Im engeren Sinne werden jedoch hinter agiler Softwareentwicklung agile Methoden verstanden, mit denen Software entwickelt wird. SCRUM ist hierbei der populärste und vollständigste Vertreter, da bei dieser Methodik nicht nur Anforderungen und die eigentliche Entwicklungstätigkeit, sondern auch das Management der gesamten Projektstätigkeit berücksichtigt werden.

**Fehler! Verweisquelle konnte nicht gefunden werden.** stellt die wichtigsten Grundelemente des Scrum Frameworks dar, die es auf das Projekt abzubilden gilt: Das „Product Backlog“ mit den zu implementierenden „User Stories“ (Funktionen), das „Sprint Backlog“ mit der Auswahl der Funktionen, die im aktuellen Iterationszyklus, dem „Sprint“ (großer Kreis) umzusetzen sind, die „Daily Sprints“ (kleiner Kreis) zur Fortschrittskontrolle und schließlich das „Shippable Product“ – das einsatzfähige Produkt-Release, welches am Ende der Iteration die für diesen „Sprint“ geplanten Funktionen enthält.

In einem Scrum Team sind verschiedene Rollen einzunehmen. Der Product Owner definiert und priorisiert die Anforderungen und entscheidet über die Abnahme. Der Scrum Master verantwortet die Einhaltung der Scrum Methodik und fördert die Zusammenarbeit. Das eigentliche Team übernimmt die Entwicklung in Selbstorganisation und zeigt sich verantwortlich für das zu erstellende Produkt.

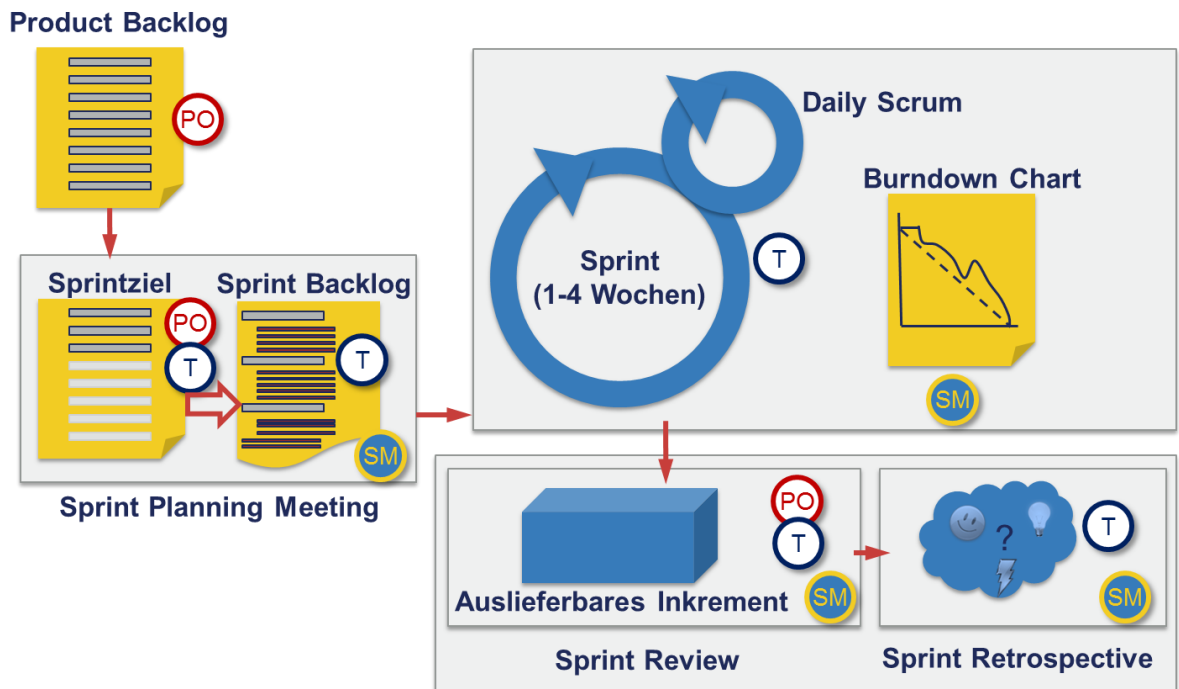


Abb. 1: SCRUM-Workflow

### Typische Herausforderungen im DWH/BI-Umfeld

Die SCRUM-Teamstruktur und die agile Arbeitsweise setzen gewisse Rahmenbedingungen voraus. Neben der notwendigen Managementunterstützung für eine solche Veränderung im Projektmanagement sind Teamgröße (5-9), interdisziplinäre BI Experten, Bereitschaft zur Transparenz und Teamfähigkeit der Beteiligten wichtige Erfolgsfaktoren.

Häufig wird Scrum durch externe Unterstützung oder durch Transfer aus einer benachbarten IT-Abteilung in die BI-Welt des Unternehmens getragen. Bei der Adaption sind jedoch die Besonderheiten der BI-Welt zu berücksichtigen. Hierbei treten typische Situationen auf, die inklusive ihrer Lösungsansätze aus der Praxis nachfolgend vorgestellt werden.

### SWOT-Analyse

<b>(S)trengths</b>	<b>(W)eaknesses</b>
Schnelle Lieferung	Langfristige Ergebnisse schwieriger planbar
Richtige Umsetzung der Anforderung	Keine Festpreise/Werkverträge
Schnelle Reaktion auf veränderte Anforderungen	SLA kompliziert
Risiken werden minimiert	Einsatz virtueller Teams schwierig
<b>(O)pportunities</b>	<b>(T)hreats</b>
Kundenzufriedenheit wird gesteigert	Das Team „spielt nicht mit“
Schlechter Ruf der unflexiblen DWH-Lösungen wird abgebaut	Der Auftraggeber/Fachbereich lässt sich nicht ausreichend involvieren
Spaß im Projektteam durch neues Vorgehen wird gefördert	DWH/Datenmodell wird unnötig komplex

## SWOT-Analyse – Maßnahmen

	<b>(S)trengths</b>	<b>(W)eaknesses</b>
<b>(O)pportunities</b>	Erfolgreiche BI Projekte durch schnelle Lieferung der genau benötigten, richtigen Lösung erzeugt hohe Kundenzufriedenheit	Langfristige Ergebnisse sind zwar schwieriger planbar, decken dafür aber genau sogar kurzfristig angepasste Anforderungen ab
<b>(T)hreats</b>	Kundeneinbindung fördern durch schnelle Reaktion auf angepasste Anforderungen und schnelle Lieferung	Frühzeitige Reaktion und Gegensteuerung auf Fehlentwicklungen

## SCRUM als Mittel der Risikominimierung

SCRUM dient in erster Linie der Risikominimierung!

<b>DWH/BI-Projekte klassisch</b>	<b>DWH/BI-Projekte agil</b>
Werden langwierig geplant	Langwierige Planung entfällt
Dauern zu lange	Releasezyklen werden kürzer
Liefern am Ende doch nicht den nötigen Mehrwert	Ergebnisse entsprechen den Anforderung des Kunden
Sind am Ende zu starr und zu unflexibel	Gehen direkt auf Kundenbedürfnisse ein
Kundenfeedback erfolgt erst nach langen Zyklen	Kundenfeedback erfolgt unmittelbar
Werden nur von wenigen Personen genutzt	Werden vielfältiger und häufiger benutzt
Projekte sind lange intransparent	Projekte werden transparent

## Pair Programming und Pair Design

Es hat sich bewährt, einfache Stories als Pair und komplexe Stories im kompletten Team zu planen. Dieses Vorgehen stellt sicher, dass kein wesentlicher Task übersehen wird.

Als notwendig hat sicher herausgestellt, dass nicht-triviale Tasks entweder direkt im Pair implementiert, mindestens aber von einem zweiten Teammitglied begutachtet werden.

Folgende Synergien kann man hierdurch bilden:

- Qualität der Software wird verbessert
- Fehler werden früh erkannt
- Mehr Spaß an der Arbeit
- Wissen verbreitet sich im gesamten Team
- Kommunikation im Team verbessert sich

Design- und Entwicklungskosten erhöhen sich dabei nur minimal.

## Team- und Skillzuschnitt

Zu Beginn des Projektes bestand das Team in der Mehrzahl aus Spezialisten der Themenfelder

- BI-Architektur
- Backend-Entwicklung
- Frontend-Entwicklung
- Test

Die Mitglieder eines Scrum-Teams müssen die Bereitschaft zeigen, sich auch mit Themengebieten außerhalb ihrer Spezialisierung zu befassen, um so das Team zu stärken. So beteiligt sich beispielsweise ein Backend-Entwickler auch an Frontend-Tests. Im Ergebnis stellt sich das Team stärker und schlagkräftiger auf und kann wesentlich besser auf Stories mit einer hohen Anforderung an einen bestimmten Skill oder Ausfälle im Team (Urlaub, Krankheit) reagieren.

### **Storyschnitt**

Konstrukte ohne fachlichen Mehrwert, wie eng geschnitten dieser auch immer sein möge, sind keine Stories. Für rein technische Stories ohne fachlichen Mehrwert gibt es häufig wenig Argumente.

Das Entwicklungsteam hat sich dazu entschieden, dass Stories neben Backendkomponenten immer auch eine nutzbare Frontendkomponente enthalten.

Notwendige Refactoring-Maßnahmen werden in den Aufwand einer Story hineingeschätzt.

Für gute Stories gelten folgende Kriterien:

- Die Sicht des Benutzers oder Kunden wird wiedergespiegelt
- Sie schaffen einen greifbaren Mehrwert, bzw. liefern eine greifbare Bedeutung
- Sie sind in der Sprache der Anwendungsdomäne dargelegt
- Sie enthalten keine Lösungsvorgabe, sondern nur fachliche Anforderungen
- Sie sind abschließbar, unabhängig, schätzbar und testbar

### **Automatisiertes Deployment**

Mit dem automatisierten Deployment werden folgende Ziele verfolgt:

- Automatisierung und Verkürzung der Arbeitsabläufe
- Erhöhte Prozesssicherheit und korrekte Darstellung von Ist-Zuständen
- Erhöhung der Qualität: konsistentes & reproduzierbares Deployment
- Verbessertes Verständnis der Systemlandschaft
- Abhängigkeit von einzelnen Personen reduziert

Im Projekt wurde dazu der Begriff des One-Batch-Deployments geprägt. Ein Deployment muss über einen Batch zu jeder beliebigen Zeit startbar sein.

### **Automatisierte Tests**

Der Aufwand für Regressionstests steigt von Story zu Story an. Um auch bei hoher Komplexität noch eine ausreichende Testabdeckung erreichen zu können, sind automatische Tests notwendig.

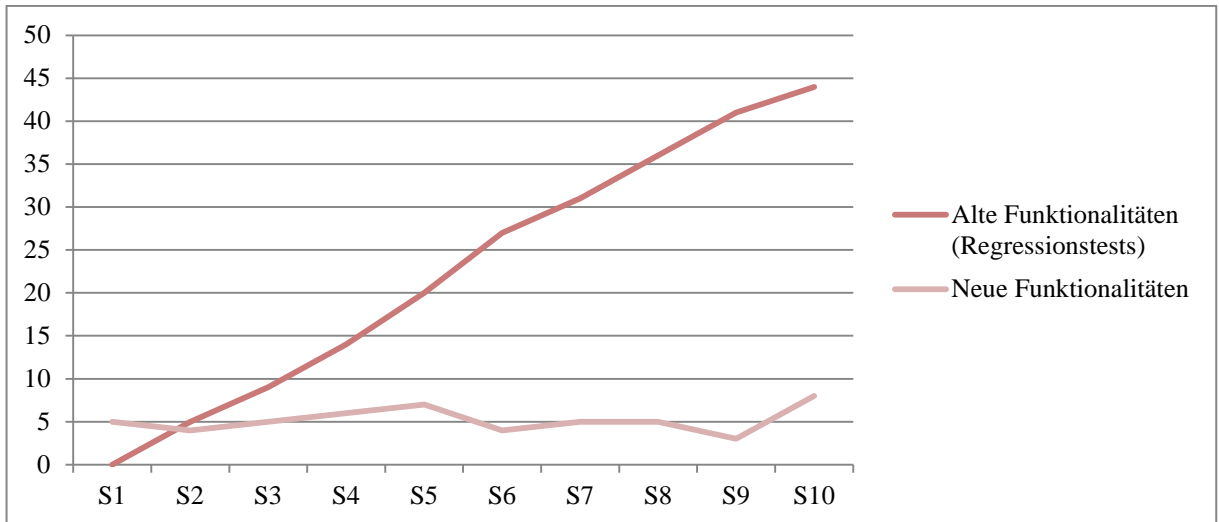


Abb. 2: Aufwand für Regressionstests

Zur Automatisierung der Backendtests wurden im Projekt ausschließlich Werkzeuge verwendet, die unter Windows skriptbar sind. Folgende Werkzeuge kamen zum Einsatz:

- Data Pump zur Erstellung oder zum Einspielen von DB-Dumps
- SQL\*Plus zur Ausführung von SQL-Skripts
- WinSCP zum Scripting von Windows-Linux-Interaktionen
- Batchdateien

Automatische Tests finden grundsätzlich nachts statt. Es wird täglich anhand von Batchvorlagen definiert, welche Tests in der nächsten Nacht stattfinden sollen. Codestände werden i. d. R. komplett vertestet. Am nachfolgenden Morgen liegen dann die Testergebnisse vor; Fehler können ggf. tagsüber behoben werden. Sehr interessant ist auch die Verwendung von WinSCP als skriptbare Windows-Linux-Interaktionskomponente. Hierzu definiert man ein Parameterfile wie folgt:

```
# Option - Batchdatei wird auch bei Fehlern ausgeführt
option batch on
# Option - keine Nachfrage beim Ueberschreiben
option confirm off
# Option - binaerer Uebertragungsmodus
option transfer binary
# Oeffne Verbindung zum Server mit den Benutzerparametern
open user:passwort@host -timeout=1800
# Parameterfile
cd /informatica/pmroot/BWParam
call cp global_parameter.parm_vorlage_1 global_parameter.parm
#Starte Workflow
cd /informatica/pmroot/script/automatic_test
put -nopermissions -nopreservetime 00150_DEP.sh
call dos2unix testfall1.sh
call chmod 750 testfall1.sh
call ./testfall1.sh
# Schliesse die Sitzung
close
# Beende WinSCP
exit
```

Dieses Parameterfile kann mit folgendem Aufruf verwendet werden:

```
"C:\Program Files\WinSCP\WinSCP.exe" /console  
/script=C:\SVN\%SPRINT%\06_test\00150_DEP.scp  
/log=C:\SVN\%SPRINT%\06_test\00150_DEP.log
```

### **Unsere Botschaft**

Scrum im DWH/BI-Umfeld funktioniert tatsächlich!

Automatisches Deployment und automatische Tests müssen etabliert werden, sonst steigen Testaufwände und/oder die Testabdeckung ist zu gering.

Es ist schwierig, das Entwicklungsteam durch den PO „abzuschirmen“. Somit liegt es in der Verantwortung des Teams, neue Anfragen von Fachbereichen abzublocken und auf den PO zu verweisen.

Es ist von großem Vorteil, wenn das Team räumlich beieinander sitzt. Virtuelle Teams harmonieren meist weniger gut.

Es ist vorteilhaft, wichtige Artefakte immer vor Augen zu haben, sodass sie sozusagen omnipräsent sind. So kann jedes Teammitglied sie zu jeder Zeit sehen, ohne erst mit drei bis sieben Klicks eine Anwendung starten zu müssen. Auch zu Abstimmungszwecken ist eine Visualisierung der Artefakte sinnvoll. Eine Aufgabe zu übernehmen wird durch das Verschieben von Tasks sicht- und erlebbar. Nach dem Motto „Haptik statt Mausclick“ hat man die Sache jetzt wahrlich „in die Hand genommen“. Damit entsteht fast von selbst ein gewisses Maß an Verpflichtung und Verbindlichkeit.

Generalisten statt Spezialisten – somit „skaliert“ das Team deutlich besser. Außerdem werden so Kopfmonopole verhindert, was sowohl für das Team, als auch für den Kunden hilfreich ist. Jeder Aufgabenbereich muss im Team mindestens doppelt besetzt sein.

Pair Design und Pair Programming sind wertvoll und notwendig. Sie erhöhen die Qualität der Software, vermindern Fehler und damit kostenaufwändige Fehlerbeseitigungen, wodurch sich die anfallenden Kosten neutralisieren.

Kontaktadresse:

Andreas Ballenthin  
OPITZ CONSULTING Gummersbach GmbH  
Kirchstraße 6  
D-51647 Gummersbach  
[andreas.ballenthin@opitz-consulting.com](mailto:andreas.ballenthin@opitz-consulting.com)

Telefon: +49 (0) 2261-6001-0  
Fax: +49 (0) 2261-6001-4000  
Internet: [www.opitz-consulting.com](http://www.opitz-consulting.com)