Benutzerhandbuch
Agile BI Modeler

Jon Nedelmann
02.01.2012

1. Einleitung

Nachdem sich agile Entwicklungsmethoden immer weiter etabliert haben, ist nun auch der Begriff *Agile BI* in den letzten Jahren mehr in den Fokus gerückt (vgl. z. B. ??).

Was mit Agiler BI gemeint ist, ist dabei aber nicht Konsens. Einige sehen darin die Adaption agiler Entwicklungsmethoden wie z. B. SCRUM oder XP auf Business Intelligence Projekte. Andere betonen eher die Tatsache, dass durch Cloud Technologien ein agileres Management der benötigten technischen Infrastruktur möglich ist. Wieder andere betonen die Architektur von Data Warehouse Systemen, die so gestaltet werden soll, dass Änderungen des Systems möglichst einfach werden.

In diesem Dokument möchte ich einen weiteren Aspekt hinzufügen: das agile Modellieren von Kennzahlsystemen. Damit schließe ich mich eher der ersten der beiden Sichtweisen an, indem ich tatsächlich SCRUM als Ausgangspunkt nehme, um ein Vorgehen für ein agiles BI-Projekt zu beschreiben.

Wie in den nächsten Abschnitten zu sehen ist, wird basierend auf der einfachen Struktur der User Story eine komplexe Kennzahlenontologie aufgebaut. Was zunächst kontraproduktiv klingt, da aus etwas Einfachem etwas Kompliziertes wird, wird seinen Zweck erweisen, wenn deutlich wird, welche Unterstützung der Entwickler bei Änderungen bekommt oder wie es ihm ermöglicht wird, zu Beginn des Projekts auf bereits vorhandene Kennzahlsysteme zurückgreifen zu können.

2. Überblick über den Entwicklungsprozess

3. Projektinitialisierung

Während der Projektinitialisierung werden die grundlegenden Weichen für das Projekt gestellt:

- Das Projektteam wird zusammengestellt. Dabei gehören zu dem Team nicht nur diejenigen, die Analyse- Design oder Entwicklungsaufgaben übernehmen, sondern jeder, der im Analyseprozess Anforderungen stellen oder bewerten darf. Dieser Punkt ist wichtig, da die gesamte Kommunikation bzgl. der Anforderungen innerhalb des Systems gehalten werden soll.
- Die Konfiguration des Szenarios wird vorgenommen. Damit werden bereits wichtige Entscheidungen bzgl. der Struktur der Szenariovision, des logischen und physikalischen Datenmodells getroffen
- In einer knappen Szenariovision werden die wesentlichen Ziele und Rahmenbedingungen dokumentiert.

In der Einleitung haben wir dargestellt, dass uns User Stories nur bedingt weiterhelfen. Wir möchten sie gerne durch etwas anderes ersetzen, und zwar durch Geschäftsfragen bzw. Business Questions. In diesem Kapitel werden wir die Struktur der zu verwendenden Geschäftsfragen erarbeiten.

Kontextelement und Kennzahlkandidat

Die beiden elementaren Bausteine, aus denen wir Vergleiche aufbauen sind Kennzahlkandidaten und Kontextelemente. Dabei verstehen wir unter einem Kennzahlkandidaten ein Element unserer Anforderungssprache, das eine Größe beschreibt, die in irgendeiner Form messbar ist. Wir bleiben an dieser Stelle bewusst ungenau. Ein Kennzahlkandidat wird durch seinen eindeutigen Namen identifiziert und durch umgebende Dollarzeichen gekennzeichnet.

Kontextelemente beschreiben den Kontext, in dem gemessen wird. Wir kennzeichnen sie durch eckige Klammern.

In der Frage „Wieviel Umsatz wurde in der letzten Woche in der Region Nord gemacht?“ sind letzte Woche und Region Nord die Kontextelement, Umsatz ist ein Kennzahlenkandidat. Wir können also in unserer erweiterten Wiki-Grammatik schreiben:

Wieviel $Umsatz$ wurde in der [letzten Woche] in der [Region Nord] gemacht?

Für die Kennzahlenontologie merken wir uns:

```
:MeasureCandidate  rdf:type rdfs:Class.
:ContextElement rdf:type  rdfs:Class.
```

Vergleich

In [Say it with charts] wird ein Verfahren vorgestellt, wie für Präsentationsfolien die geeigneten Grafiken ermittelt werden können. Dabei wird in drei Schritten vorgegangen:

1. Formuliere die Aussage.
2. Finde den passenden Vergleich.
3. Finde die passende Grafik.

In Standardberichten, Dashboards etc. geht es zunächst darum, Informationen darzustellen, weniger darum, sie bereits zu interpretieren bzw. zu einer markanten Aussage zusammenzufassen.

...

Die Schritte 2 und 3 sind aber auch für uns interessant. Dabei ist eine Grundbeobachtung, dass wir im Allgemeinen nicht an absoluten Zahlen sondern an Vergleichen interessiert sind. Er postuliert dabei, dass alle Vergleiche einem der folgenden fünf Vergleichstypen zugeordnet werden können.:

Zitieren aus Zelazny

COMPARISON_TYPE

Feststellung, dass es einen Typ und einen Subtypen gibt, dass es eine Unter- und eine Obergrenze für Kenzahlen und Attribute gibt.

Grammatik

Ontologie

```
:Comparison rdf:type rdfs:Class.
:ComparisonType: rdf:type  rdfs:Class.
:hasComparisonType rdfs:domain :Comparison.
:hasComparisonType rdfs:range :ComparisonType
```

Indikatoren

Die fünf Vergleichstypen erweitern wir noch um zwei Elemente, die vor allem in Dashboards und Scorecards gerne Verwendung finden: Trend- und Statusindikatoren:

Ein semantischer Wert ist ein beliebiger Ausdruck, dem eine Interpretation zugeordnet werden kann. Eine semantische Werteliste ist eine geordnete Reihe von Semantischen Werten. Eine Funktion, die Wertebereichen einer Kennzahl einen semantischen Wert zuordnet, nennen wir einen Indikator.

Beispiel. Zur Bewertung werden Schulnoten herangezogen

Wir können zwe

SEMANTIC_VALUE is a class

Interpretation und hat Interpretation dazunehmen

Eine semantische Werteliste ist eine geordnete Reihe von Semantischen Werten

SEMANTIC_VALUE _LIST is a class

Ausdrücken, dass Werteliste eine geordnete Reihe von Werten hat

Zwei Wertelisten können im Vergleich stehen, müssen aber nicht.

Ein Indicator ist ein Status Indicator oder ein Trend Indicator.
Einem Indicator wird damit der Typ (Status, Trend) zugeordnet und eine Verbindung zu einer Semantic Value List.

Allgemeiner Vergleich = Multidimensionale Frage

Ontologie

DIMENSIONAL_QUESTION is a class.
DIMENSIONAL_QUESTION has MEASURE_CANDIDATE
DIMENSIONAL_QUESTION has CONTEXT_ELEMENT

Grammatik

dimensionalQuestion ->
<question><words>measureCandidate*<words>context</question>

Beispiel einer Frage

A comparison type expects a certain amount of measures and context elements.

A multidimensional question has measures and context elements.

An atomic question is either an indicator or a comparison or a multidimensional question.

The <role>Sales Department</role> wants to<comparison> compare the time series of the $gross profit$ with respect to each [product group]</comparison> and represented for <filter>the last ?n? months</filter>

(role, List((type, List(measure), List(context), Filter)

4. Product Backlog

Im Aufbau eines Product Backlogs folgen wir zunächst wieder den Vorgaben von SCRUM:

Das Product Backlog enthält die Anforderungen an das zu entwickelnde BI-System. Jeden Eintrag eines Backlogs nennen wir Feature. Eine Anforderung Feature kann einen der folgenden Typen haben (ich habe mir erst gar nicht die Mühe genommen, die Begriffe ins Deutsche zu übersetzen):

- User Story
- Business Question
- Compliance Request
- Constraint
- Subject Request
- Information Object Request

Die Anforderungen werde in einer Baumstruktur abgelegt, die dem Prinzip folgt, dass die Kindelemente eine Anforderung verfeinern. Für die Anordnung in der Baumstruktur gelten außerdem die folgenden Regeln:

- Auf der obersten Ebene kann eine Anforderung eines beliebigen Typs angegeben werden.
- Ein Subject Request kann durch Business Questions, Compliance Requests , Constraints oder Information Object Requests verfeinert werden.
- Ein Information Object Request kann durch Business Questions, Compliance Requests oder Constraints verfeinert werden.
- Eine Business Questions oder ein Compliance Requests kann durch Constraints verfeinert werden.
- Ein Constraint auf oberster Ebene hat eine allgemeine Relevanz, ansonsten bezieht er sich auf sein Vaterelement.

Beispiel

5. Kennzahlsystem

6. Kennzahlontologie

7. Ableitung und Änderung

8. Architektur

Bisher hatten wir die Architektur des zu entwickelnden Systems grob vernachlässigt. Deshalb wird es dringend Zeit, dass wir uns ein wenig darum kümmern.

9. Deduktion

Fragen:

Realisiert das Kennzahlenmodell die Anforderungen?
Sind die Anforderungen realisierbar?
Ist ein Backlog genauer als ein anderes (ggf. nach Mapping).
Welche Teilmenge meiner Anforderungen kann durch ein Kennzahlenmodell realisert werden.

Kann ich mein Kennzahlenmodell auf ein anderes Mappen

10. Projektmanagement und Aufwandsschätzung

11. Anhang

Vollständige Kennzahlenontologie

Vollständige Grammatik

A Business Question hast the form

<role> and n atomic questions and n parameter or filter conditions

A comparison type can have furthermore parameter and filter conditions.

A Compliance Request ist he same as a Business Question, but from outside.

A User Story is either:

a) das Normale
b) die Angabe, dass n Business Questions innerhalb eines Informationsobjekts beantwortet werden sollen

Ein Constraint kann nichtfunktionale Anforderungen an dieses Objekt stellen.

Role has a is a relationship.
Zwischen User Stories kann folgender Zusammenhang stehen

Wenn A erfüllt ist, dann auch B.
Wenn das Datenmodell A beantworten kann (unter Berücksichtigung von Constraints) dann auch B.

Mapping von Kontextelementen
Mapping von Kennzahlenkandidaten

Ein Product Backlog enthält Business Questions, User Stories und Constraints.

Das ist dann aber auch die einzige Verbindung

Gilt dann die Beziehung <=